PC87591E, PC87591S and PC87591L LPC Mobile Embedded Controllers

National Semiconductor

PRELIMINARY
March 2004
Revision 1.07

PC87591E, PC87591S and PC87591L LPC Mobile Embedded Controllers

General Description

The National Semiconductor PC87591E, PC87591S and PC87591L are highly integrated, embedded controllers with an embedded-RISC core and integrated advanced functions. These devices are targeted for a wide range of portable applications that use the Low Pin Count (LPC) interface.

The PC87591S is targeted for security applications and includes supporting hardware such as the Hardware Random Number Generator. The PC8591L replaces the on-chip flash with 4K of boot ROM for value solutions using shared BIOS architecture. “PC87591x” refers to all the devices.

The PC87591x incorporates National’s CompactRISC™ CR16B core (a high-performance 16-bit RISC processor), on-chip flash (ROM for the PC87591L) and RAM memories, system support functions and a Bus Interface Unit (BIU) that directly interfaces with optional external memory (such as flash) and I/O devices.

System support functions include: WATCHDOG and other timers, interrupt control, general-purpose I/O (GPIO) with internal keyboard matrix scanning, PS/2® Interface, ACCESS.bus® interface, high accuracy analog-to-digital (ADC) and digital-to-analog converters (DAC) for battery charging, system control, system health monitoring and analog controls.

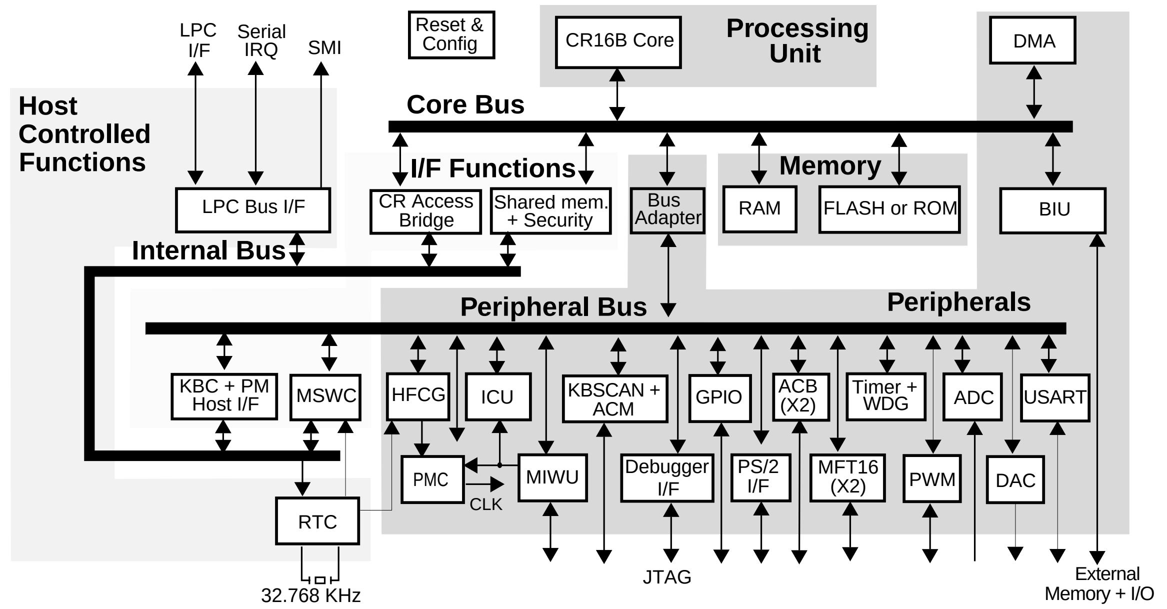
The PC87591x interfaces with the host via an LPC interface that provides the host with access to the Keyboard and embedded controller interface channels, integrated functions, Real-Time Clock (RTC), BIOS firmware and security functions.

Like members of National’s SuperI/O family, the PC87591x is PC01 and ACPI compliant.

Outstanding Features

- Host interface, based on Intel’s *LPC Interface Specification Revision 1.0*, September 29th, 1997
- PC01 Rev 1.0, and ACPI 2.0 compliant
- 16-bit RISC core, with 2 Mbyte address space, and running at up to 20 MHz
- Software and Hardware controlled clock throttling
- Shared BIOS flash memory (internal and/or external)
- Y2K-compliant RTC
- 84/117 GPIO ports (for 128-pin/176-pin packages) with a variety of wake-up events
- Extremely low current consumption in Idle mode
- JTAG-based debugger interface
- 128-pin and 176-pin options, in LQFP package (PC87591L is 176-pin only)

Block Diagram



Features (Continued)

Device-Specific Information

The following table compares features for the devices in the PC87591x family. The features listed below are a superset of the PC87591x family.

Function	PC87591E 128-Pin	PC87591E 176-Pin	PC87591S 128-Pin	PC87591S 176-Pin	PC87591L 176-Pin
Flash Size	64K	64K	128K	128K	–
ROM Size	–	–	–	–	4K
RAM Size	2K	2K	4K	4K	2K
General-Purpose Input/Output Ports (GPIO)	62	93	62	93	93
Shared BIOS	NO	YES	NO	YES	YES
RNG	NO	NO	YES	YES	NO
Memory Protection	NO	NO	YES	YES	YES

Features

- Processing Unit
  - — CompactRISC CR16B 16-bit embedded RISC processor core (the “core”)
  - — 2 Mbyte address space
- Internal Memory
  - — Up to 128 Kbytes of on-chip flash memory (4 Kbytes of ROM in the PC87591L)
  - — Supports BIOS (flash) memory sharing with PC host
  - — On-chip flash is field upgradable by host, CR16B, parallel interface or JTAG
  - — Boot blocks for both CR16B and Host Code
  - — Memory contents protection and security
  - — Hardware-protected boot zone with block protection circuit
  - — 2K (PC87591E/L) or 4K of on-chip RAM (PC87591S)
  - — All memory types can hold both code and data
- Expansion Memory (Optional in 176-pin package)
  - — Up to 1 Mbyte of additional code and data
  - — Supports BIOS (flash) memory sharing with PC host
  - — Supports external memory power-down mode
  - — Field upgradable with flash or SRAM devices
  - — Supports host-controlled code download and update
  - — Bus Interface Unit (BIU)
    - ❏ Three address zones for static devices (SRAM, ROM, flash, I/O)
    - ❏ Configurable wait states and fast-read, single-cycle bus cycles
    - ❏ 8- or 16-bit wide bus
- LPC System Interface
  - — Synchronous cycles, up to 33 MHz bus clock
  - — Serial IRQ
  - — I/O and Memory read and write cycles
  - — Bootable memory support
  - — Bus Master read and write cycles
  - — Reset input
  - — Base Address (BADDR) strap to determine the base address of the Index-Data register pair
  - — $\overline{\text{LPCPD}}$ and $\overline{\text{CLKRUN}}$ support
  - — FWH Transaction support
- Security Function Support
  - — Random Number Generator (RNG)
  - — Full Random using temperature, voltage and system noise.
  - — Memory access protection

Embedded Controller System Features

- Host Bus Interface (HBI)
  - — Three host interface channels, typically used for the KBC, ACPI Private or Shared EC channels
  - — 8042 KBC standard Interface (legacy 60_{16}, 64_{16})
  - — Intel 80C51SL compatible
  - — IRQ1 and IRQ12 support
  - — Fast Gate A20 and Fast Host Reset via firmware
  - — PM interface port (legacy 62_{16}, 66_{16})
  - — ACPI Embedded Controller with either Shared or Private interface
  - — IRQ, SMI or SCI generation

Features (Continued)

- Interrupt Control Unit (ICU)
  - 31 maskable vectored interrupts (of which 26 are external)
  - General-purpose external interrupt inputs through MIWU
  - Enable and pending indication for each interrupt
  - Non-maskable interrupt input
- Multi-Input Wake-Up (MIWU)
  - Supports up to 32 wake-up or interrupt inputs
  - Generates wake-up event to PMC (Power Management Controller)
  - Generates interrupts to ICU
  - Provides user-selectable trigger conditions
- General-Purpose I/O (GPIO)
  - 84/117 port pins in 128-pin/176-pin package, respectively.
  - I/O pins individually configured as input or output
  - Configurable internal pull-up resistors
  - Special ports for internal keyboard matrix scanning
    - 16 open-collector outputs
    - Eight Schmitt inputs with internal pull-ups
  - Input for system On/Off switch
  - 27 external wake-up events
  - Low-cost external GPIO expansion through the BIU I/O Expansion protocol
- PS/2 Interface
  - Supports four external ports: Keyboard, mouse and two additional pointing devices
  - Supports byte-level handling via hardware accelerator
- Two ACCESS.bus (ACB) Interface modules. Each is:
  - Intel SMBus® and Philips I^{2}C® compatible
  - ACCESS.bus master and slave
  - Up to three simultaneous slave addresses detected
  - Supports polling and interrupt controlled operation
  - Generates a wake-up signal on detection of a Start Condition while in Idle mode
  - Optional internal pull-up on SDA and SCL pins
- Two 16-bit Multi Function Timer (MFT16) modules. Each module:
  - Contains two 16-bit timers
  - Supports Pulse Width Modulation (PWM), Capture and Counter
- Universal Synchronous/ Asynchronous Receiver-transmitter (USART)
  - A full-duplex USART channel
  - Programmable baud rate
  - Data transfer via interrupt or polling
  - Synchronous mode with either internal or external clock
  - 7-, 8- or 9-bit protocols.
- Pulse Width Modulation (PWM) Module
  - Eight outputs
  - 8-bit duty cycle resolution
  - Common input clock prescaler
- Timer and WATCHDOG (TWM)
  - 16-bit periodic interrupt timer with 30 µs resolution and 5-bit prescaler for system tick and periodic wake-up tasks
  - 8-bit WATCHDOG timer
- Analog to Digital Converter (ADC)
  - Fourteen channels, with 10-bit resolution
  - Sigma-delta technology for high noise rejection
  - Three voltage measurements and one temperature measurement every 100 ms
  - Internal voltage reference
- Hardware Monitoring
  - Controlled by embedded controller
  - System Voltage Measurement
    - Up to eight external measurement points
    - Four internal measurement points
    - Smart power failure detection
  - Diode-Based Temperature Measurement
    - Software-controlled fault detection
    - Hardware-monitored over-temperature detection
  - Production time calibration using flash parameters
- Digital to Analog Converter (DAC)
  - Four channels, 8-bit resolution
  - 1 µs conversion time for 50 pF load
  - Full output range from AGND to AVCC
- Analog Comparators Monitor (ACM)
  - Eight comparator inputs on KBD scan inputs
  - 6-bit input measurement resolution
  - Scan and Threshold modes
  - Supports low-current system wake-up
- Development Support Features
  - Interface to debugger via JTAG pins
    - ISE/ADB mode
    - On-board Debug mode
  - Flash programing via JTAG
- CR16B Access to Host Controlled Functions
  - Enabled when host inactive

Host Controlled Functions Features

- Supports *Microsoft® Advanced Power Management (APM) Specifications Revision 1.2*, February 1996
  - Generates the System Management Interrupt (SMI)
- PC1 and ACPI Compliant
  - PnP Configuration Register structure
  - Flexible resource allocation for all logical devices
    - Relocatable base address

Features (Continued)

    - ❑ 15 IRQ routing options

- Legacy free support
- Real-Time Clock (RTC)
  - DS1287 and MC146818 compatible
  - 242-byte battery backed-up CMOS RAM
  - Calendar including century and automatic leap-year adjustment (Y2K compliant)
  - Optional adjustment for daylight saving time
  - BCD or binary format for timekeeping
  - Three individually maskable interrupt event flags: periodic rates from 122 µs to 500 ms; time-of-day alarm, once-per-second to once-per-day
  - Double-buffer time registers
  - Alarm wake-up
- Mobile System Wake-Up Control (MSWC)
  - Wake-up on detection of $\overline{\text{RI1}}$, $\overline{\text{RI2}}$, $\overline{\text{RING}}$ activity
    - ❑ External modem ring on serial port
    - ❑ Ring pulse or pulse train on $\overline{\text{RING}}$ input signal
    - ❑ Software-controlled off events
  - Optional routing of power-up request on IRQ and/or SMI lines

Clocking, Supply and Package Information

- Strap Input Controlled Operating Modes
  - Turn on shared BIOS
  - TRI-STATE
  - Development
  - On-board development
  - Programing environment
- Clocks
  - Single 32.768 KHz crystal oscillator
  - LPC clock, up to 33 MHz
  - On-chip high frequency clock generator
    - ❑ CPU clock 4-20 MHz
    - ❑ Software-controlled frequency generation
    - ❑ Multiplier source 32 KHz input
  - 32 KHz clock out
  - CR16B clock out
- Power Supply
  - 3.3V supply operation
  - 5V tolerance and back-drive protection on all pins (except LPC bus pins and keyboard scan inputs)
  - Separate supply for Host I/F (V_{DD}) and Embedded Controller functions (V_{CC})
  - Backup battery input for RTC, and wake-up configuration.
  - Reduced power consumption capability
  - Four power modes, switched by software or hardware
    - ❑ Active mode current (TBD mA)
    - ❑ Active mode executing WAIT
    - ❑ Idle (15 µA)
    - ❑ Power off for RTC only (0.9 µA typical) from backup battery
  - Automatic wake-up on system events
- Package Options
  - 128-pin LQFP package
  - 176-pin LQFP package for more GPIO pins, Expansion Memory use and development systems

Revision Record

Revision Date	Status	Comments
April 11, 00	0.13	External version of Architectural Specification
December 25, 00	1.0	Preliminary
May 1, 01	1.02	Preliminary
May 8, 01	1.03	Preliminary
July 19, 01	1.04	Preliminary
October 28, 01	1.05	Preliminary
July 2002	1.06	Preliminary
April 2004	1.07	Preliminary: All references to CSP package are removed in Rev 1.07.

Table of Contents

Table of Contents (Continued)

3.0 Power, Reset and Clocks

4.0 Embedded Controller Modules

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

5.0 Host Controller Interface Modules

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

7.0 Device Specifications

Table of Contents (Continued)

Table of Contents (Continued)

1.0 Introduction

1.1 DOCUMENT ORGANIZATION

This document describes the PC87591x architecture and device specifications. It is organized as follows:

Chapter 1, Introduction, provides an overview of PC87591x modules, system connections, operating modes and configuration.

Chapter 2, Signal/Pin Description and Configuration, lists the PC87591x pins and describes their functions and multiplexing options.

Chapter 3, Power, Reset and Clocks, describes the PC87591x power supplies, clock scheme and reset sequence.

Chapter 4, Embedded Controller Modules, describes the modules that comprise the CompactRISC core peripherals.

Chapter 5, Host Controller Interface Modules, describes the modules and functions that interface core operation with the host.

Chapter 6, Host-Controlled Modules and Host Interface, defines the configuration and control functions.

Chapter 7, Device Specifications, defines the AC, DC and analog characteristics of the PC87591x.

Appendix A, Summary of Registers, provides a composite listing of all relevant data on core domain registers and summarizes the registers' layouts.

Appendix B, This section includes directions for the software to handle some predetermined hardware interfaces and provides details of software-to-hardware interface conventions.

1.2 GENERAL DESCRIPTION

The PC87591x is a highly integrated, embedded controller with an embedded RISC core and system functions. Targeted for a wide range of portable applications that use the Low Pin Count (LPC) interface, it also features a security system and host BIOS firmware.

1.2.1 System Connections

Figure 1 shows the system connections of the PC87591x in a typical mobile PC application. The PC87591x requires little, if any, system glue elements. For a typical application, the PC87591x includes all required memory and peripherals on-chip. For more complex applications, it allows simple low-cost expansion, using its bus. Some of the features illustrated are mutually exclusive, depending on pin functions.

The major elements of the PC87591x are:

- Embedded Controller (EC) functions, which include: PS/2 devices, keyboard matrix, ACCESS.bus, timers, D/A and A/D converters and GPIO pins that can be assigned to various functions, as needed. External memory and peripheral devices may be added to extend the functionality of the on-chip resources.
- Host Processor interface based on the LPC bus and additional signals for interrupts and system power management
- Power Supplies for Host interface functions, EC and backup battery
- Clocks, using a 32.768 KHz crystal and optional clock output
- Strap inputs to initialize the PC87591x to different operation modes

In addition to the wide range of internal peripherals, the PC87591x provides hooks so that the system can be expanded in an easy and cost-effective manner, as follows:

- I/O expansion to support additional I/O port pins, using low-cost, standard 74HCxx devices or ASICs
- On-chip memories may be expanded to interface with external RAM, flash or ROM devices (176-pin package only).

1.2.2 Power Management

The PC87591x has an advanced power management scheme controlled by the host and/or the PC87591x firmware. The supported SuperI/O functions may be controlled directly by the host, using ACPI compliant schemes. The EC may also interface with the host via one or two communication channels. Power Management events are available for ACPI-compliant operation with the respective parts of the host chipset.

The PC87591x is designed to operate as the embedded controller of an ACPI-compliant system. It is equipped with various system power monitor and control functions and advanced means to control its own power consumption and power modes.

ACPI-compliant wake-up and sleep control is integrated into the PC87591x. These functions are powered by $V_{CC.}$

1.0 Introduction (Continued)

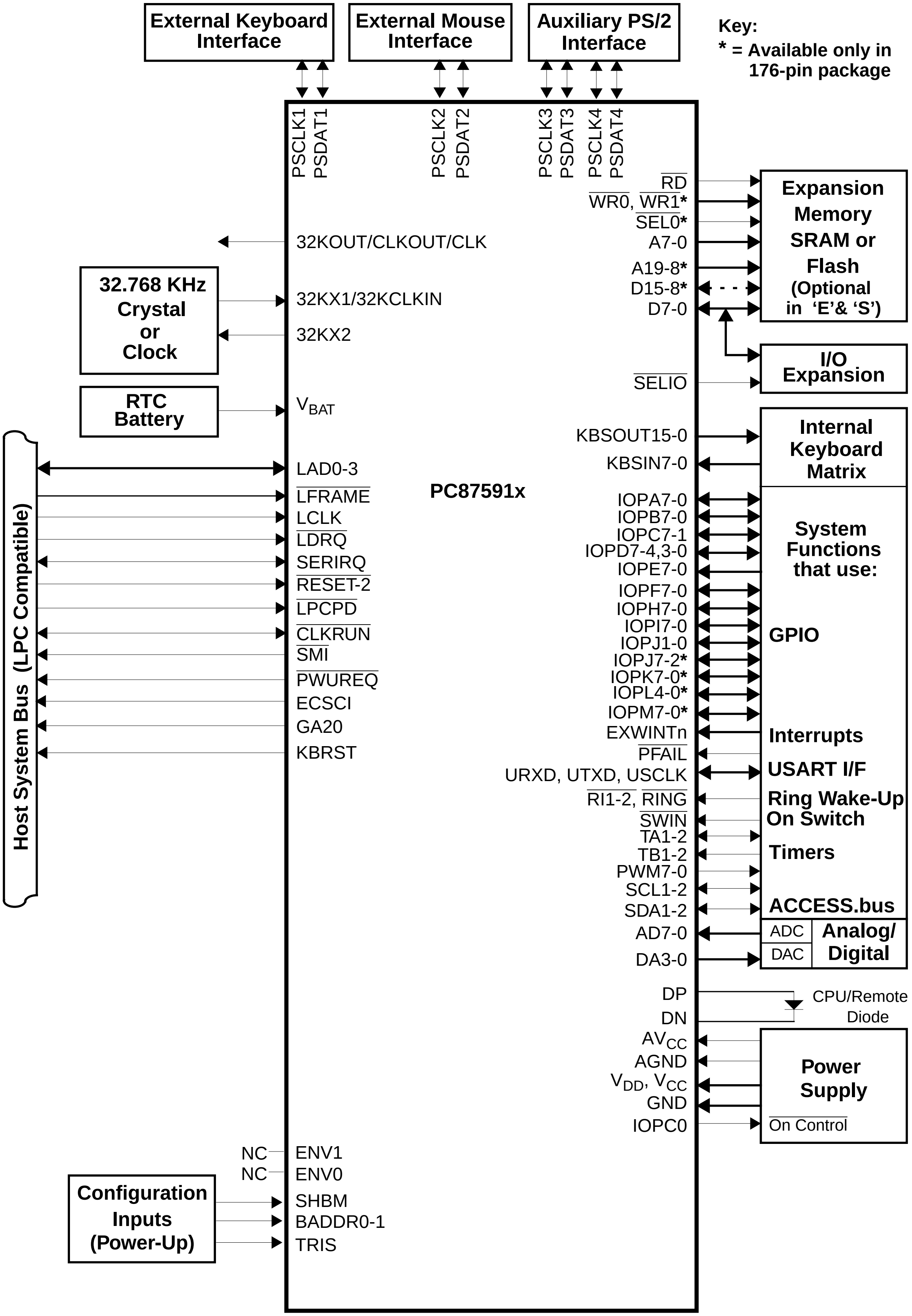


Figure 1. PC87591x System Connection Diagram, 128-Pin and 176-Pin Packages

1.0 Introduction (Continued)

The Keyboard Controller and the EC functions (core and associated peripherals) are powered by V_{CC}, which remains active as long as the system has a power source (e.g., main battery or outlet). Using V_{CC}, the core may be programed to monitor and control the system even when the host processor is turned off. To support this, the PC87591x is equipped with advanced means to control its power consumption: software controlled clock frequency throttling, the ability to disable modules, and four power modes. These power modes include:

- Active Mode - Full functionality
- Active Mode Executing WAIT Instruction - Core execution and associated operations (such as memory access) are suspended
- Idle - Main clock is stopped, but the device can be woken up by internal or external events. The system tick timer is still operational and can be used to periodically wake up the device.
- Power Off - V_{CC} is absent and only backup battery is available to supply the Real Time Clock (RTC) and retain the state of some memory and configuration elements.

1.2.3 Operating Environments

The PC87591x can operate in one of the following four environments:

- **Internal ROM Enabled (IRE)**. Used while the PC87591x operates in the production system and executes the application. The on-chip flash or ROM is the main source of code for the device.
- **On-Board Development (OBD)**. Used to debug the PC87591x code while it is mounted on its final production board. All pins have their IRE functionality. Interface to a debugger (running on the host) is through the JTAG-based debugger interface. OBD environment is binary and cycle-by-cycle compatible with IRE environment.
- **Development (DEV).** Used in Application Development Boards (ADB) or In System Emulators (ISE). In this environment, the on-chip flash or ROM is replaced with off-chip SRAM memory to allow flexible and fast development of application code. Some pins are allocated to development system use, and the GPIO functions associated with them are replicated using off-chip logic as part of the ADB system. DEV environment is binary and cycle-by-cycle compatible with OBD and IRE environments.
- **Programing (PROG).** Used to program the on-chip flash in parallel programers. No other operation is possible in this environment.

1.3 INTERNAL ARCHITECTURE

The PC87591x consists of several main components divided into three groups:

- Core Domain
  - CR16B core processing unit
  - Bus Interface Unit and Memory Controller (BIU)
  - RAM and flash memory
  - Core peripherals
- Host Domain
  - Host-Controlled functions
  - Host Interface
- Host-Controller Interface

The descriptions below detail the functions of the various blocks shown in Figure 2.

1.0 Introduction (Continued)

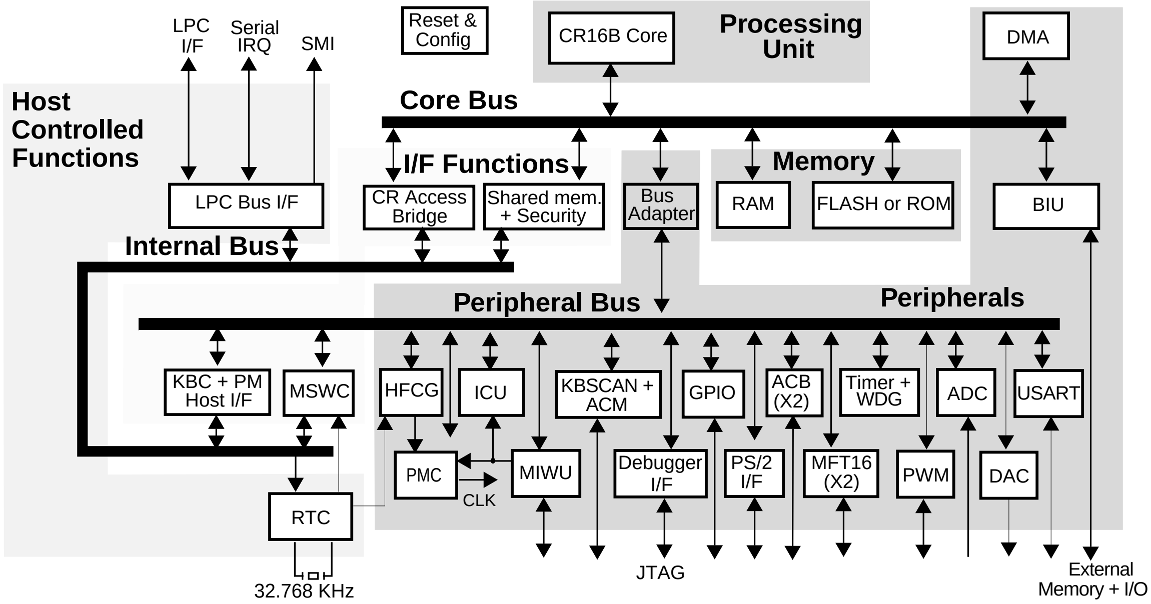


Figure 2. PC87591x Functional Block Diagram

1.3.1 Processing Unit

The CompactRISC CR16B core (referred to in this datasheet as the “core”) is an advanced, general-purpose 16-bit microprocessor core with a RISC architecture. The core is responsible for arithmetic and logic operations, as well as program control.

For more details about the core structure and instruction set, see *CR16B Programmer’s Reference Manual, Revision 2.2,* September 1999 (Literature Number: 633150-001)

1.3.2 Bus Interface Unit and Memory Controller (BIU)

The BIU enables access to off-chip memory and I/O devices. It is organized in zones, as follows:

- Zone 0 - Expansion Memory (flash and/or SRAM). This memory may be used for the core code, data and/or the host BIOS program.
- Zone 1 - This zone is available for off-chip in DEV environment only and is used for emulating the operation of the off-chip base memory, using an off-chip SRAM. In IRE and OBD environments, the configuration of this zone should be the same as in DEV environment to enable cycle-by-cycle compatibility.

  The Base memory is partitioned into two zones, fast and slow. The access time to the fast zone is defined by the configuration registers of Zone 1 in the BIU. The access time to the slow zone is defined by the configuration registers of Zone 2 in the BIU. Only one chip select signal is used—the AND of the two BIU select outputs.
- I/O Zone - This zone can be used for I/O expansion. In DEV environment, it can be used to recreate GPIO signals, whose pins are used for the development system interface.

Configuration registers that control the bus transactions are associated with each zone and are part of the BIU module, See Section 4.1 on page 69 for details. For details about the link between DEV environment and the BIU, see Section 1.4.3 on page 30.

1.3.3 On-Chip Memory

RAM. The 2048/4096-byte on-chip RAM is mostly used for the storage of program variables and the stack. It can also store short programs used while the flash memory is being updated. Part of the on-chip RAM is reserved for use by the core development tools monitor program, TMON. For information, see the *CompactRISC™ PC87591x Tmonlib Version 3.1.2.3 Release Letter*, March 2001,

Flash. The PC87591E and PC87591S are equipped with a large on-chip flash. This flash includes a main block and an information block. The flash main block is also referred to as on-chip Base Memory and may be used to store core code and constant data.

The PC87591x hardware arbitrates flash usage by the core firmware and the host processor BIOS program when Shared-BIOS configuration is selected. Flash sharing is based on in-parallel “cycle stealing” so both the host processor and the core can execute code in parallel from the same memory device. The host processor typically copies the flash contents to the host’s main memory (DRAM) on system boot to improve access time to it and enable execution when the flash’s contents is compressed. It is important to do this early in the boot process to reduce resource contention between the core and the host.

The PC87591E and PC87591S flash memory may be updated by the core firmware, host software, JTAG protocol or via the parallel interface. Protection mechanisms limit the read, write and erase rights of various sources to guarantee flash content integrity and confidentiality, where applicable. See Section 4.16.6 on page 226 for more details about the flash update and protection mechanisms.

In DEV environment, access to the main block of the flash is disabled and replaced by off-chip SRAM to enable easy and efficient debugging. For operation in OBD environment, the application should be linked with the TMON program and loaded into the on-chip flash. See the *CompactRISC™ PC87591x Tmonlib Version 3.1.2.3 Release Letter*, March 2001, for information on TMON program integration and specification of requirements.

The PC87591L is equipped with a small pre-programed flash, which functions as a boot ROM. The devices are shipped pre-programed and may not be programed in the field. Some devices intended for development purpose may be programed via the JTAG interface.

The information block of the flash has two dedicated areas: one for factory parameters and another for the protection word. The factory parameters are saved during the device production and are used for various calibrations (see Appendix B.1 on page 434 for the factory parameters block organization and usage). The protection word holds information on access rights to the flash and the size of the core and host boot blocks. See Section 4.16.6 on page 226 and Appendix B.2 on page 435 for more information.

1.3.4 Peripherals

The ICU (Interrupt Control Unit) collects interrupts from various internal and external (through the MIWU) sources and notifies the core of the event using the vectored interrupt mechanism. It supports 31 maskable interrupt inputs (see Table 16 on page 99 for the interrupt assignment) and a non-maskable interrupt (NMI) through the $\overline{\text{PFAIL}}$ input.

The MIWU (Multi-Input Wake-Up) module enables collecting various internal and external interrupt sources (events), generates interrupts in Active mode and enables the PC87591x to return from Idle mode to Active mode. The core can individually enable or disable the various wake-up conditions. The PC87591x has a total of 28 wake-up signals, some of which are grouped to generate a single interrupt signal to the ICU.

The PMC (Power Management Controller) controls PC87591x power consumption according to the required activity level. Power consumption is adjusted by controlling the clock frequency and selective enabling/disabling of three power modes: Active, Idle and Power Off. Activity can be resumed by external events (through the MIWU) or internal events, such as a periodic wake-up.

The Clock Generator provides clocks for the various core-related on-chip modules. These clocks are generated directly from a 32.768 KHz crystal or from the on-chip High-Frequency Clock Generator (HFCG). The HFCG generates the high-frequency clock using the RTC’s 32.768 KHz clock signal as a reference. The PC87591x operation frequency is set by programing the HFCG registers. The PMC enables and disables high-frequency clock generation, according to the required power mode.

The GPIO Ports (General-Purpose Input/Output) module consists of up to 117 GPIO signals (84 in the 128-pin package) that provide interface and control for the PC system. Some of these I/O port signals share their pins with an alternate function (see Table 8 on page 54) and may be mutually exclusive. When configured as inputs, some of these signals can interrupt the core when an event is detected, even if the device is in Idle mode. An example is the SWIN input, which is dedicated to the PC On/Off switch.

Internal keyboard scanning is supported by 16 open-drain output signals and eight input signals. Switch-based keyboard matrices are supported with CMOS Schmitt trigger inputs with internal pull-up resistors. Resistive keyboard matrices are supported with analog comparator inputs with variable thresholds and optional internal pull-up resistors. Section 4.14 on page 208 describes the functionality of the analog comparator (ACM) circuitry. For power efficiency, the inputs include an interrupt and a wake-up capability, so that pressing/releasing keys may be identified without scanning the keyboard matrix in either Active or Idle modes. The keyboard interrupt is controlled by the MIWU.

The PS/2 Interface enables interface with industry-standard, PS/2-compatible keyboard, mouse and other pointing devices. The PC87591x supports up to four PS/2 devices via its dedicated 4-channel PS/2 interface module. Each channel has two quasi-bidirectional signals. The symmetric structure of the channels enables software-controlled interchanging of devices to channels.

The PC87591x includes a hardware accelerator, which allows the PS/2 channels to be controlled with minimal software overhead. It also eliminates the sensitivity to interrupt latency that characterized traditional solutions.

1.0 Introduction (Continued)

The ACB Interface is a two-wire serial interface compatible with the ACCESS.bus physical layer. It is also compatible with Intel’s SMBus and Philips’ I^{2}C. This module can serve as a bus master or slave and performs both transmit or receive operations. As a slave, it can respond to two assigned addresses, a global call address and an SMBus ARP address.

The PC87591x includes two ACB Interface modules to allow operation on two isolated buses in the system.

The USART (Universal Synchronous Asynchronous Receiver Transmitter) gives full-duplex support for a wide range of software programmable baud rates and data formats. It handles automatic parity generation and several error detection schemes. It also supports DMA transfers to allow for fast processor-independent receive and transmit. It implements flow control logic to provide the capability for hardware handshaking.

The MFT16 (Multi-Function 16-Bit Timer) contains two 16-bit timers with a range of operation modes. These timers can operate, using several clock sources, in PWM, Capture or Counter mode to satisfy a wide range of application requirements.

The PC87591x includes two MFT16 modules, each of which may be assigned to functions and configured independently.

The TWD (Timer and WATCHDOG) module has a 16-bit periodic interrupt timer that can be programed to generate interrupts at pre-defined intervals and an 8-bit WATCHDOG timer that can reset the PC87591x whenever the software loses control of the processor.

The periodic timer is typically used as a system tick timer. This timer is fed by the 32.768 KHz clock. Thus its counting is not impacted by the setting of the HFCG, and it may continue to operate even in Idle mode. This enables it to serve as a periodic wake-up source during Idle mode.

The PWM (Pulse Width Modulator) module provides eight modulated outputs. The output signals have the same programmable frequency and individually-controlled duty cycles.

The ADC (Analog to Digital Converter) provides the PC87591x with an accurate means for measuring slow changing voltages and temperature. The ADC module can measure up to ten external and four internal voltages with 10-bit resolution over a voltage range of 0 to 2.97V. It can measure temperature using thermistors or perform diode-based temperature measurement. Special circuits are provided to allow accurate measurement of temperature, based on the voltage drop on the diode junction. This technique is tuned for measuring the temperature of processors with an internal diode.

The DAC (Digital to Analog Converter) has four channels of voltage output. Each of the four DAC channels has an 8-bit resolution with a full output range from AGND to AV_{CC}. The DAC provides a settling time of about 1 µs on a 50 pF load.

The Debugger Interface module provides a JTAG-based interface to a remote, host-based debugger. This interface enables device debugging while in OBD environment (i.e., in the final production board) or in DEV environment once in the development system. The Debugger Interface module uses the JTAG interface to enable programing of the on-chip flash in both these environments.

1.3.5 Host-Controller Interface Modules

Chapter 5 on page 277 describes a set of modules that resides in the boundary between the host-controlled functions and the core-controlled functions. These modules are used for message communication, data exchange, memory access and generating power management events to the host. Chapter 5 also discusses the mechanism that enables the core to access the host-controlled peripherals.

The Keyboard Controller, Power Management module has three channels that are available for keyboard and power management (EC)-related host-controller communication.

The keyboard and mouse data channel (i.e., host legacy I/O addresses 60_{16} and 64_{16}) is compatible with the legacy interface of keyboard controllers. It may be used with polling or interrupts. For use with interrupts, the module can generate the two legacy IRQ signals: IRQ1 and IRQ12. In addition, the PC87591x generates the gate A20 control signal (GA20 pin) and a soft reset signal ($\overline{\text{KBRST}}$ pin) to the host. Optionally, this $\overline{\text{KBRST}}$ reset signal can be used to prevent the host from accessing the shared flash when the PC87591x cannot perform the shared memory access during PC87591x boot-up. See “GA20 Pin Functionality” on page 322 and “Host Keyboard Fast Reset” on page 322 for details).

The PC87591x supports three Power Management channels, compliant with ACPI requirements for Embedded Controller (EC) interface. This enables the PC87591x to implement an EC interface that operates in either shared or private modes. The number of Power Management channels in use and the addresses they respond to (i.e., legacy host I/O addresses 62_{16} and 66_{16}) are configured in the Host Controlled Functions configuration space. These channels may generate IRQ, $\overline{\text{SMI}}$ or $\overline{\text{SCI}}$ events to the host. The Power Management channels include a PC87570-compatible mode and an enhanced scheme that enables more efficient control by the core.

Shared Memory and Security is supported between the host and the core. This sharing may be used for the support of a shared BIOS scheme, secured information storage and/or for the PC87591x firmware update by the host. The base memory of the PC87591x and, if needed, the expansion memory can be used as shared memory. The use of the Expansion memory is optional. The Shared Memory module provides means for the host to access the shared memory. It can also secure access to portions of the shared memory for read and/or write operations to allow reliable and tamper-protected storage and secured update. In addition (PC87591S only), this module includes a random number generator (RNG) that provides hardware-generated random numbers to the core.

1.0 Introduction (Continued)

The Mobile System Wake-Up module includes various system wake-up and power management services that may be handled either by the host or the core. ¨(wake-upsources may be the RTC or external events such as ring detection on the $\overline{\text{RING}}$ input or modem RI inputs. The module provides hooks for ACPI-compliant drivers, which enable the drivers to handle wake-up events, change the system power state (including turning it off) and interface to the core firmware. Mask bits can be enabled to determine whether the core or the host handles each one of the events. In addition, this module provides status information about the host domain (e.g., reset input state and V_{DD} supply status).

The Core Access to Host-Controlled Peripherals module enables core access to SuperI/O modules. It can interleave usage of a module with the host or take control of it and prevent any host access to that module.

1.3.6 Host-Controlled SuperI/O Modules and Host Interface

The Host Interface is based on Intel's Low Pin Count (LPC) interface, as defined in *LPC Interface Specification, Revision 1.0*. This interface enables the host to perform read and write cycles using I/O space accesses and memory space accesses and FWH transactions. Interrupts are sent to the host, using the serial IRQ protocol.

The PC87591x supports the advanced power management features of the LPC bus. The $\overline{\text{SMI}}$ signal may be sent to interrupt the host and put it in System Management Mode (SMM). The $\overline{\text{PWUREQ}}$ signal may be connected to one of the wake-up inputs of the host chipset and used to trigger an SCI event for various EC communication purposes. The PC87591x can operate with a slowed down or stopped LPC clock and can re-start the LPC clock as part of the system power management capabilities, using the $\overline{\text{CLKRUN}}$ signal. The $\overline{\text{LPCPD}}$ input enables turning off LPC bus supply while the PC87591x and some Host Controlled functions are operating.

Host Configuration. The PC87591x includes seven logical devices, each with associated configuration registers, in addition to a set of global configuration registers.

The central configuration register set supports ACPI-compliant PnP configuration. The configuration registers are structured as a subset of the Plug and Play Standard registers defined in Appendix A of the *Plug and Play ISA Specification, Revision 1.0a* by Intel and Microsoft. All system resources assigned to the functional blocks (I/O address space and IRQ lines) are configured in and managed by the central configuration register set. In addition, some function-specific parameters are configurable through the configuration registers and distributed to the functional blocks through special control signals.

The RTC (Real Time Clock) has a low-power timekeeping mechanism that provides a time-of-day, year-2000-compatible calendar with a century counter and alarm features. It can work from either V_{CC} or a backup battery, using an internal switch. Other features include three maskable interrupt sources and 242 bytes of general-purpose RAM. An external battery source maintains valid RAM and time during V_{CC} failure. The RTC is software compatible with the DS1287 and MC146818.

1.4 OPERATING ENVIRONMENTS

On power-up reset, the ENV1-0 and TRIS input signals select one of the following operating environments:

- Internal ROM Enabled (IRE)
- On Board Development (OBD)
- Development (DEV)
- Programing (PROG)

See Section 2.3 on page 53 for more information about these pins and controlling the loads connected to them.

Code written for IRE environment is executable in all environments, since it is binary compatible. The execution time of code in on-chip Base Memory (in IRE environment) is identical to that in OBD and DEV environments; i.e., the operation is cycle-by-cycle compatible.

PC87591x devices are tested to ensure that they operate in either IRE or OBD environment. Only selected parts are tested for operation in DEV environment.

1.4.1 IRE Environment

IRE environment is used while the PC87591x operates in the production system and executes the application. The on-chip flash (ROM in PC87591L) is the main source of code for the device. In this environment, after reset, the PC87591x starts running the code written in the internal flash first address, which is part of the flash's core boot block. In PC87591E and PC87591S the core boot-code size may vary from 4 Kbytes to128 Kbytes, as defined in the protection word. To prevent execution from non-programed flash (typically due to interrupted update), the code in the boot sector should verify the validity of the information in other parts of the flash. Once this is successfully done, the core may start executing from these parts as well. The External Memory, if available, can also be used to store code and/or data.

The PC87591L is shipped with 4 Kbytes of on-chip boot code. The user is expected to use an external memory for most of the code and constant data.

To maximize on-chip flash (ROM) performance, configure the BIU as described in Section 4.1.11 on page 86 and Section 4.16.2 on page 219.

1.0 Introduction (Continued)

See Figure 1 on page 24 for a system example in IRE environment. In this environment, the ENV0, ENV1 and TRIS strap pins do not need any external pull-up resistors.

1.4.2 OBD Environment

OBD environment is used for debugging the PC87591x firmware while it is mounted on its final production board. All pins have their IRE functionality, and the interface to a debugger running on the host is enabled using the JTAG based debugger interface. In OBD Environment, code is executed from the on-chip flash memory. Breakpoints on data and code access may be applied using the core hardware breakpoint mechanism. The monitor stored in the on-chip flash is used as part of the debugging environment. Erasing and reprograming the flash is enabled through the JTAG-based debugger interface.

OBD environment is binary and cycle-by-cycle compatible with IRE environment.

See Figure 3 on page 31 for a system example in OBD environment. In this environment, the ENV0 and TRIS strap pins are left unconnected, and ENV1 requires an external pull-up resistor. If the flash Contents Protect bit in the protection word is cleared (0), the PC87591x remains reset until the bit is set; see Section 4.16.6 on page 226.

1.4.3 DEV Environment

DEV environment is used in Application Development Boards (ADBs) or In System Emulators (ISEs). It may only be used when the chip is packed in its 176-pin package. In this mode, the on-chip flash is replaced with off-chip SRAM memory to allow flexible and fast development of application code. Some pins are allocated for development system use, and the GPIO functions associated with the pins are replicated using off-chip logic as part of the ADB system. DEV environment is binary and cycle-by-cycle compatible with OBD and IRE environments.

In this environment, the pins of ports PH, PI, PJ, PK, PL and PM are allocated for the interface to the off-chip Base Memory, core status signals, reset output and a breakpoint input. The system may regain these ports using the I/O Expansion protocol and off-chip logic, while maintaining cycle-by-cycle and binary compatibility with IRE and OBD environments. Using the same software, this environment is binary and cycle-by-cycle compatible with IRE and OBD environments. All features of IRE environment can be implemented either directly or by using additional external logic.

See Figure 4 on page 32 for a system example in DEV environment. In this environment, the ENV0 strap pin needs an external pull-up resistor and the ENV1 and TRIS pins are left unconnected. If the flash Contents Protect bit in the protection word is cleared (0), the PC87591x remains reset until the bit is set; see Section 4.16.6 on page 226.

1.4.4 PROG Environment

PROG environment is used to program the on-chip flash. It can be used to program the on-chip flash with an external flash programer. In PROG environment, the PC87591x remains reset indefinitely, I/O pins are allocated for the on-chip flash interface ports and the on-chip flash interface is accessible directly through the I/O pins.

In this environment, both ENV0 and ENV1 strap pins need an external pull-up resistor. The TRIS pin can be either left unconnected or connected to a pull-down resistor.

1.5 MEMORY MAP

The PC87591x has two address domains: core and host. The host address space is composed of the host I/O address space and the host memory space. Section 1.5.1 discusses the mapping of memories and peripherals into the core address space. Figure 5 illustrates the memory map and the shared access schemes that are possible. Section 1.5.2 discusses the mapping of the host address space and ways of accessing it. The PC87591x enables several memories in the core address space to have restricted access to the host. These memories are referred to as “shared memory” or “shared BIOS”. In addition, the core can access the Host Controlled Functions. Section 1.5.2 and Section 1.5.3 discuss these instances of cross-domain access, respectively.

1.0 Introduction (Continued)

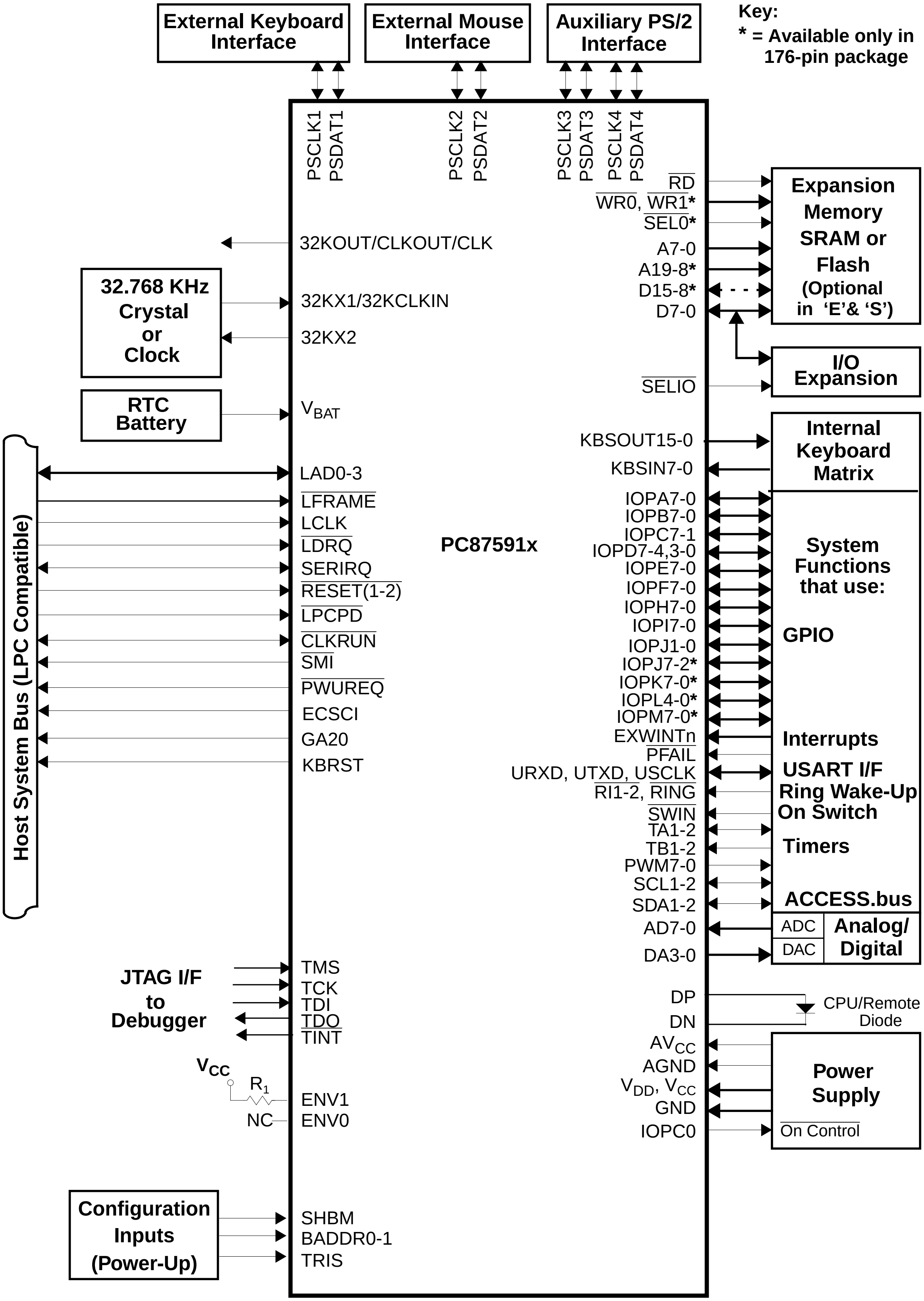


Figure 3. OBD Environment PC87591x System Connection Diagram, 128-Pin and 176-Pin Packages

1.0 Introduction (Continued)

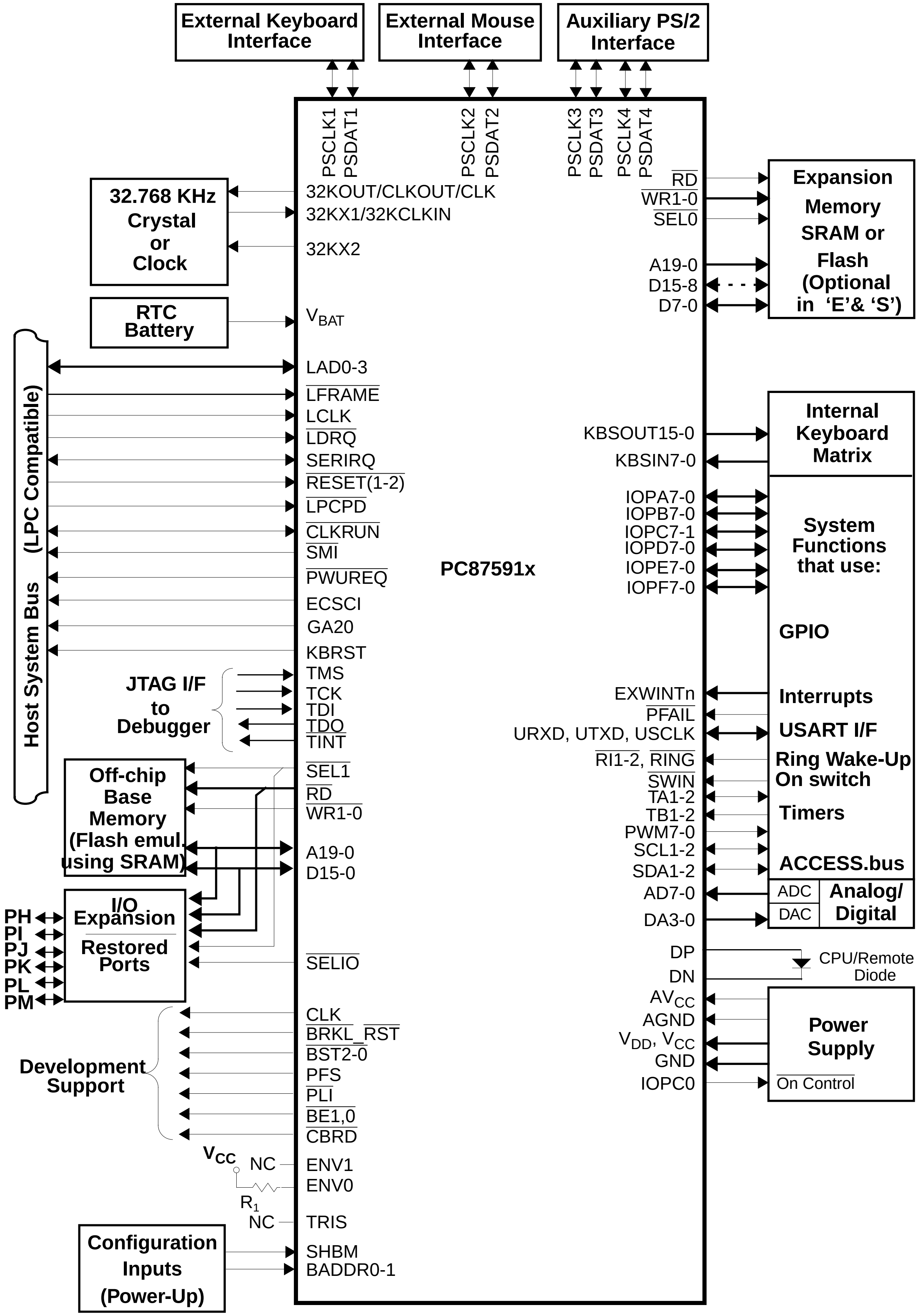


Figure 4. DEV Environment PC87591x System Connection Diagram,176-Pin Package

1.0 Introduction (Continued)

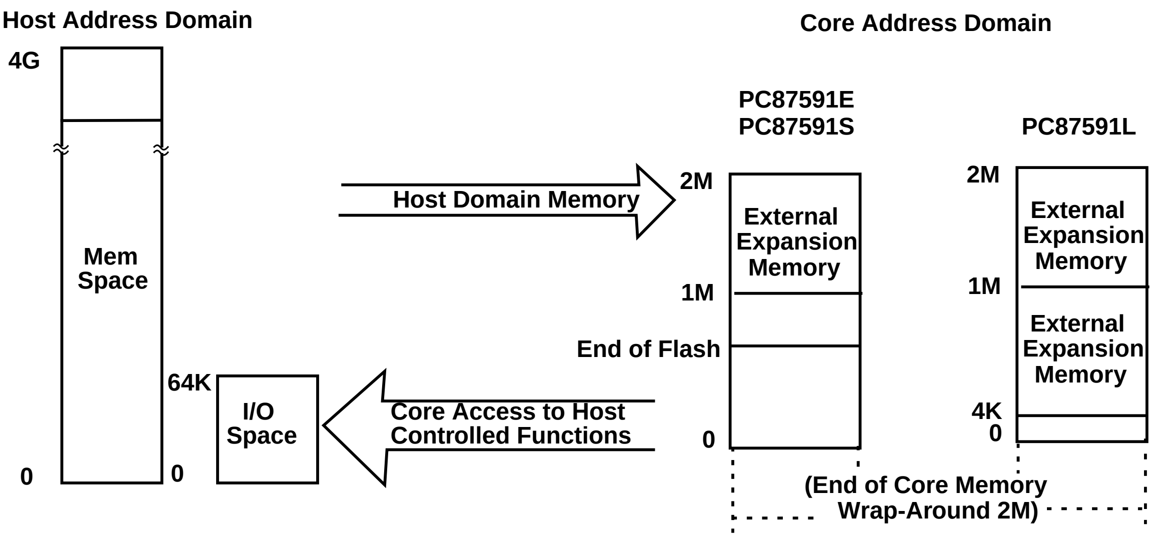


Figure 5. PC87591x Memory Domains

1.5.1 Core Address Domain Memory Map

The memory and I/O devices are directly mapped into the 2 Mbyte address space of the core. The core address space can include both code and data. However, access to data stored in the first 64 Kbytes of the address space is more efficient.

Most of the core code and constant data (including its boot section) is stored in the Base Memory. This memory is:

- On-chip flash in IRE and OBD environments
- Off-chip memories (SRAM or flash memory) in DEV environment

External memory may be added to the Base Memory to expand the amount of memory available for the application. The on-chip RAM and various peripherals are also mapped into the core address space.

Tables 1 to 3 show how the PC87591x memory and I/O devices are mapped in the core address space. Appendix A on page 408 shows the address map of the registers for the other modules.

Addresses not included in the following three tables or Appendix A are reserved. Attempts to access reserved addresses produce unpredictable results.

Table 1. PC87591S Memory Map

Address	Size (Bytes)	Description	
		Purpose	Environment
00 0000$_{16}$ – 00 DFFF$_{16}$	56K	Base Memory	IRE & OBD - Internal Flash DEV - External Base Memory
00 E800$_{16}$ – 00 F7FF$_{16}$	4K	System RAM	
00 F880$_{16}$ – 00 F8FF$_{16}$		Flash Control Registers[1]	
00 F900$_{16}$ – 00 F90A$_{16}$	11	Shared BIOS and Security Registers[1]	
00 F980$_{16}$ – 00 F98F$_{16}$	16	BIU Registers[1]	
00 FA00$_{16}$ – 00 FA7F$_{16}$		DMA Controller Registers1[1]	
00 FB00$_{16}$ – 00 FBFF$_{16}$	256	I/O Expansion	
00 FC00$_{16}$ – 00 FFFF$_{16}$	1K	On-chip Module Registers[1]	
01 0000$_{16}$ – 01 FFFF$_{16}$	64K	Base Memory (cont.)	IRE & OBD - Internal flash
			DEV - External Base Memory
02 0000$_{16}$ – 0F FFFF$_{16}$	896K	Reserved	IRE & OBD
		Base Memory	DEV - External Base Memory
10 0000$_{16}$ – 1F FFFF$_{16}$	1024K	Expansion Memory[2]	

[1] See Appendix A on page 408 for details of the implemented registers.
[2] Available in 176-pin package only.

1.0 Introduction (Continued)

Table 2. PC87591E Memory Map

Address	Size (Bytes)	Description	
		Purpose	Environment
00 0000$_{16}$ – 00 DFFF$_{16}$	56K	Base Memory	IRE & OBD - Internal Flash DEV - External Base Memory
00 F000$_{16}$ – 00 F7FF$_{16}$	2K	System RAM	
00 F880$_{16}$ – 00 F8FF$_{16}$		Flash Control Registers[1]	
00 F900$_{16}$ – 00 F90A$_{16}$	11	Shared BIOS and Security Registers[1]	
00 F980$_{16}$ – 00 F98F$_{16}$	16	BIU Registers[1]	
00 FA00$_{16}$ – 00 FA7F$_{16}$		DMA Controller Registers1[1]	
00 FB00$_{16}$ – 00 FBFF$_{16}$	256	I/O Expansion	
00 FC00$_{16}$ – 00 FFFF$_{16}$	1K	On-chip Module Registers[1]	
01 0000$_{16}$ – 0F FFFF$_{16}$	960K	Reserved	IRE & OBD
		Base Memory	DEV - External Base Memory
10 0000$_{16}$ – 1F FFFF$_{16}$	1024K	Expansion Memory[2]	

1. See Appendix A on page 408 for details of the implemented registers.
2. Available in 176-pin package only.

Table 3. PC87591L Memory Map

Address	Size (Bytes)	Description	
		Purpose	Environment
00 0000$_{16}$ – 00 0FFF$_{16}$	4K	Base Memory	IRE & OBD - Internal Flash DEV - External Base Memory
00 1000$_{16}$ – 00 DFFF$_{16}$	52K	Expansion Memory [1]	
00 F000$_{16}$ – 00 F7FF$_{16}$	2K	System RAM	
00 F880$_{16}$ – 00 F8FF$_{16}$		Flash Control Registers[1]	
00 F900$_{16}$ – 00 F90A$_{16}$	11	Shared BIOS and Security Registers[2]	
00 F980$_{16}$ – 00 F98F$_{16}$	16	BIU Registers[1]	
00 FA00$_{16}$ – 00 FA7F$_{16}$		DMA Controller Registers1[1]	
00 FB00$_{16}$ – 00 FBFF$_{16}$	256	I/O Expansion	
00 FC00$_{16}$ – 00 FFFF$_{16}$	1K	On-chip Module Registers[1]	
01 0000$_{16}$ – 0F FFFF$_{16}$	960K	Expansion Memory	
10 0000$_{16}$ – 1F FFFF$_{16}$	1024K	Expansion Memory	

1. The Expansion Memory size is limited to 1 Mbyte and is aliased between addresses above and below 01 0000$_{16}$.
2. See Appendix A on page 408 for details of the implemented registers.

Register Abbreviations and Access

The following abbreviations are used to indicate the Register Type:

- R/W= Read/Write
- R= The Read portion of a register, where a read from a specific address returns the value of a specific register; a write to the same address is to a different register.
- W= The Write portion of a register as described above for ‘R’.
- RO= Read Only
- WO= Write Only
- R/W1C= Read/Write 1 to Clear. Writing 1 to a bit clears it to 0. Writing 0 has no effect.

1.0 Introduction (Continued)

Either byte-wide or word-wide transactions to any address within the memory address space may be used to access memory devices.

Only byte-wide transactions may be used to access byte-wide registers, and only word-wide transactions may be used to access word-wide registers. Attempts to read a write-only register or write to a read-only register cause unpredictable results.

Zeros must be written to reserved bits unless stated otherwise. Reading reserved bits returns an undefined value. When modifying a register with reserved bits, the data read from reserved a bit can be written back to it.

Accessing Base Memory

In the PC87591E and PC87591S the Base memory is the main storage for code and data. When Shared BIOS is enabled, the Base memory may also be used for storing BIOS code and data. Figure 6 on page 35 illustrates how on-chip and off-chip Base Memory are mapped to the PC87591E and PC87591S address space.

The Base Memory is divided into two zones:

- Fast Zone - for addresses with Base Memory in the range of 00 0000 to 01 FFFF. The access time to this zone is controlled by the BIU Zone 1 configuration registers.
- Slow Zone - for addresses with Base Memory in the range of 02 0000 to 0F FFFF. The access time to this zone is controlled by the BIU Zone 2 configuration registers.

In the PC87591L, the base memory is used for storing code and data required for basic boot operations. The rest of the code and data are stored in the Expansion Memory shared by the core firmware and host BIOS. Figure 7 on page 36 illustrates how on-chip and off-chip Base Memory are mapped to the PC87591L address space.

IRE and OBD Environments - On-Chip Base Memory. In IRE and OBD environments, the on-chip flash (ROM) is used as Base memory. Note that the size of the available memory may differ among members of PC87591x family (see Tables 1 to 3). The access time to it is controlled by BIU Zones 1 and 2. To allow cycle-by-cycle compatibility with DEV environment, both zones should be programed in the same way for all environments. Thus to maximize on-chip flash (ROM) performance, configure BIU Zones 1 and 2, as described in Section 4.1.11 on page 86 and Section 4.16 on page 218, with minimal access time supported at the frequency in use, as defined in “Flash Read, Write and Erase Time” on page 222.

Off-Chip Base Memory. In DEV environment (when on-chip flash is disabled), the code and constant data are stored in off-chip Base Memory. In the PC87591E and PC87591S, this off-chip Base Memory has 1016 Kbytes of address space (00 0000_{16} to 00 DFFF_{16} and 01 0000_{16} to 0F FFFF_{16}). In the PC87591L, the size of the off-chip Base Memory is 4 Kbytes.

To ensure that the code fits into the on-chip flash in IRE and OBD environments later on, both the code and constant data should fit within the defined flash size, for the selected member of the PC87591x family.

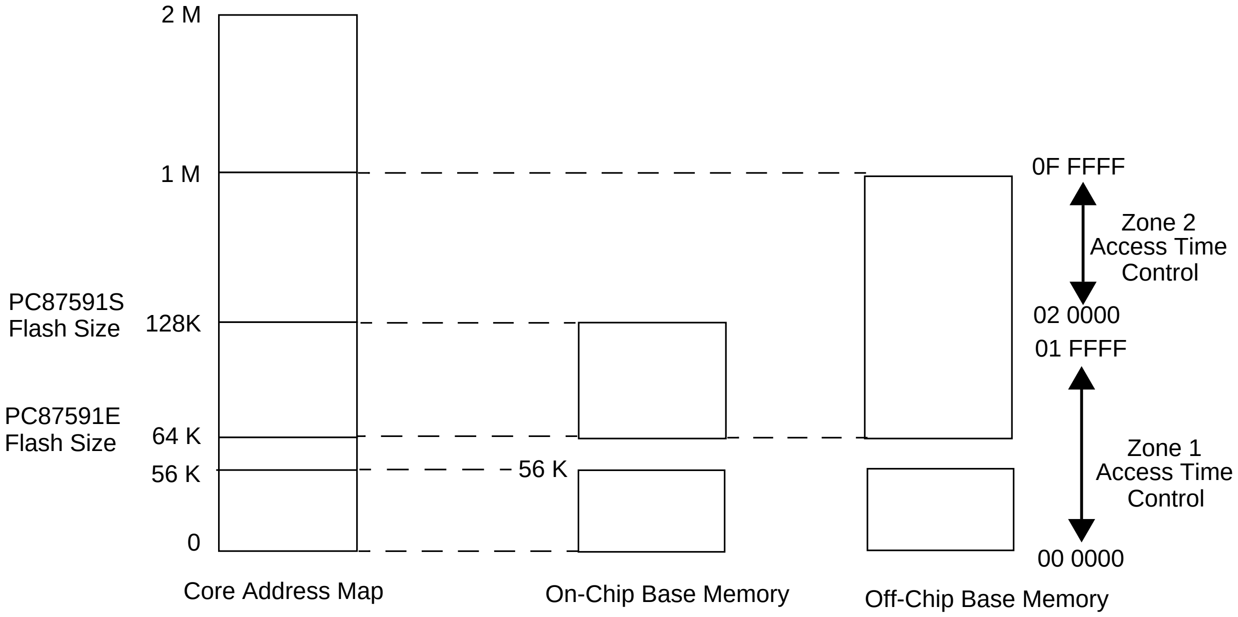


Figure 6. Base Memory Address Mapping - PC87591E and PC87591S

1.0 Introduction (Continued)

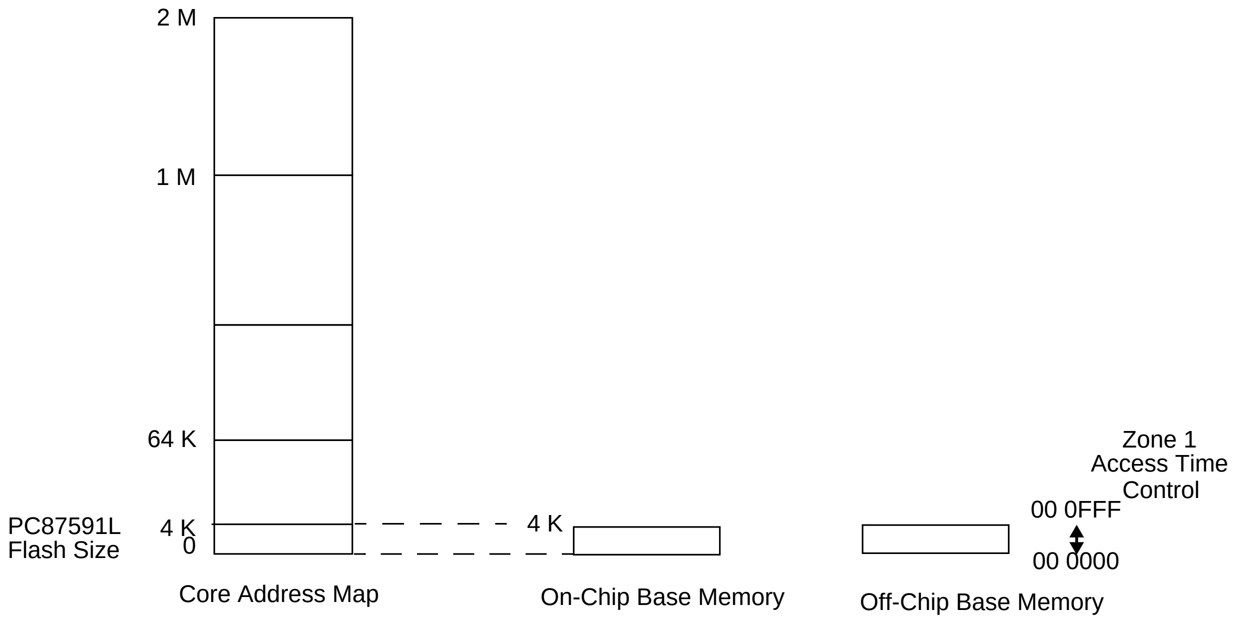


Figure 7. Base Memory Address Mapping - PC87591L

Accessing Expansion Memory

Expansion Memory Configuration is available in PC87591x versions packed in 176-pin packages. Access to it is enabled only after:

- The pins used for the memory interface are configured to operate as Expansion Memory interface signals (see Section 2.4 on page 54 for details on the alternate functions configuration).
- The BIU Zone 0 Configuration register (SZCFG0), which controls the memory access parameters (e.g., bus width and access time), is set to support the configuration in use.

The interface signals to the Expansion Memory are:

- 8-bit flash or SRAM: $\overline{\text{SEL0}}$, $\overline{\text{RD}}$, $\overline{\text{WR0}}$, D0-7 and A0-19 (fewer address lines may be used with a smaller flash).
- 16-bit flash or SRAM: $\overline{\text{SEL0}}$, $\overline{\text{RD}}$, $\overline{\text{WR0-1}}$, D0-15 and A1-19 (fewer address lines may be used with a smaller flash).

Figure 8 shows the External Memory Address Range mapping to the core, in the PC87591E and PC87591S devices. Figure 9 shows the External Memory Address Range mapping to the core in the PC87591L device. Note that in PC87591L devices, the External Memory is aliased to low addresses of the core address space, enabling access to memory locations in the flash from both an address in the low memory and an address in the upper part of the memory.

1.0 Introduction (Continued)

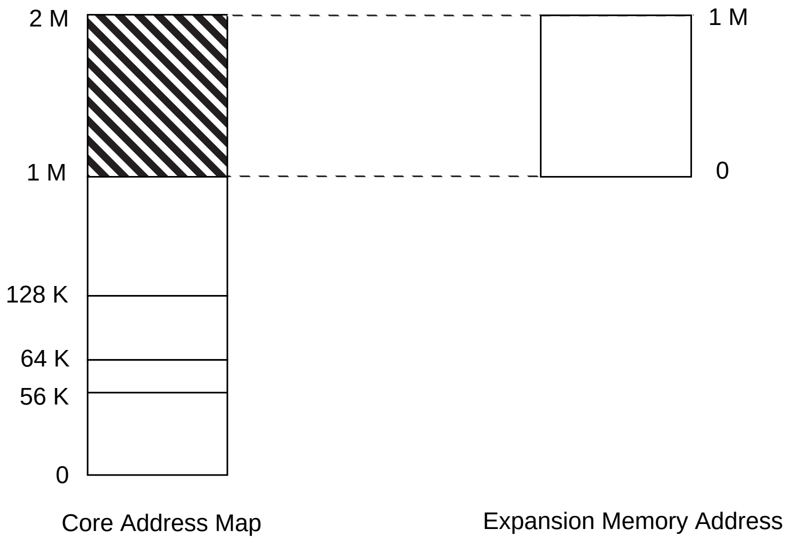


Figure 8. Expansion Memory (Zone 0) Address Range - PC87591E and PC87591S

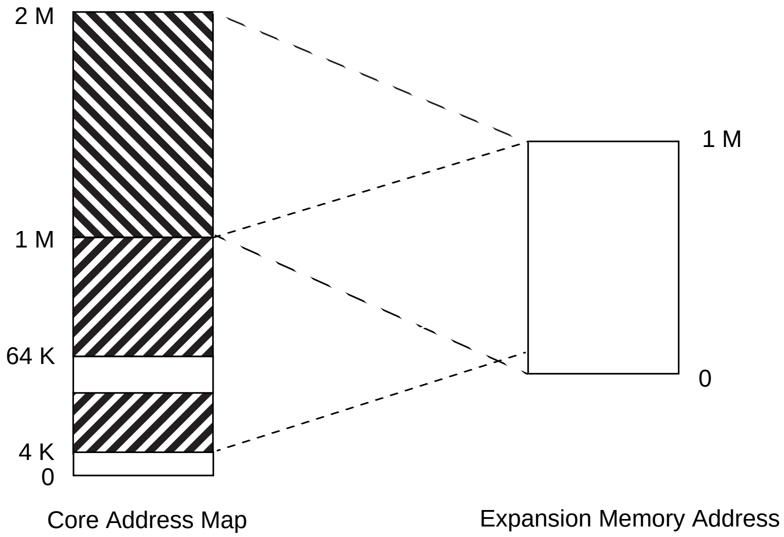


Figure 9. Expansion Memory (Zone 0) Address Range - PC87591L

External Memory Mapping into Shared BIOS Memory

When the shared BIOS memory is enabled using the SHBM strap input (SHBM=1) or the Shared Memory Configuration Registers (LDN=10h), the Expansion Memory address range is mapped into the address range of the host. The PC87591x uses the wrap-around effect of the core address space (on a 2 Mbyte boundary) using “growing down” addresses, which enables the host to view the Expansion flash as a continuation of the BIOS Memory. See Section 5.3.2 on page 297 for details of the memory mapping scheme.

Accessing I/O Expansion Space

The I/O expansion protocol enables implementing I/O devices or GPIO ports in the system, in addition to those available on-chip, for IRD, OBD and DEV environments. In addition, in DEV environment, some of the on-chip I/O port pins are used to interface with off-chip peripherals. In such a case, the I/O expansion protocol is used to implement the functionality of these I/O ports, using off-chip external logic. Access to these ports is through the same addresses used for the ports’ on-chip implementation.

The I/O expansion space is mapped to the address space 00 $FB00_{16}$ to 00 $FBFF_{16}$. This space is partitioned as follows:

- Addresses in the range 00 $FB00_{16}$ to 00 $FB22_{16}$ are used by GPIO ports PH, PI, PJ, PK, PL and PM (either on-chip or in their off-chip implementation, while the chip is in DEV environment).
- Address 00 $FBFE_{16}$ is used only in DEV environment by the MCFGSH register and must be written after each write to the MCFG with the same data written to the MCFG.
- Addresses in the range 00 $FBC0_{16}$ to 00 $FBFF_{16}$ are reserved for development board use.
- All other addresses may be used by the application for adding additional I/O elements.

The PC87591x accesses the off-chip I/O expansion using the I/O zone of the BIU. The zone select signal ($\overline{\text{SELIO}}$), address lines A0-7 and the $\overline{\text{RD}}$ and $\overline{\text{WR0}}$ signals are used to interface to the off-chip logic.

1.5.2 Host Address Domain Memory Map

The host address space includes memory space and I/O space.

The I/O space used by the PC87591x is configured through the PC87591x configuration registers. The configuration register address is defined by strap inputs (BADDR0-1) to be one of two fixed addresses or an address defined by the core using registers in the MSWC module.

When a Shared BIOS scheme is enabled via the SHBM strap input, the PC87591x is mapped to enable the host to boot from a shared flash device. The PC87591x supports either memory or FWH transactions for the host memory interface, with automatic selection between them (see Section 6.1.11 on page 349). Section 5.3 on page 297 discusses the mapping of memory between the host and core domains and the read and write access protection scheme from the host and core sides.

Following the boot process, the Shared Memory configuration registers (see Section 6.1.11 on page 349) enable setting memory sharing. The configuration setting includes defining the memory protocol in use (memory or FWH) and the address range used in the host address space. The configuration registers allow the defaults set by the SHBM strap input to be overridden; this enables using the shared memory for purposes other than system BIOS (e.g., PC87591x firmware update and secured storage of information).

1.5.3 Core Access to Host Controlled Peripherals

The core may access host domain devices through the Core to Host Controlled Functions access bridge. The bridge employs an indirect mapping scheme.

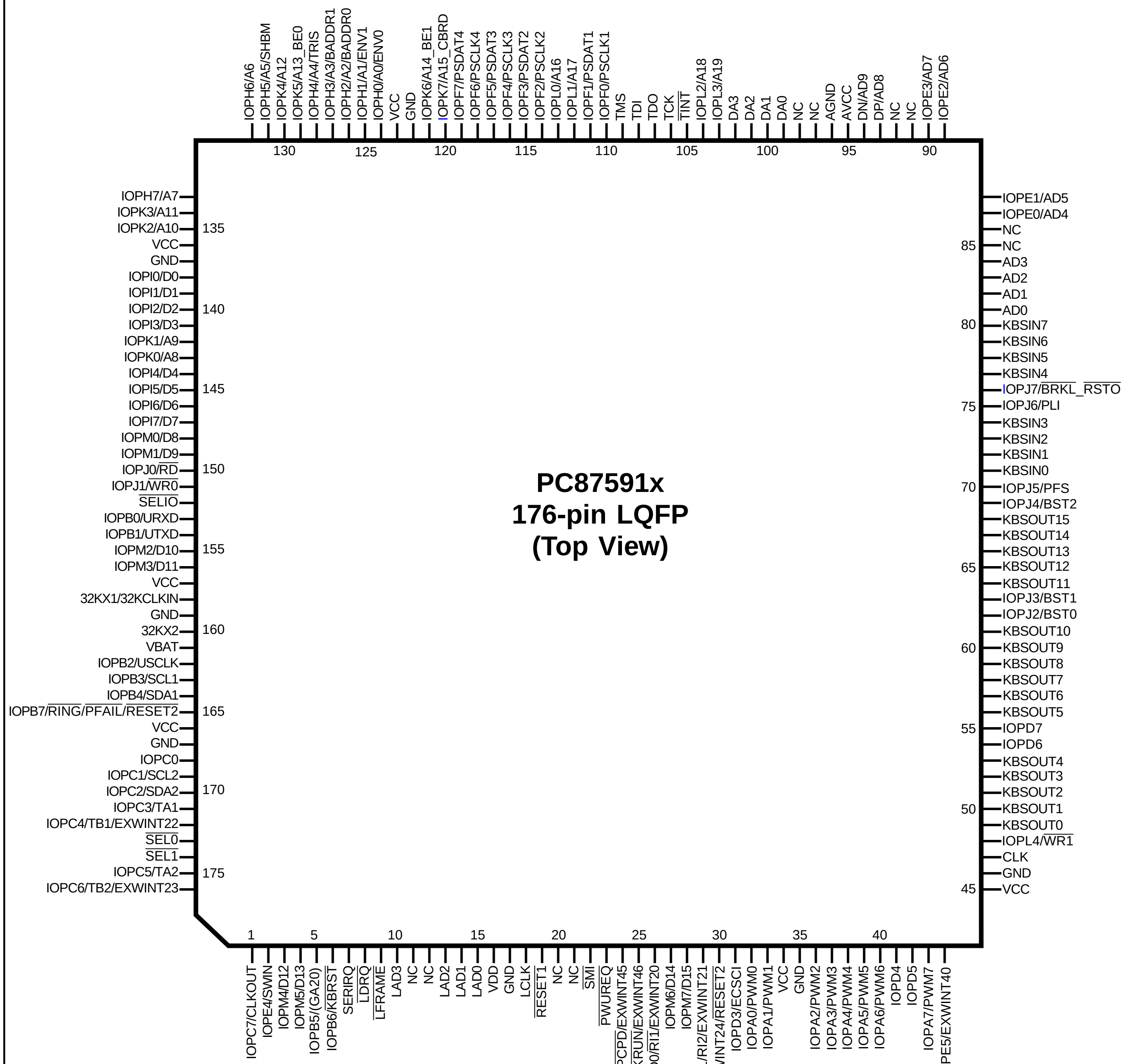
There is only a single set of peripheral registers for host and core use. The bus arbitration guarantees that only one of the two register accesses occurs at any given time, but this does not prevent problems that may be caused by conflicting write transactions. When such a case is expected, the core lock mechanism may be used to protect access to one or more of the devices. The lock may also be used to protect from the host from access to devices due to the security reasons.

The core accesses a register by specifying the logical device and the offset of the register within the logical device. Note that the configuration registers' index and data registers are also handled as a logical device. The core triggers a read by writing 1 to the read start bit and waits for the bit to clear; it can then read the data from the data register. The core triggers a write by performing a write operation to the data register.

Section 5.4 on page 313 provides details of the bridge and its operation.

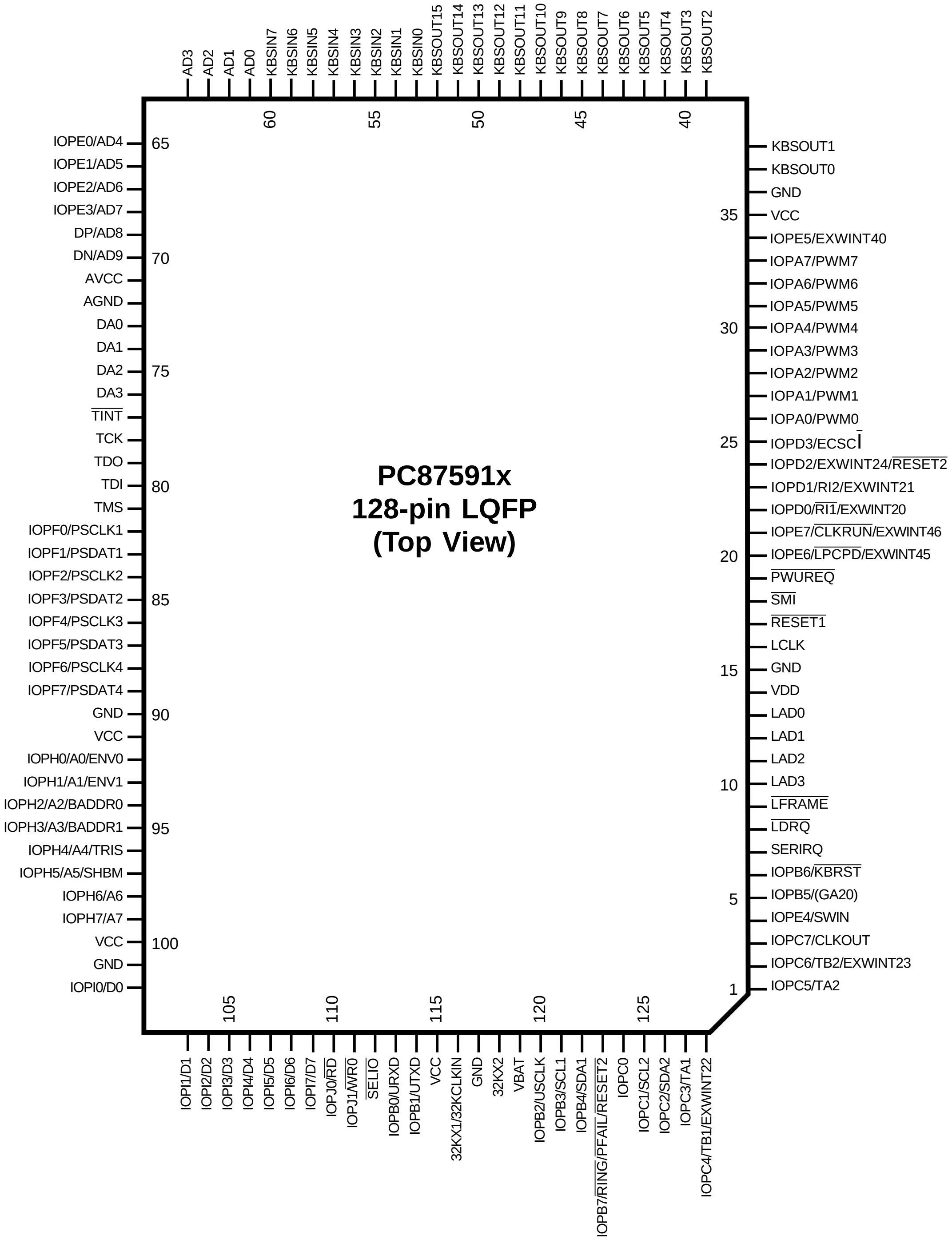
2.0 Signal/Pin Description and Configuration

2.1 CONNECTION DIAGRAMS



176-pin Low Profile Plastic Quad Flatpack (LQFP)
Order Number PC87591E-VPC / PC87591S-VPC / PC87591L-VPC
NS Package Number VPC176

2.0 Signal/Pin Description and Configuration (Continued)



128-pin LQFP Package
Order Number: PC87591E-VLB / PC87591S-VLB
NS Package Number VLB128

2.0 Signal/Pin Description and Configuration (Continued)

2.2 BUFFER TYPES AND SIGNAL/PIN DIRECTORY

The following sections contain detailed functional descriptions and electrical DC characteristics for the PC87591x signals. The pin multiplexing and function selection criteria are described in Section 2.4 on page 54. The signal DC characteristics are denoted by a buffer type symbol described briefly below and in further detail in Section 7.2 on page 374. The pin multiplexing information refers to two different types of multiplexing:

- MUX - Multiplexed, denoted by a slash (/) between pins in the diagram in Section 2.1 on page 39. Pins are shared between two different functions. Each function is associated with different board connectivity. Normally, the function selection is determined by the board design and can not be changed dynamically. The multiplexing options must be configured by the core in order to comply with the board implementation.
- MM - Multiple Mode, denoted by an underscore (_) between pins in the diagram in Section 2.1 on page 39. Pins have two or more modes of operation within the same function. These modes are associated with the same external (board) connectivity. Mode selection may be controlled by the device driver through the registers of the functional block and do not require a special setup. These pins are not considered multiplexed pins from the configuration perspective. The mode selection method as well as the signal specification in each mode are described in the functional description.

Table 4. Buffer Types

Symbol	Description
IN_{AC}	Input, analog to ACM
IN_{AD}	Input, analog to ADC
IN_{CS}	Input, CMOS compatible, with Schmitt Trigger
IN_{DI}	Input, analog from remote diode
IN_{OSC}	Input, from crystal oscillator (not characterized)
IN_{PCI}	Input, PCI 3.3V
IN_{SM}	Input, SMBus compatible
IN_{T}	Input, TTL compatible
IN_{TS}	Input, TTL compatible, with Schmitt Trigger
IN_{ULR}	Input, power, resistor protected (not characterized)
$O_{p/n}$	Output, push-pull output buffer that is capable of sourcing *p* mA and sinking *n* mA
OD_{n}	Output, open-drain output buffer that is capable of sinking *n* mA
O_{DA}	Output, analog from DAC
O_{DI}	Output, analog to remote diode
O_{OSC}	Output, to crystal oscillator (not characterized)
O_{PCI}	Output, PCI 3.3V
PWR	Power pin
GND	Ground pin

2.0 Signal/Pin Description and Configuration (Continued)

2.2.1 ACCESS.bus (ACB1 and ACB2) Interface

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
SCL2-1	125, 121	169, 163	I/O	IN_{SM}/OD_2	V_{CC}	**ACCESS.bus Serial Clock 1 and 2 Signals**. An internal pull-up for this pin is optional.
SDA2-1	126, 122	170, 164	I/O	IN_{SM}/OD_2	V_{CC}	**ACCESS.bus Serial Data 1 and 2 Signals**. An internal pull-up for this pin is optional.

2.2.2 Analog Interface

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
AD9-0	70-61	94-93, 90-87, 84-81	I	IN_{AD}	AV_{CC}	**Analog to Digital Converter Inputs.**
DA3-0	76-73	102-99	O	O_{DA}	AV_{CC}	**Digital to Analog Converter Outputs**.
DP	69	93	O	O_{DI}	AV_{CC}	**Diode Anode.** Remote diode current source. Connected to remote discrete diode.
DN	70	94	I	IN_{DI}	AV_{CC}	**Diode Cathode**. Remote diode current sink. Connected to remote discrete diode. Must be grounded when not used.

2.2.3 Clocks

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
32KCLKIN	116	158	I	IN_T	V_{PP}	**32.768 KHz Clock Input.**
32KX1	116	158	I	IN_{OSC}	V_{PP}	**32.768 KHz Crystal Oscillator Input.** Input from external crystal oscillator circuitry. See Figure 113 on page 356.
32KX2	118	160	O	O_{OSC}	V_{PP}	**32.768 KHz Crystal Oscillator Output.** Output to external crystal oscillator circuitry. See Figure 113 on page 356.
CLK	N/A	47	O	$O_{1/2}$	V_{CC}	**Clock Output.** PC87591x Core Domain system clock. Available in all environments, in the 176-pin package option.
CLKOUT	3	1	O	$O_{2/4}$	V_{CC}	**Clock Output.** This pin may output either the 32.768 KHz clock or the Core Domain system clock (CLK). CLKM bit in MCFG register selects the output clock frequency.

2.0 Signal/Pin Description and Configuration (Continued)

2.2.4 Core Bus Interface Unit (BIU)

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
A3-0	95-92	127-124	O	$O_{1/2}$	V_{CC}	**Address Bus, bits 0-19**. Core external address bus. Affected or used by any transaction on the internal core-bus, including DMA transactions and accesses to the internal memory.
A19-8	N/A	103-104 112-113 120-121 129-130 134-135 142-143				
A7-4	99-96	133-131, 128	O	$O_{2/12}$	V_{CC}	
D7-4, D3-0	109-102	147-144 141-138	I/O	$IN_T/O_{1/2}$	V_{CC}	**Data bus, bits 0-15**. Core external data bus. Used for core access to external memory or I/O devices. When accessing an 8-bit external memory or I/O device, only D0-7 should be used.
D15-14, D13-12, D11-10, D9-8	N/A	149-148 156-155 4-3 28-27				
RD	110	150	O	$O_{1/2}$	V_{CC}	**Read Control.** Core external read strobe. Can be used as output enable for external memory of I/O devices.
$\overline{\text{SEL0}}$	N/A	173	O	$O_{1/2}$	V_{CC}	**Zone Select 0.** Chip-select signal for external memory devices mapped to zone 0.
$\overline{\text{SEL1}}$	N/A	174	O	$O_{1/2}$	V_{CC}	**Zone Select 1.** Chip-select signal for external memory devices mapped to zones 1 and 2 (Base Memory). Available in DEV environment for off-chip emulation of the on-chip flash or ROM. In IRE and OBD environments, zones 1 and 2 are mapped to the on-chip flash or ROM and $\overline{\text{SEL1}}$ output is inactive.
$\overline{\text{SELIO}}$	112	152	O	$O_{1/2}$	V_{CC}	**I/O Zone Select.** Chip-select signal for external I/O devices mapped to this zone. Can be used for mapping off-chip I/O expansion devices to the core address space.
$\overline{\text{WR1}}$	N/A	48	O	$O_{1/2}$	V_{CC}	**Write Control 1 and 0.** Indicates writes to bytes 1 and 0, respectively, of the BIU data bus.
$\overline{\text{WR0}}$	111	151				

2.2.5 Development System Support

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
BE1-0	N/A	121,129	O	$O_{1/2}$	V_{CC}	**Byte Enable** bits 1 and 0 on monitor bus cycles.
$\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$	N/A	76	I	$IN_T/O_{1/2}$	V_{CC}	**Break Line and Reset Out.** When the Core bus is active, the pin is used as a $\overline{\text{BRKL}}$ input. It Indicates to the core that a breakpoint is needed once the currently fetched instruction goes into execution. When the core bus is not active, the signal is used as a $\overline{\text{RSTO}}$ output. It indicates to the system that an internal reset to the core and its peripherals occurred.
BST2-0	N/A	69, 63-62	O	$O_{1/2}$	V_{CC}	**Bus Status** bits 2-0 on monitor bus cycles. In DEV environment, these pins allow monitoring of the external bus cycles. When OBR bit in BCFG register is set, the internal bus cycles are also visible on the external bus. See also Table 41 on page 273.
CBRD	N/A	120	O	$O_{1/2}$	V_{CC}	**Core Bus Read Status** on monitor bus cycles. Available in all modes. See Section 4.1.9 on page 81.
PFS	N/A	70	O	$O_{1/2}$	V_{CC}	**Pipe Flow Status.**
PLI	N/A	75	O	$O_{1/2}$	V_{CC}	**Pipe Long Instruction.**
TCK	78	106	I	IN_{TS}	V_{CC}	**JTAG Test Clock.**
TDI	80	108	I	IN_T	V_{CC}	**JTAG Test Data In.**
TDO	79	107	O	$O_{1/2}$	V_{CC}	**JTAG Test Data Out.**
TINT	77	105	O	OD_2	V_{CC}	**JTAG Test Interrupt.**
TMS	81	109	I	IN_T	V_{CC}	**JTAG Test Mode Select.**

2.0 Signal/Pin Description and Configuration (Continued)

2.2.6 General-Purpose I/O (GPIO) and Internal Keyboard Scan

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
KBSIN7-0	60-53	80-77, 74-71	I	IN_{CS}/IN_{AC}	V_{CC}	**Keyboard Scan Inputs.** The input side of the internal keyboard scan lines. These inputs may be configured to work with either Schmitt trigger inputs, for operation with switch based keyboards, or with internal analog voltage comparators to interface to a resistive keyboard matrix. KBSIN7-0 are referenced to V_{CC} and are designed to work using a 3.3V supply. See Section 4.14 on page 208 for more details.
KBSOUT15-0	52-37	68-64, 61-56, 53-49	O	OD_6	V_{CC}	**Keyboard Scan Outputs.** The output side of the internal keyboard scan lines.

2.0 Signal/Pin Description and Configuration (Continued)

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
IOPA7-0	33-26	43, 40-36, 33-32	I/O	$IN_{TS}/O_{3/6}$	V_{CC}	**General-Purpose I/O Ports.** The GPIO registers are accessible by the core for read, write and configuration. Each of these GPIO signals can be individually configured to be input or output. The pins may be used as GPIO or assigned to their respective alternate functions. See Section 2.4 on page 54 for assignment of GPIO pins to alternate functions. See Section 4.5 on page 113 for further details on the GPIO pins and their functionality.
IOPD1-0	23-22	29, 26				
IOPB7-3	123, 6-5, 122-121	165, 6-5, 164-163	I/O	$IN_{TS}/O_{1/2}$	V_{CC}	
IOPC6-1	2-1, 128-125	176-175 172-169				
IOPH3-0	95-92	127-124				
IOPI7-0	109-102	147-144 141-138				
IOPJ7-2	N/A	76-75 70-71 63-62				
IOPJ1-0	111-110	151-150				
IOPK7-0	N/A	120-121 129-130 134-135 142-143				
IOPL4-0	N/A	48, 103-104 112-113				
IOPM7-0	N/A	28-27, 4-3, 156-155 149-148				
IOPB2-0	120, 114-113	162, 154-153	I/O	$IN_{TS}/O_{2/4}$	V_{CC}	
IOPC7	3	1				
IOPD7-4	N/A	55-54, 42-41	I/O	$IN_{TS}/O_{2/12}$	V_{CC}	
IOPD3-2	25-24	31-30				
IOPF7-0	89-82	119-114 111-110				
IOPH7-4	99-96	133-131 128				
IOPC0	124	168	O	$O_{3/6}$	V_{CC}	**General-Purpose Output Port IOPC0.** IOPC0 is are targeted for use as power supply control for the power supply unit. It is accessible by the core for write and configuration. On V_{CC} Power-Up reset, the default state TRI-STATE; it is not affected by other types of reset. See Section 4.5 on page 113 for further details on these GPIO pins and their functionality.

2.0 Signal/Pin Description and Configuration (Continued)

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
IOPE7-0	21-20, 34, 4, 68-65	25-24, 44, 2, 90-87	I	IN_{TS}	V_{CC}	**General-Purpose Input Port.** These pins serve as input-only pins. The GPIO registers are accessible by the core for read and configuration. The pins may be used as GPIO or assigned to their respective alternate functions. IOPE0-3 and IOPE5 do not have an internal pull-up resistor option. Note that IOPE0-3 and IOPE6-7 are not 5V tolerant. See Section 2.4 on page 54 for GPIO pin assignment to alternate functions. See Section 4.5 on page 113 for further details on the GPIO pins and their functionality.

2.0 Signal/Pin Description and Configuration (Continued)

2.2.7 Host Interface

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
$\overline{\text{CLKRUN}}$	21	25	I/O	$\text{IN}_{\text{PCI}}/\text{OD}_6$	V_{DD}	**Clock Run.** Same as PCI $\overline{\text{CLKRUN}}$. When high, indicates that the LPC clock will be slowed down or stopped. In this case, the PC87591x may pull it down to request full speed of the clock.
GA20	5	5	O	$\text{O}_{1/2}$	V_{CC}	**Gate A20.** Implemented using IOPB5 port output. See **Section 5.5.4 on page 322** for signal operation and behavior when V_{DD} is down.
$\overline{\text{KBRST}}$	6	6	O	$\text{O}_{1/2}$	V_{CC}	**Keyboard Reset Output.** See **Section 5.5.4 on page 322** for signal operation and behavior when V_{DD} is down.
LAD0-3	10-13	15-13, 10	I/O	$\text{IN}_{\text{PCI}}/\text{O}_{\text{PCI}}$	V_{DD}	**LPC Address-Data.** Multiplexed command, address bidirectional data and cycle status.
LCLK	16	18	I	IN_{PCI}	V_{DD}	**LPC Clock.** Practically the PCI clock (up to 33 MHz).
$\overline{\text{ECSCI}}$	25	31	O	$\text{O}_{2/12}$	V_{CC}	**EC SCI.** Generates an Embedded Controller SCI interrupt to the chipset. This signal is typically connected to one of the chipset GPI inputs.
$\overline{\text{LDRQ}}$	8	8	O	O_{PCI}	V_{DD}	**LPC DMA Request**. Encoded Bus Master request for LPC I/F.
$\overline{\text{LFRAME}}$	9	9	I	IN_{PCI}	V_{DD}	**LPC Frame**. Low pulse indicates the beginning of a new LPC cycle or the termination of a broken cycle.
$\overline{\text{LPCPD}}$	20	24	I	IN_{PCI}	V_{CC}	**Power Down.** Indicates that power will be shut off on the LPC interface.
$\overline{\text{RESET1}}$	17	19	I	IN_{PCI}	V_{CC}	**Reset 1.** A falling edge on this signal starts a reset sequence of the PC87591x. For details, see details in Section 3.2 on page 65.
$\overline{\text{RESET2}}$	24/123	30/165	I	IN_{CS}	V_{DD}	**Reset 2.** A level low reset to the LPC interface and Host Controlled Functions configuration registers and Shared Memory host registers. For details, see Section 3.2 on page 65. $\overline{\text{RESET2}}$ is either assigned to one of two optional pins or it is disabled, as defined in the protection words stored in the flash memory. See Section 2.4.1 on page 54 and Section 4.16.6 on page 226 for the configuration of this signal.
$\overline{\text{PWUREQ}}$	19	23	O	$\text{O}_{1/2}$	V_{CC}	**Power-Up Request.**
SERIRQ	7	7	I/O	$\text{IN}_{\text{PCI}}/\text{O}_{\text{PCI}}$	V_{DD}	**Serial IRQ.** The interrupt requests are serialized over a single pin, where each IRQ level is delivered during a designated time slot.
$\overline{\text{SMI}}$	18	22	I/O	$\text{IN}_{\text{TS}}/\text{OD}_{12}$	V_{CC}	**System Management Interrupt.**

2.0 Signal/Pin Description and Configuration (Continued)

2.2.8 Interrupt and Wake-Up Inputs (ICU and MIWU)

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
EXWINT20 EXWINT21 EXWINT22 EXWINT23 EXWINT24 EXWINT40 EXWINT45 EXWINT46	22 23 128 2 24 34 20 21	26 29 172 176 30 44 24 25	I	IN_{TS}	V_{CC}	**External Wake-up/Interrupt Inputs.** Fed into the Multi-Input Wake-up Unit (MIWU) for wake-up or interrupt. Can be used as external source for wake-up or interrupt. For details on interrupt and wake-up event assignment, see Table 17 on page 105.
$\overline{\text{PFAIL}}$	123	165	I	IN_{CS}	V_{CC}	**Power Fail.** Non-maskable external interrupt source. This assigned interrupt is non-maskable only after being enabled by software after reset.
SWIN	4	2	I	IN_{CS}	V_{CC}	**Power Switch Input.** Indicates a user request to turn the power on or off. This signal is connected to the Multi-Input Wake-up Unit (MIWU) for wake-up of the core domain and interrupt generation to the core for handling. For details on interrupt and wake-up event assignment, see Table 17 on page 105.

2.2.9 Power and Ground

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
AGND	72	96	I	GND		**Analog Ground.** Used as ground for the Analog to Digital Converter (ADC) and the Digital to Analog Converter (DAC).
AV_{CC}	71	95	I	PWR		**Analog 3.3V Power** supply. Used as power supply for the Analog to Digital Converter (ADC) and the Digital to Analog Converter (DAC).
GND	15, 36, 90, 101, 117	17, 35, 46,122, 137, 159,167	I	GND		**Ground.** Serves for both on-chip logic, output drivers and back-up battery circuit. See “Power Connection and Layout Guidelines” on page 63 for details on connections with AGND.
V_{BAT}	119	161	I	IN_{ULR}		**Battery Power Supply.** Provides battery backup to the Mobile System Wake-Up Control registers, to the RTC and to the 32 KHz crystal oscillator when V_{CC} is lost. The pin is connected to the internal logic through a series resistor for UL protection.
V_{CC}	35, 91,100, 115	34, 45, 123,136, 157,166	I	PWR		**Standby Digital 3.3V Power Supply.** Used for all core-domain digital logic and all associated I/O pins.
V_{DD}	14	16	I	PWR		**Digital 3.3V Power Supply.** Serves as power supply for the LPC interface and some of the host-controlled functions.

2.0 Signal/Pin Description and Configuration (Continued)

2.2.10 PS/2 Interface

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
PSCLK4-1	88, 86, 84, 82	118, 116, 114, 110	I/O	$IN_T/O_{2/12}$	V_{CC}	PS/2 Channel 1 through 4 Clock signal.
PSDAT4-1	89, 87, 85, 83	119, 117, 115, 111	I/O	$IN_T/O_{2/12}$	V_{CC}	PS/2 Channel 1 through 4 Data signal.

2.2.11 Strap Configuration

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
BADDR1-0	95-94	127-126	I	IN_{CS}	V_{CC}	**I/O Base Address.** Sampled at V_{CC} Power-Up reset to determine the base address of the configuration Index-Data register pair as follows: No pull-up resistor: $2E_{16}$-$2F_{16}$ 10 KΩ external pull-up resistor on BADDR0:$4E_{16}$-$4F_{16}$ 10 KΩ external pull-up resistor on BADDR1:Core defined
ENV1-0	93-92	125-124	I	IN_{CS}	V_{CC}	**Environment Select 1 and 0.** Sampled at V_{CC} Power-Up reset to determine the device operation environment, either IRE, OBD or DEV. Each pin is pulled to 0 (set environment to IRE = 00) by an internal resistor or set to 1 by an external 10 KΩ pull-up resistor. For further details refer to Section 2.3 on page 53.
SHBM	97	131	I	IN_{CS}	V_{CC}	**Shared Host BIOS Memory.** Sampled at V_{CC} Power-Up reset to determine the state of the shared Host BIOS memory. Pulled to 0 (disables the shared host BIOS memory) by an internal resistor or set to 1 by an external 10 KΩ pull-up resistor to enable the shared BIOS mode.
TRIS	96	128	I	IN_{CS}	V_{CC}	**TRI-STATE.** Forces the device to float all its output and I/O pins (except for DAC outputs and 32KX2) if an external pull-up resistor is connected. Sampled at V_{CC} Power-Up reset.

2.2.12 Mobile System Wake-Up Control (MSWC)

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
$\overline{\text{RING}}$	123	165	I	IN_{CS}	V_{CC}	**Telephone Line Ring.** Detection of a pulse train on this pin is a wake-up event that can activate the power-up request ($\overline{\text{PWUREQ}}$). The pin has a Schmitt Trigger input buffer, powered by V_{CC}.
$\overline{\text{RI1}}$	22	26	I	IN_{TS}	V_{CC}	**Ring Indicator.** When low, indicates that a telephone ring signal has been received by the modem. Ring signals are monitored during power-off for wake-up event detection.
RI2	23	29				

2.2.13 Timers and PWM

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
TA2	1	175	I/O	$IN_{T,}$ $O_{1/2}$	V_{CC}	**Timer Pin A for Timers 2 and 1.**
TA1	127	171				
TB2	2	176	I	IN_T	V_{CC}	**Timer Pin B for Timers 2 and 1.**
TB1	128	172				
PWM7-0	33-26	43, 40-36, 32-33	O	$O_{3/6}$	V_{CC}	**PWM Output 7-0.**

2.2.14 Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

Signal	128-Pin LQFP	176-Pin LQFP	I/O	Buffer Type	Power Well	Description
UTXD	114	154	O	$O_{2/4}$	V_{CC}	**USART Transmit Data.**
URXD	113	153	I	IN_T	V_{CC}	**USART Receive Data.**
USCLK	120	162	I/O	$IN_{CS}/O_{2/4}$	V_{CC}	**USART Serial Clock.** May be used as input or output when the USART is configured to operate in its synchronous mode of operation.

2.0 Signal/Pin Description and Configuration (Continued)

2.2.15 Internal Pull-Up and Pull-Down Resistors

The signals listed in Table 5 can optionally support internal pull-up (PU) and/or pull-down (PD) resistors. See Section 7.3 on page 377 for the values of each resistor type.

Table 5. Internal Pull-Up and Pull-Down Resistors

Signal	128-Pin LQFP	176-Pin LQFP	Type	Comments
ACCESS.bus (ACB)				
SCL2-1	125, 121	169, 163	PU$_{80}$	Programmable
SDA2-1	126, 122	170, 164	PU$_{80}$	Programmable
General-Purpose I/O (GPIO) Ports				
IOPA7-0	33-26	43, 40-36, 33-32	PU$_{80}$	Programmable
IOPB7-0	123, 6-5, 122,121,120, 114-113	165, 6-5, 164-163, 120,114-113	PU$_{80}$	Programmable
IOPC7-0	3-1,128-125	3,176-175, 172-169	PU$_{80}$	Programmable
IOPD7-4	N/A	55-54, 42-41	PU$_{80}$	Programmable
IOPD3-0	25-21	31-30, 29, 26		
IOPE7,6,4	21-20, 4	25-24, 2	PU$_{80}$	Programmable
IOPF7-0	89-82	119-114, 111-110	PU$_{80}$	Programmable
Internal Keyboard Scan (KBC)				
KBSIN7-0	60-53	80-77, 74-71	PU$_{80}$	Programmable
PS/2 Interface				
PSCLK4-1	88, 86, 84, 82	118,116, 114,110	PU$_{80}$	Programmable
PSDAT4-1	89, 87, 85, 83	119,117, 115,111	PU$_{80}$	Programmable
Strap Configuration				
BADDR1-0	95-94	127-126	PD$_{80}$	Strap[1]
ENV1-0	93-92	125-124	PD$_{80}$	Strap[1]
TRIS	96	128	PD$_{80}$	Strap[1]
SHBM	97	131	PD$_{80}$	Strap[1]

[1] Active only during V$_{CC}$ Power-Up reset.

2.3 STRAP PINS

During V_{CC} Power-Up reset, the ENV(0-1), TRIS, SHBM and BADDR strap input signals are sampled. Internal pull-down resistors set these signals to 0. These resistors are active only during V_{CC} Power-Up reset. An external 10 KΩ resistor connected to V_{CC} may be used to set them to 1.

Setting the Environment

ENV0 and ENV1 determine the operating environment. Table 6 shows the settings allowed. Pulling both ENV0 and ENV1 to 1 produces unpredictable results. In IRE and OBD environments, the TRIS strap input may be used for floating all the device signals. In other cases it should be kept low.

Table 6. Environment Pin Settings

Environment	ENV0	ENV1	TRIS
IRE	0	0	0[1]
OBD	0	1	0[1]
DEV	1	0	0
PROG	1	1	0

[1] When set to 1, the PC87591x is put in TRI-STATE mode.

Figures 1 on page 24, 3 on page 31 and 4 on page 32 demonstrate how to use the ENV0-1 signals to configure the PC87591x for IRE, OBD and DEV environments, respectively.

Other Strap Pin Settings

Table 7 provides brief descriptions of other strap inputs. For details on SHBM and TRIS, see Section 5.3 on page 297 and Section 4.20.4 on page 272, respectively.

Table 7. Other Strap Pin Settings

Strap Pin	Internal Pull-Down (0)	External Pull-Up (1)
BADDR1-0	SuperI/O Configuration Base Address see Table 47 on page 335	
SHBM	Disables shared memory with host BIOS	Enables shared memory with host BIOS
TRIS	Normal operation	While in IRE and OBD environments, causes PC87591x to float its output and I/O signals for system test purposes and clip-on ISE use

System Load on Strap Pins

The loads on the strap pins should not cause the voltage on them to drop below V_{IH} when the pins should be high (1), or to rise above V_{IL} when they should be low (0). See Section 7.3.2 on page 377.

If the load caused by the system on the strap pins exceeds 10 µA, use either an external pull-down resistor to keep the pin at 0 or a pull-up resistor with lower resistance to keep the pin at 1.

To reduce power consumption, in Idle mode, pins with strap inputs on them and a signal function other than GPIO are put in TRI-STATE or drive the strap-pin value. For pins with strap inputs that function as GPIO, it is recommended that the application drive the strap value as output to the value defined by the strap pin. For pins with strap and address line functionality, when the address configuration is enabled, the signal is driven by the hardware to its strap value on reset.

2.0 Signal/Pin Description and Configuration (Continued)

Strap Pin Status Register (STRPST)

The STRPST register is a byte-wide, read-only register. It enables the software to read the value set to strap pins during Power-Up reset. STRPST bits provide the value of their respective strap input. See Table 7 for bit details.

Location: 00 $FF12_{16}$

Type: RO

Reset Value: According to external straps

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**BADDR1**	**BADDR0**	**SHBM**
Reset								

2.4 ALTERNATE FUNCTIONS

2.4.1 GPIO with Alternate Functions

The PC87591x uses the GPIO port pins to multiplex functions, thereby maximizing the device's flexibility. Table 8 lists all the pins with such alternate functions and summarizes their selection criteria. To select alternate pin functions, strap inputs flash memory and configuration registers are used as follows:

- The ports' Alternate Function Control register controls the IOPA, IOPB, IOPC, IOPD, IOPE and IOPF pins. Each of the ports' pins may be used as a GPIO port or its alternate function, based on the setting of the corresponding bit in PxALT register.
- The environment setting and MCFG bits control port IOPH, IOPI, IOPJ, IOPK, IOPL and IOPM pins. In IRE and OBD environments, these pins may function either as GPIO or their alternate function (as indicated in the table). In DEV environment, they all function as their alternate functions.
- Interrupts may be enabled together with the use of the pin as input or with its alternate function when the alternate function is an input. Enabling the interrupt input is done via the port's alternate function or the respective bit in EICFG register.
- Strap inputs and their associated internal pull-down resistor function during V_{CC} Power-Up reset.
- The Protection Word in the flash configures the use of $\overline{\text{RESET2}}$ input.

When a pin is used as GPIO and not as its alternate function, disable the alternate function in the relevant module's register.

Note: In Table 8, the PKG column shows pins available in both 128- and 176-pin packages and pins available only in 176-pin packages.

Table 8. Alternate Function Selection

PKG	**GPIO**		**Alternate Function 1**			**Alternate Function 2**		
128/176	Name	Type	**Module**	**Signal Name**	**Select**	**Module**	**Signal Name**	**Select**
128/176	IOPA0	I/O	PWM	PWM0	PAALT.0			
128/176	IOPA1	I/O		PWM1	PAALT.1			
128/176	IOPA2	I/O		PWM2	PAALT.2			
128/176	IOPA3	I/O		PWM3	PAALT.3			
128/176	IOPA4	I/O		PWM4	PAALT.4			
128/176	IOPA5	I/O		PWM5	PAALT.5			
128/176	IOPA6	I/O		PWM6	PAALT.6			
128/176	IOPA7	I/O		PWM7	PAALT.7			
128/176	IOPB0	I/O	USART	URXD	PBALT.0			
128/176	IOPB1	I/O		UTXD	PBALT.1			
128/176	IOPB2	I/O		USCLK	PBALT.2			
128/176	IOPB3	I/O	ACCESS.bus	SCL1	PBALT.3			
128/176	IOPB4	I/O		SDA1	PBALT.4			

2.0 Signal/Pin Description and Configuration (Continued)

Table 8. Alternate Function Selection (Continued)

PKG	GPIO		Alternate Function 1			Alternate Function 2		
128/176	Name	Type	**Module**	**Signal Name**	**Select**	**Module**	**Signal Name**	**Select**
128/176	IOPB5	I/O	GA20 and $\overline{\text{KBRST}}$	(GA20)	PBALT.5=0[1]			
128/176	IOPB6	I/O		KBRST	PBALT.6			
128/176	IOPB7	I/O	MSWC	RING	PBALT.7	ICU	PFAIL/ RESET2[2]	PBALT.7/ REST2EN=01
128/176	IOPC1	I/O	ACCESS.bus	SCL2	PCALT.1			
128/176	IOPC2	I/O		SDA2	PCALT.2			
128/176	IOPC3	I/O	Timers	TA1	PCALT.3			
128/176	IOPC4	I/O		TB1	PCALT.4	MIWU	EXWINT22	PCALT.4
128/176	IOPC5	I/O		TA2	PCALT.5			
128/176	IOPC6	I/O		TB2	PCALT.6	MIWU	EXWNT23	PCALT.6
128/176	IOPC7	I/O	Clocks	CLKOUT	PCALT.7 & MCFG.CLKOM			
128/176	IOPD0	I/O	MSWC	RI1	PDALT.0	MIWU	EXWINT20	PDALT.0
128/176	IOPD1	I/O		RI2	PDALT.1		EXWINT21	PDALT.1
128/176	IOPD2	I/O		RESET2	RST2EN=10		EXWINT24	PDALT.2
128/176	IOPD3	I/O	Host I/F	$\overline{\text{ECSCI}}$	PDALT.3			
128/176	IOPE0	I	ADC Voltage	AD4	PEALT.0			
128/176	IOPE1	I		AD5	PEALT.1			
128/176	IOPE2	I		AD6	PEALT.2			
128/176	IOPE3	I		AD7	PEALT.3			
128/176	IOPE4	I	MIWU	SWIN	PEALT.4			
128/176	IOPE5	I				MIWU	EXWINT40	PEALT.5
128/176	IOPE6	I	LPC Interface	LPCPD	PEALT.6		EXWINT45	EICFG. EXWINT45
128/176	IOPE7	I		$\overline{\text{CLKRUN}}$	PEALT.7		EXWINT46	EICFG. EXWINT46
128/176	IOPF0	I/O	PS2 Interface	PSCLK1	PFALT.0			
128/176	IOPF1	I/O		PSDAT1	PFALT.1			
128/176	IOPF2	I/O		PSCLK2	PFALT.2			
128/176	IOPF3	I/O		PSDAT2	PFALT.3			
128/176	IOPF4	I/O		PSCLK3	PFALT.4			
128/176	IOPF5	I/O		PSDAT3	PFALT.5			
128/176	IOPF6	I/O		PSCLK4	PFALT.6			
128/176	IOPF7	I/O		PSDAT4	PFALT.7			

2.0 Signal/Pin Description and Configuration (Continued)

Table 8. Alternate Function Selection (Continued)

PKG	GPIO		Alternate Function 1			Alternate Function 2		
128/176	Name	Type	**Module**	**Signal Name**	**Select**	**Module**	**Signal Name**	**Select**
128/176	IOPH0	I/O	Core BIU	A0	DEV Env or MCFG. ENEIO=1 or MCFG. ENEMEM=1	Straps	ENV0	Power-up
128/176	IOPH1	I/O		A1			ENV1	
128/176	IOPH2	I/O		A2			BADDR0	
128/176	IOPH3	I/O		A3			BADDR1	
128/176	IOPH4	I/O		A4			TRIS	
128/176	IOPH5	I/O		A5			SHBM	
128/176	IOPH6	I/O		A6				
128/176	IOPH7	I/O		A7				
128/176	IOPI0	I/O		D0				
128/176	IOPI1	I/O		D1				
128/176	IOPI2	I/O		D2				
128/176	IOPI3	I/O		D3				
128/176	IOPI4	I/O		D4				
128/176	IOPI5	I/O		D5				
128/176	IOPI6	I/O		D6				
128/176	IOPI7	I/O		D7				
128/176	IOPJ0	I/O		RD				
128/176	IOPJ1	I/O		WR0				
176	IOPJ2	I/O	Development System Observability	BST0	DEV Env.			
176	IOPJ3	I/O		BST1				
176	IOPJ4	I/O		BST2				
176	IOPJ5	I/O		PFS				
176	IOPJ6	I/O		PLI				
176	IOPJ7	I/O		BRKL_RSTO				

2.0 Signal/Pin Description and Configuration (Continued)

Table 8. Alternate Function Selection (Continued)

PKG	**GPIO**		**Alternate Function 1**			**Alternate Function 2**		
128/176	Name	Type	**Module**	**Signal Name**	**Select**	**Module**	**Signal Name**	**Select**
176	IOPK0	I/O	Core BIU	A8	DEV Env. or MCFG. ENEMEM=1			
176	IOPK1	I/O		A9				
176	IOPK2	I/O		A10				
176	IOPK3	I/O		A11				
176	IOPK4	I/O		A12				
176	IOPK5	I/O		A13_BE0				
176	IOPK6	I/O		A14_BE1				
176	IOPK7	I/O		A15_CBRD				
176	IOPL0	I/O		A16				
176	IOPL1	I/O		A17				
176	IOPL2	I/O		A18				
176	IOPL3	I/O		A19				
176	IOPL4	I/O		WR1	DEV Env. or (MCFG. ENEMEM=1 and MCFG. EXMEM16=1)			
176	IOPM0	I/O		D8				
176	IOPM1	I/O		D9				
176	IOPM2	I/O		D10				
176	IOPM3	I/O		D11				
176	IOPM4	I/O		D12				
176	IOPM5	I/O		D13				
176	IOPM6	I/O		D14				
176	IOPM7	I/O		D15				
128/176			ADC Voltage	AD9	ADCCNF. REMDEN=0	ADC Temp	DN	ADCCNF. REMDEN=1
128/176				AD8			DP	

1. GA20 is implemented using the GPIO.
2. $\overline{\text{RESET2}}$ use is orthogonal to the other alternate functions and is determined by the setting in the on-chip flash Protection Word (see Section 4.16.6 on page 226), although it is recommended to use it in conjunction with an interrupt on the same pin. See “Using RESET2 Input” on page 67.

2.4.2 System Configuration Registers

For a summary of the abbreviations used for Register Type, see Section 2 on page 34.

Register Map

Mnemonic	Register Name	Type
MCFG	Module Configuration Register	R/W
EICFG	External Interrupts Configuration Register	R/W
IOEE1 and IOEE2	Input to Output Echo Enable Register 1 and 2	R/W

Module Configuration Register (MCFG)

The MCFG register is a read/write, byte-wide register. It is used for global system configuration and setup.

Write operations to the MCFG register should write zeros to all reserved bits. On reset, non-reserved bits of MCFG are cleared to 0. MCFG can be written in Active mode only. Its contents is preserved in Idle mode.

In IRE and OBD environments, all MCFG fields should be used to designate associated pins as GPIO ports or their alternate functions. In DEV environment, the pins are always allocated for development system use. The I/O ports functionality can be implemented using off-chip logic.

To guarantee binary and cycle-by-cycle compatibility among the different environments, define the MCFG fields as required for IRE and OBD even when in DEV environment, and use the I/O Expansion protocol to build an off-chip implementation of the I/O ports when they are used by the application.

ADBs or ISE systems use the MCFG Shadow (MCFGSH, write only) register to select the functionality of the signal that reaches the user’s application.

All write operations to the MCFG must be immediately followed by a write to the MCFG Shadow (MCFGSH) register. This register is part of the development system and is accessed by an access to the I/O Zone ($\overline{\text{SELIO}}$).

MCFG is loaded with either 00_{16} or 80_{16} on reset. See the description of GTMON for behavior of bit 7 during reset.

Software should load MCFGSH with the MCFG register’s value after reset.

The format of MCFG (and MCFGSH) is as follows:

MCFG Location: 00 $FF10_{16}$

MCFGSH Location: 00 $FBFE_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**GTMON**	**Reserved**		**CLKOM**		**EXMEM16**	**ENEMEM**	**ENEIO**
Reset	See text	0	0	0	0	0	0	0

Bit	Description
0	**ENEIO (Enable Expansion I/O).** Enables the use of the I/O Expansion protocol for expanding the amount of GPIO pins available to the application. In IRE and OBD environments, when ENEIO is cleared, the associated pins are used as GPIO signals. When set, enables the use of BIU I/O Zone (SELIO) for the interface with off-chip logic. Use the BIU I/O Zone configuration to select the access parameters to the I/O Expansion logic and its bus width.
1	**ENEMEM (Enable Expansion Memory).** Enables the use of the Expansion memory for expanding the amount of memory available to the application to more than what is provided on chip. When cleared, the associated pins are used as GPIO signals. When set, enables the use of BIU Zone 0 (SEL0) and the associated address and data lines for the interface with flash or SRAM devices. Use BIU Zone 0 configuration to select the access parameters to the Expansion memory and its width. Set this bit in the 176-pin package only.

Bit	Description
2	**EXMEM16 (16-bit-Wide Expansion Memory).** Enables the use of the 16-bit-wide Expansion memory, when the ENEMEM is set. The bus width indicated in this register should be the same as the bus width defined in Zone 0 of the BIU (SEL0). 0: The External Memory to be eight bits wide 1: Enables the use of a 16-bit-wide External Memory
4-3	**CLKOM (Clock Out Mode).** Selects the mode of operation of the CLKOUT signal. This field is in effect only when this signal is set for output. **Bits** **4 3 Description** 0 0: Reserved (default) 0 1: CLK - a clock at the core frequency is driven out 1 0: 32.768 KHz clock output 1 1: Reserved
6-4	**Reserved.**
7	**GTMON (Go-to Target Monitor Set Flag).** This bit is set to indicate that the code should jump to the beginning of the Target Monitor (TMON). The reset routine of the application should check this bit and behave accordingly. Once cleared, this bit can not be set by software.

External Interrupts Configuration Register (EICFG)

The EICFG register is a read/write, byte-wide register. It enables the use of some of the external interrupts. The EICFG bits have impact only when the pin is configured to its GPIO function. EICFG is cleared (00_{16}) on reset.

Note that some of the pins that have an external interrupt (not controlled by the EICFG register) have an alternate function that is an input. When a pin’s alternate function is selected, its interrupt function is also enabled.

The format of EICFG is as follows:

Location: 00 $FF00_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**EXWINT46**	**EXWINT45**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**EXWINT45 (Enable EXWINT45).** 0: EXWINT45 input is blocked (disabled) while not being read as a GPIO. 1: EXWINT45 input is open; this enables the detection of changes on the input to trigger interrupts.
1	**EXWINT46 (Enable EXWINT46).** 0: EXWINT46 input is blocked (disabled) while not being read as a GPIO. 1: EXWINT46 input is open; this enables the detection of changes on the input to trigger interrupts.
7-2	**Reserved.**

2.4.3 GPIO with Echo Configuration

Some of GPIO pins may be paired to generate an input to output echoing with software masking. The input GPIO are input ports (Px or Py) with a wake-up function. The output ports are a Px type enhanced with the echo mechanism, as described in Section 4.5.2 on page 114. The input echoing is enabled only when the respective Echo Enable bit in IOEE1-2 registers is set. The input port should be configured to enable the interrupt function.

Table 9. GPIO Echo Functions Routing and Echo Enable Bit Assignments

Output Port	Echo Enable Bit	Input Port
IOPA0	EEPA0 bit in IOEE1 register	IOPC4/EXWINT22
IOPA1	EEPA1 bit in IOEE1 register	IOPE6/$\overline{\text{LPCPD}}$/EXWINT45
IOPA2	EEPA2 bit in IOEE1 register	IOPE7/$\overline{\text{CLKRUN}}$/EXWINT46
IOPA3	EEPA3 bit in IOEE1 register	IOPF6/PSCLK4
IOPA4	EEPA4 bit in IOEE1 register	IOPF7/PSDAT4
IOPC0	EEPC0 bit in IOEE2 register	IOPE4/SWIN
IOPB0	EEPB0 bit in IOEE2 register	IOPD0/EXWINT20
IOPB1	EEPB1 bit in IOEE2 register	IOPD1/EXWINT21
IOPB2	EEPB2 bit in IOEE2 register	IOPC6/EXWINT23
IOPD3	EEPD3 bit in IOEE2 register	IOPD2/EXWINT24

Input to Output Echo Enable Register 1 and 2 (IOEE1 and IOEE2)

The IOEE1 and IOEE2 registers are read/write, byte-wide registers. They enable the echoing of some of the General-Purpose Input ports, equipped with wake-up, to a specified output. The IOEE bits have an impact only when the output GPIO is configured to its GPIO function. IOEE1 and IOEE2 are cleared (00_{16}) on reset.

The format of IOEE1 is:

IOEE1 Location: 00 $FF02_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**EEPA4**	**EEPA3**	**EEPA2**	**EEPA1**	**EEPA0**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**EEPA0.** 0: Echo Disabled 1: Echo Enabled
1	**EEPA1.** 0: Echo Disabled 1: Echo Enabled
2	**EEPA2.** 0: Echo Disabled 1: Echo Enabled
3	**EEPA3.** 0: Echo Disabled 1: Echo Enabled
4	**EEPA4.** 0: Echo Disabled 1: Echo Enabled
7-5	**Reserved.**

2.0 Signal/Pin Description and Configuration (Continued)

The format of IOEE2 is:

IOEE2 Location: 00 $FF04_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**EEPC0**	**EEPD3**	**EEPB2**	**EEPB1**	**EEPB0**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**EEPB0.** 0: Echo Disabled 1: Echo Enabled
1	**EEPB1.** 0: Echo Disabled 1: Echo Enabled
2	**EEPB2.** 0: Echo Disabled 1: Echo Enabled
3	**EEPD3.** 0: Echo Disabled 1: Echo Enabled
4	**EEPC0.**
7-5	**Reserved.**

See Table 9 on page 60 for the assignment of the Echo Enable bits to GPIO pairs.

3.0 Power, Reset and Clocks

3.1 POWER

3.1.1 Power Planes

The PC87591x has four power planes (wells), as shown in Table 10.

Table 10. PC87591x Power Planes

Power Plane	Description	Power Pins	Ground Pins
Host Domain	Powers the LPC interface and host controlled functions (except for RTC and MSWC) and some external signals[1]	V_{DD}	GND
Suspend	Powers the core domain, the host-core interface, the MSWC and their external signals[1]	V_{CC}	GND
Analog	Powers some internal analog circuits and their external signals[1]	AV_{CC}	AGND
Battery Backed	Powers the RTC, some MSWC registers, the 32.768 KHz clock/crystal oscillator signals and some other storage elements that should be preserved at all times[1]	V_{PP}[2]	GND

1. See the tables in Section 2.2.1 on page 42 to Section 2.2.13, specifically the Power Well column, for how the PC87591x external interface signals are assigned to various power planes.
2. V_{PP} is an internal power signal derived from V_{CC} or $V_{BAT.}$ V_{PP} is taken from V_{CC} if it is greater than the minimum value defined; otherwise, V_{BAT} is used as the V_{PP} source. For more details on the switching between them, refer to Section 6.2.9 on page 360.

For correct PC87591x operation, V_{CC} must be applied whenever V_{DD} is applied. Apply AV_{CC} at the same time that V_{CC} is applied, protection is provided against rise-time differences only.

3.1.2 Power States

The PC87591x has three main power states:

- **Battery Fail**
  Host Domain, Suspend, Analog and Battery Backed power planes are all powered off (i.e., in this case V_{DD}, V_{CC}, AV_{CC} and V_{BAT} are all inactive).
- **Power Fail**
  Host Domain, Suspend and Analog power planes are powered off. Battery Backed power plane is powered on (i.e., in this case V_{DD}, V_{CC} and AV_{CC} are inactive; V_{BAT} is active).
- **Power Applied**
  Suspend, Analog and Battery Backed power planes are powered on, Host Domain power plane may be on or off (i.e., in this case V_{CC}, AV_{CC} are active, V_{BAT} and V_{DD} may be active or inactive).
  The Power Applied state has several sub-states, depending on the domain:
  - — The host domain has Host Power On and Host Power Off states, according to the V_{DD} status;
  - — The core domain and host-core interface have Active and Idle states. For details, see Section 4.17 on page 235.

The following power states are illegal:

- Host domain on and Suspend and/or Analog off (i.e., V_{DD} active and V_{CC} and/or AV_{CC} are inactive).
- Suspend on and Analog off (i.e., V_{CC} active and AV_{CC} inactive) and vice-versa.

3.0 Power, Reset and Clocks (Continued)

The following table summarizes the power states described in relation with the PC87591x power planes.

Table 11. PC87591x Power States and Related Power Planes

Power State		Host Domain (V_{DD})	Suspend (V_{CC})	Analog (AV_{CC})	Battery Backed (V_{PP})	V_{BAT}
Battery Fail		-	-	-	-	-
Power Fail		-	-	-	+	+
Power Applied	Active and Idle	x	+	+	+	x
	Host Power Off	-	+	+	+	x
	Host Power On	+	+	+	+	x
Illegal[1]		+	-	-	x	x
		x	+	-	x	x
		x	-	+	x	x

[1] Operation is not guaranteed.

Figure 10 shows the power state transitions.

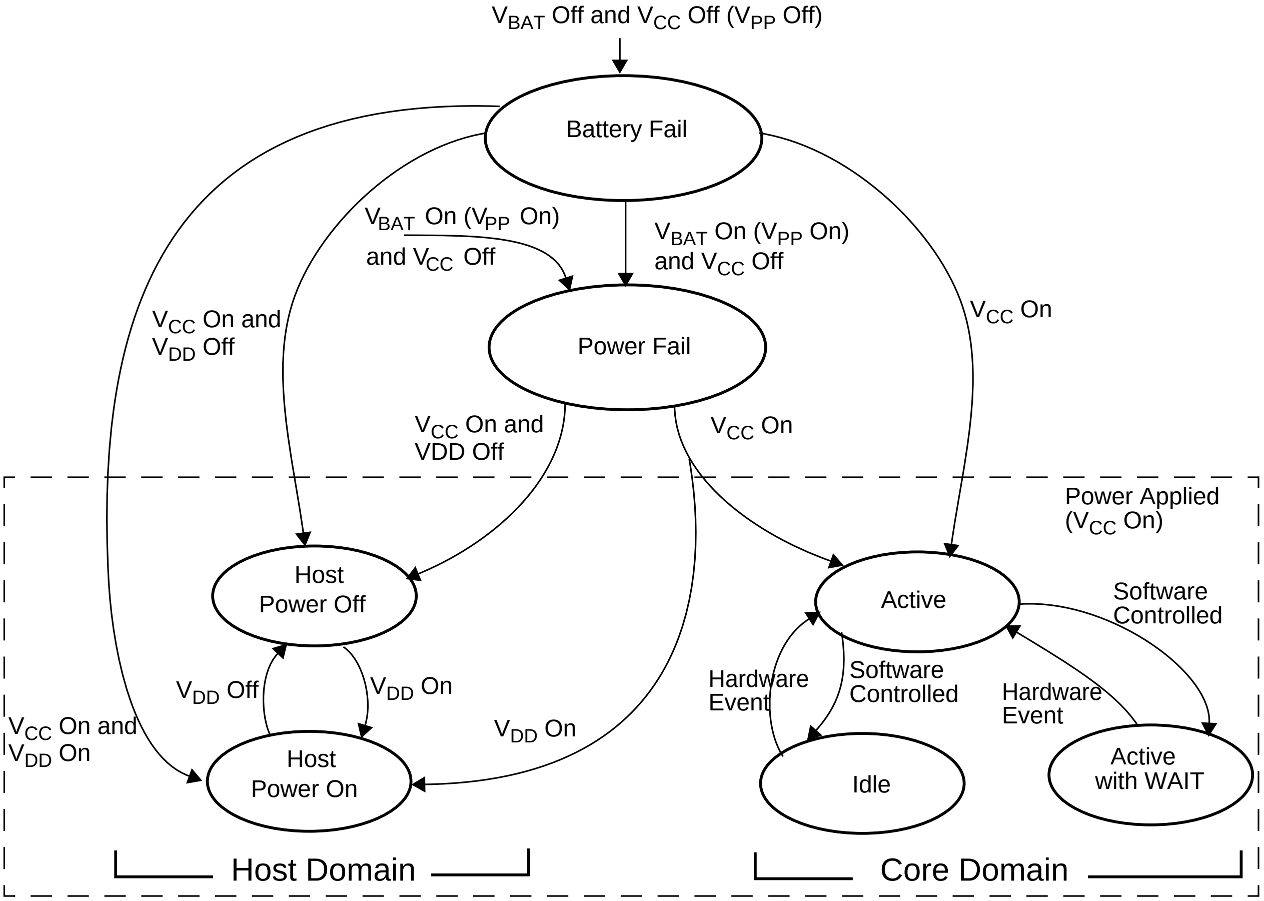


Figure 10. Power State Transitions

3.1.3 Power Connection and Layout Guidelines

The PC87591x requires a power supply voltage of 3.3V ± 10% for all the digital supplies (V_{CC} and V_{DD}). The analog power supply requires a voltage of 3.3V ± 10% or 3.3V ± 5%. A 3.3V ± 5% analog supply is recommended since it improves the accuracy of measuring input voltage values close to the full-scale (see Section 7.4.1 on page 378).

3.0 Power, Reset and Clocks (Continued)

Both AV_{CC} and V_{CC} should be applied for correct operation; and although the PC87591x is protected against damage if operated with only one of them (AV_{CC} or V_{CC}), the ADC and DAC modules do not function correctly in such a case and may cause current leakage. Therefore, ADC and DAC modules should be left at their reset condition (disabled) or disabled by firmware until AV_{CC} is within the limits specified above. In addition, the AV_{CC} is internally isolated from V_{CC} to enable AV_{CC} to be externally filtered or driven by a low-noise power supply.

The PC87591x is designed to operate with a Lithium backup battery that supplies voltage up to 3.6V. The PC87591x includes an internal current-limiting resistor on the V_{BAT} input as required to meet UL regulations.

V_{DD}, V_{CC} and V_{BAT} use a common ground return, named digital ground and marked GND. The analog circuits supplied by AV_{CC} use a separate ground return, named analog ground and marked AGND. This ensures effective isolation of the analog modules from noise caused by the digital modules.

The following directives are recommended for the PC87591x power and ground connections (see Figure 11 for the power supply connections):

Ground Connection

Use two ground planes, one for the digital signals (GND) and one for the analog signals (AGND). The following ground connections should also be made:

- The analog ground plane (AGND) should be connected at only one point to the digital ground plane (GND). This point should be located physically near the PC87591x.
- The analog ground return pin of the PC87591x (AGND) should be connected to the analog ground plane.
- The decoupling capacitors of the analog supply (AV_{CC}) pin should be connected to the analog ground plane as close as possible to the AGND pin.
- The ground reference of the voltage input signals to the ADC module (V_{IN}0-9 in Figure 11) should be connected to the AGND plane close to the PC87591x.
- The ground reference of the voltage output signals from the DAC module (V_{OUT}0-3 in Figure 11) should be connected to the AGND plane close to the PC87591x.
- All GND pins of the PC87591x must be connected to the GND plane.
- The decoupling capacitors of the Suspend digital supply (V_{CC}) pin should be located near each V_{CC}-GND pin pair; one side of each capacitor should be connected to the ground plane.
- The ground reference of the voltage input signals to the ACM module (KBV_{IN}0-7 in Figure 11) should be connected to the GND plane close to the PC87591x.
- The decoupling capacitors of the Host digital supply (V_{DD}) pin should be located near the V_{DD}-GND pair; one side of each capacitor should be connected to the ground plane.
- The ground return and the decoupling capacitor of the backup battery supply (V_{BAT}) pin should be located near the PC87591x; one of the capacitor's sides should be connected to the ground plane.

Note that low-impedance ground layers improve noise isolation and reduce ground bounce problems.

Power Connection

All PC87591x supply pins must be connected to the appropriate power plane, and decoupling capacitors must be used as recommended below.

The analog supply pin, AV_{CC}, should be connected to a low-noise power supply that has the same voltage as the digital supply (3.3V). If the AV_{CC} pin is connected to the same power supply as the V_{CC} pin, it is recommended to use an external LC or RC filter for the AV_{CC} pin. An example of an LC filter [L_1 and (C_1+C_2)] is shown in Figure 11. An RC filter is obtained by replacing L_1 (in Figure 11) with a 10Ω resistor (not shown in the figure).

Decoupling Capacitors

The following decoupling capacitors should be used to reduce power supply deeps, ground bounce and EMI (refer to Figure 11 for the position of capacitors, e.g., C1, C2, etc.):

- Suspend digital supply (V_{CC}): Place one 0.1 µF capacitor (C3) on *each* V_{CC}-GND pin pair as close as possible to the pin pair. Also, place one 10–47 µF C_4 tantalum capacitor on the common net as close as possible to the chip.
- Host digital supply (V_{DD}): Place one 0.1 µF capacitor (C5) on the V_{DD}-GND pin pair as close as possible to the pin pair. Also, place one 10-47 µF tantalum capacitor (C6) on the common net as close as possible to the chip.
- Backup battery (V_{BAT}): Place one 0.1 µF capacitor (C7) on the V_{BAT} pin as close as possible to the pin.
- Analog supply (AV_{CC}): Place a 0.1 µF capacitor (C2) and a 10-47 µF tantalum capacitor (C1) on the AV_{CC} pin as close as possible to the pin.

3.0 Power, Reset and Clocks (Continued)

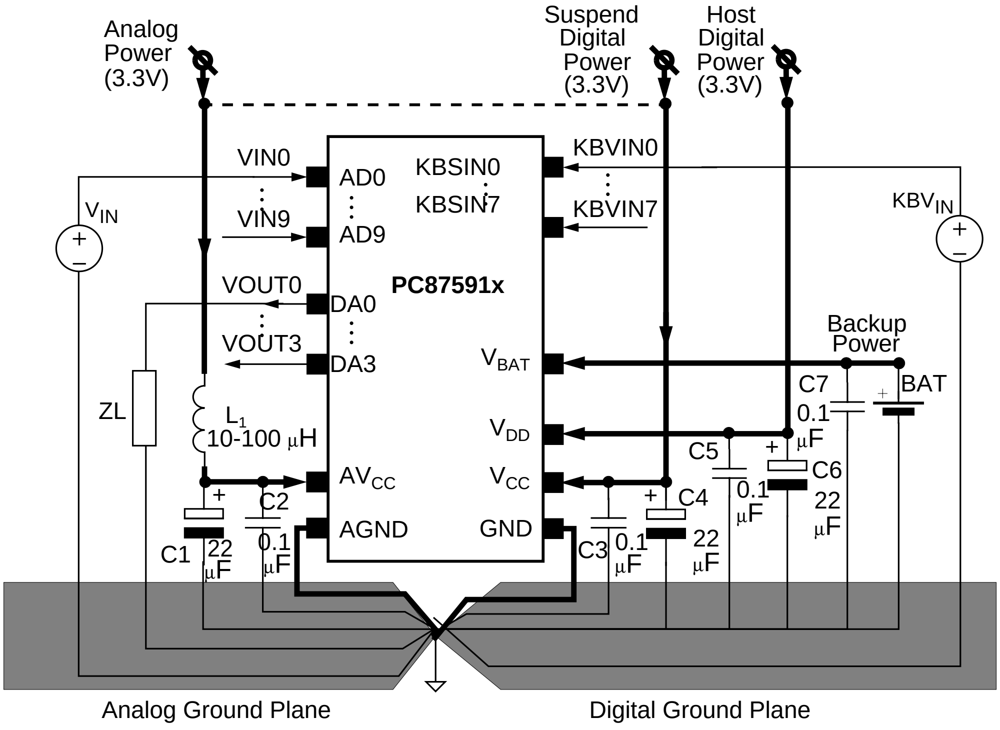


Figure 11. PC87591x Power Supply Connection

3.2 RESET SOURCES AND TYPES

The PC87591x has several input reset types:

- V_{PP} Power-Up reset (for V_{PP} supplied functions only)
  Activated when either V_{CC} or V_{BAT} is powered up after both have been off.
- V_{CC} Power-Up reset
  Activated when V_{CC} is powered up.
- Warm reset
  Activated on $\overline{\text{RESET1}}$ input falling edge.
- WATCHDOG and Debugger Interface resets:
  — WATCHDOG reset
  Activated on request from the TWD module (WATCHDOG signal is asserted); see Section 4.10 on page 164.
  — Debugger Interface reset
  Activated on request from the Debugger Interface module used during debug and flash updates; see Section 4.19 on page 248.
- Host Domain reset
  This is divided into two sub-groups: Host Domain Hardware and Host Domain Software reset. Combinations consisting of the flowing events are used to trigger these reset operations:
  — $\overline{\text{RESET1}}$ active
  — When V_{DD} is active, $\overline{\text{RESET2}}$ is active
  — On V_{DD} power-up
  — A Software reset triggered by a write of 1 to bit 1 of SIOCF1 register in the SuperI/O Configuration registers; see Section 6.1.8 on page 343

Unless otherwise noted, reset references throughout the PC87591x modules default to the following types:

- For V_{PP} retained functions: V_{PP} Power-Up reset
- For core domain functions and host-core interface functions: V_{CC} Power-Up reset, Warm reset, WATCHDOG reset, Debugger Interface reset and Software reset
- For host domain functions: Host Domain reset

3.0 Power, Reset and Clocks (Continued)

In DEV environment, the PC87591x outputs to the $\overline{\text{BRKL\_RSTO}}$ signal an indication that a reset occurred at the core domain. See Section 4.20.3 on page 272 for the implementation and usage of $\overline{\text{RSTO}}$.

The following sections detail the sources and effects of the various resets on the PC87591x, per reset type.

3.2.1 V_{PP} Power-Up Reset

V_{PP} is an internal power signal derived from V_{CC} and V_{BAT}. V_{PP} Power-Up reset is generated by an internal circuit that detects the status of the V_{PP} power. V_{PP} Power-Up reset signal is active from the rising of V_{PP} until the V_{PP} power is detected as “good” (i.e., V_{PP} is above V_{BATDTC}). When active, this signal resets all registers whose values are retained by V_{PP}.

For more details, see Section 6.2.9 on page 360.

3.2.2 V_{CC} Power-Up Reset

V_{CC} Power-Up reset is generated by an internal circuit. The PC87591x performs a V_{CC} Power-Up reset when V_{CC} power is applied. This reset is completed t_{IRST} after the internal clocks have stabilized (see Section 7.6.2 on page 384).

If the 32 KHz crystal is disabled before V_{CC} power-up, external devices should wait at least t_{32KOSC} before accessing the PC87591x. Any host processor access during this time results in:

- The host processor is stalled (by driving a “Long WAIT sync” response on the LPC bus) until after the reset process is completed and the bus request can be performed.
- If HRAPU bit in MSWCTL1 register is set (1), the host processor is reset by asserting $\overline{\text{KBRST}}$ until the internal reset is completed.

On V_{CC} Power-Up reset, the PC87591x responds as follows:

- Enables the 32 KHz crystal, if it is disabled.
- Resets the High-Frequency Clock Generator (HFCG) to its default frequency.
- Loads default values to all registers whose values are retained by V_{CC}.
- Puts pins with strap options into TRI-STATE and enables the internal pull-downs on the strap pins.
- Samples the values of the strap pins.
- Resets the TAP controller of the Debugger Interface module.
- Resets the MSWC, excluding those MSWC registers whose values are retained by V_{PP}.
- Resets Port PC0.
- Carries out all the Warm reset actions (see below).

3.2.3 WATCHDOG Reset and Debugger Interface Reset

The PC87591x generates a WATCHDOG reset on request from the TWD module (i.e., a WATCHDOG signal is asserted). It generates a Debugger Interface reset on request from the Debugger Interface module (reset command). During these resets, the PC87591x performs the V_{CC} Power-Up reset actions, with the following exceptions:

- The PC87591x does not sample the value of any strap pin; instead, it maintains the configuration determined by the strap pins at V_{CC} Power-Up reset.
- It does not reset the TAP controller.
- On Debugger I/F, reset PC0 is not reset (it is reset on WATCHDOG reset).
- It resets the HFCG to its default frequency.
- Some MSWC registers do not reset on WATCHDOG or Debugger I/F reset.

The reset periods are identical to the V_{CC} Power-Up reset period.

3.2.4 Warm Reset

Warm reset is activated on the falling edge of $\overline{\text{RESET1}}$ input. If Warm reset and V_{CC} Power-Up reset occur at the same time, V_{CC} power-up takes precedence.

During a Warm reset, the PC87591x responds as follows:

- Terminates core executed instructions
- Discards results not yet written to memory
- Eliminates any pending core interrupts and traps
- Clears the internal latch for the core domain’s edge-sensitive external interrupt
- Deactivates the external bus control signals $\overline{\text{WR(0-1)}}$, $\overline{\text{SEL(0-1)}}$, $\overline{\text{SELIO}}$, $\overline{\text{RD}}$ and BST(0-2)
- Puts the address A(0-19) and data D(0-15) buses in TRI-STATE
- Switches to core domain’s Active mode
- Loads default values into registers, with the exception of:
  - — The Mobile System Wake-Up Control (MSWC) and RTC registers retained by $V_{PP.}$
  - — Some registers that are specifically noted to be cleared only by other events
- Resets the Debugger interface, except for TAP controller

During Warm reset, strap pins are not sampled and the configuration determined at V_{CC} power-up is unaffected by subsequent Warm resets.

The PC87591x core domain and some parts of the host-core interface functions can operate when $\overline{\text{RESET1}}$ is still asserted (low). Some parts of the Host Domain Functions that are reset by the Host Domain reset (see following section) are kept reset as long as $\overline{\text{RESET1}}$ is asserted.

3.2.5 Host Domain Reset

Using $\overline{\text{RESET2}}$ Input

The $\overline{\text{RESET2}}$ input signal is enabled as an alternate function on one of two pins. The Protection Word in the flash memory is used to define whether the $\overline{\text{RESET2}}$ input is enabled and, if so, on which of the two pins. See Section 4.16.6 on page 226 for details on $\overline{\text{RESET2}}$ input configuration.

Note that enabling $\overline{\text{RESET2}}$ input on either of the signals causes that pin to be used as an input GPIO with an interrupt input associated with it. It is recommended to use this interrupt function to interrupt the core when a $\overline{\text{RESET2}}$ event occurs and to use the interrupt routing to stop activities and resume default values to some system elements not directly reset by the event.

Host Domain Reset Actions

The reset actions on the host domain are broken into two categories that depend on the reset event source; some of the actions may happen for both these reset source types and thus are named Host Domain Rest.

Host Domain Hardware Reset: While $\overline{\text{RESET1}}$ is active, $\overline{\text{RESET2}}$ is also active if V_{DD} is on or during V_{DD} power-up.

Host Domain Hardware reset performs the following actions:

- It brings the LPC interface state machine to its inactive state.
- It resets all SuperI/O configuration registers except for those that are battery-backed (see Section 6.1.8 on page 343).
- It resets Shared Memory Host Controlled registers

Host Domain Software Reset: This reset is triggered by a write of 1 to bit 1 in SIOCF1 register in the SuperI/O Configuration registers.

Host Domain Software reset performs the following actions:

- It resets all SuperI/O configuration registers except those noted to be protected from software reset (i.e., bits that are locked from write accesses).
- It resets Mobile System Wake-Up Control (MSWC) bits and RTC bits marked to be reset by software.

Note that lock bits and memory protect bits are not reset by software reset; instead, they require a hardware reset to unlock them; this requirement protects these bits from any faulty or malicious software.

For more details, see Section 6.1.3 on page 341.

Host Domain Reset: This term is used when a bit is reset by either a Host Domain Hardware reset or Host Domain Software reset.

3.3 CLOCK DOMAINS

The PC87591x has three clock domains, as shown in Table 12.

Table 12. PC87591x Clock Domains

Clock Domain	Frequency	Source	Usage
Core	See Section 4.18 on page 239	HFCG	Core domain and host-core interface
LPC	Up to 33 MHz	LPC clock input	LPC bus interface
RTC	32 KHz	Clock input or on-chip oscillator[1]	RTC, HFCG, TWD, PMC, ACM

[1] See Section 6.2.3 on page 356 and Section 6.2.4 on page 357.

Core Domain Clock

The core clock is sourced by the HFCG. The following section gives an overview of the clock domain.

On V_{CC} Power-Up reset, the HFCG is set to generate a 4 MHz clock. The HFCG frequency can be modified by core firmware. See Section 4.18 on page 239.

On WATCHDOG reset (or Debugger interface reset), the HFCG is reset.

LPC Clock

The LPC interface is driven by the LCLK input. LCLK frequency can be up to 33 MHz. The clock $\overline{\text{CLKRUN}}$ signal may be used as part of a power management scheme that slows the clock down; see “CLKRUN Functionality” on page 343.

RTC (32 KHz) Clock

The RTC clock is the reference for the generation of all other clocks in the PC87591x. The clock is enabled at V_{CC} power-up and is kept active while V_{PP} is active. Keeping the RTC clock active significantly reduces the PC87591x wake-up time. The RTC clock is used for time keeping in the RTC; it is also fed into the Power Management Controller (PMC), which provides the clock to other functions (such as TWD and ACM) that are active at all times (including Idle).

4.0 Embedded Controller Modules

4.1 BUS INTERFACE UNIT (BIU)

The BIU directly interfaces with a wide variety of devices, including ROM, SRAM and flash memory devices and I/O devices. It interfaces via address, data and control buses without the need for external glue logic.

The BIU also defines the access time to the on-chip flash Main block to provide cycle-by-cycle compatibility between environments; see Section 1.4 on page 29 and “Accessing Base Memory” on page 35. To optimize access time to the on-chip Base Memory, the Base Memory is split into Fast and Slow Zones. These zones are associated with Zone 1 and Zone 2 of the BIU, respectively. The SEL1 output signal of the PC87591x is the logical AND of the two SELi outputs of Zones 1 and 2.

4.1.1 Features

- Four address zones for static devices (SRAM, ROM, flash, I/O).
- Basic bus cycle: two clock cycles.
- Configurable fast read bus cycles with 1-cycle read duration.
- Wait states: configurable between zero and seven clock cycles.
- Hold cycles: configurable between zero and three clock cycles.
- I/O expansion support.
- Configurable burst on read.
- Burst read: one clock cycle.
- Configurable early write or late write.
- Bus width: configurable per zone - 16-bit or 8-bit.

4.1.2 Functional Description

Interface

The BIU interfaces between:

- Internal core bus
- External static memory
- Off-chip I/O (memory-mapped) devices

The BIU performs the following functions:

- Distinguishes between four static memory zones
- Selects the relevant configured parameters of the accessed zone (e.g., the number of wait states)
- Issues the appropriate bus cycle to access the zone

Each memory zone has a different address range and a set of parameters that define access to this zone. The set of parameters is software configurable.

Static Memory and I/O Support

The BIU accesses static memory devices (ROM, SRAM, flash and I/O devices) using static read and write bus cycles. The BIU can be configured to extend the bus cycles with wait cycles.

The BIU supports burst read bus cycles if the accessed zone is configured as burstable. (A burst-read bus cycle is an extension of the basic-read bus cycle in which additional data is accessed. A burst access usually requires only one clock cycle per additional data item. It may be extended by up to two clock cycles per additional data item.)

To support both I/O and static memory devices that require long hold times at the end of the access, the BIU can be configured to add up to three T_{hold} clock cycles at the end of the bus cycle. In addition, the BIU can be configured to insert a T_{idle} clock cycle between two consecutive accesses to different zones.

Byte Access

The internal core bus is 16-bits wide and supports byte and word transactions.

The BIU issues the appropriate bus cycle to access the right bytes, according to the core bus transaction and the memory device bus width. Table 13 and Table 14 summarize the details:

4.0 Embedded Controller Modules (Continued)

Table 13. Bus Cycles of a 16-Bit Data Bus

Number of Bytes	Transferred Core Bus Bytes		Address (LSB)	Data Bus Pins
1		B0	0	0-7
1	B1		1	8-15
2	B1	B0	0	0-15
2	B1		1	8-15
		B0	Then 0	0-7

Table 14. Bus Cycles of an 8-Bit Data Bus

Number of Bytes	Transferred Core Bus Bytes		Address (LSB)	Data Bus Pins
1		B0	0	0-7
1	B1		1	0-7
2		B0	0	0-7
	B1		Then 1	0-7[1]
2	B1		1	0-7
		B0	Then 0	0-7

[1] Burst bus cycle, if burstable; otherwise, the core transaction is broken into “basic” bus cycles.

On write cycles of a single byte, the remaining eight bits of the bus are floating.

On read cycles of a single byte, the remaining eight bits of the bus are ignored. There is no need for external pull-up resistors.

Clock and Bus Cycles

There are two types of bus cycles: data transfer and non-data transfer. Data transfer bus cycles cause transfer of data from or to the memory device. Non-data transfer bus cycles (described in Section 4.1.9 on page 81) are used for observability of internal bus transactions and do not involve data transfer from or to external devices.

There are four types of data transfer bus cycles:

- Early write
- Late write
- Normal read
- Fast read

The BIU uses EWR configuration bit in BCFG register to select the early or late write data transfer bus cycle. It uses FRE in SZCFGn register (where “n” refers to zone 0, 1 or 2) to select normal read or fast read data transfer bus cycles.

The basic late write bus cycle takes two clock cycles. The basic early write bus cycle takes three clock cycles. When the BIU uses the early write bus cycle, the $\overline{\text{RD}}$ signal is not required for interfacing with the memory device (with the exception of flash). On reset, the early write bus cycle is configured.

The basic normal read bus cycle takes two clock cycles. Fast read bus cycle always takes one clock cycle. On reset, the normal read bus cycle is configured.

Notes:

1. In the descriptions that follow, the “n” in $\overline{\text{SELn}}$ signal refers to two of the three available BIU select signals (numbered 0 or 1, corresponding to zone 0 or zone 1 respectively). The third signal is labelled $\overline{\text{SELIO}}$.
2. For all timing diagrams, the value of BST0-2 depends on the type of core bus transaction.
3. In the following paragraphs, SZCFGn refers to three of the four BIU zone configuration registers (n = 0, 1 or 2) - the fourth configuration register is labelled IOCFG.

4.0 Embedded Controller Modules (Continued)

4.1.3 Clock Cycles

Basic Bus Cycle

A basic bus cycle comprises one to three clock cycles (depending on the type of bus cycle). Adding extra wait or hold clock cycles extends the data transfer bus cycles. Every data transfer bus cycle has the T1 and T2 clock cycles, with the exception of the fast read bus cycle, which has only one clock cycle (T1-2).

T_{idle} Cycle Clock cycles that are not used for bus cycles are called Idle clock cycles (T_{idle}). T_{idle} cycles are added when the BIU does not need to generate a bus transaction or as specifically configured pauses between two consecutive transactions. When more than one T_{idle} cycle is requested as a pause, the T_{idle} cycles overlap and only one T_{idle} cycle is added.

T_{idle} clock cycles can be inserted between two consecutive accesses in different zones (to allow long hold times or buffer disable times). To do this, either program IPRE and/or IPST in SZCFGn register (or IPST in IOCFG register); see Figure 18 on page 77.

T_{idle} clock cycles are also added between an early write and a read bus cycle, and between a late write and a fast read bus cycle; see Figure 23 on page 80.

T1 Cycle Every bus cycle starts with T1. In this clock cycle, the address of the selected device (either external or internal) is set on the address pins. Write bus cycles never drive data during T1.

T2 Cycle The read T2 bus cycles always sample the data at the end of T2.

The write T2 bus cycles always drive data during T2. If no T_{hold} clock cycles follow, the data bus is put in TRI-STATE after the T2 cycle.

T1-2 Cycle The fast read T1-2 bus cycle is a one-cycle read transaction.

At the start of the clock cycle, the address of the selected device is set on the address pins, and the $\overline{\text{SELn}}$ and $\overline{\text{RD}}$ signals are activated. At the end of the clock cycle, the BIU samples the data.

T3 Cycle Early write bus cycles always have the T3 clock cycle. No other bus cycles have this clock cycle.

At the start of this clock cycle, $\overline{\text{SELn}}$ (or $\overline{\text{SELIO}}$) is deactivated; then $\overline{\text{WR0-1}}$ is deactivated. The address and data remains valid until T3 is completed. If no T_{hold} clock cycles follow, the data bus is put in TRI-STATE after the T3 cycle.

Optional Clock Cycles

The following clock cycles are optional in a data transfer bus cycle:

- TIW (Internal Wait)
- T2B (T2 burst)
- TBW (Burst Wait)
- T_{hold}

TIW Cycle Extend the basic data transfer bus cycle by adding wait clock cycles. To do this, program WAIT in SZCFGn register (or IOCFG register) with the required additional wait clock cycles. Wait clock cycles generated by this action are named TIW (internal wait). TIW cycles are added after T1 and followed by T2 cycles. Data is always driven during wait clock cycles of a write bus cycle.

T2B Cycle Data of read burst bus cycles is sampled at the end of T2B. If the TBW cycle is not configured, the address is changed at the start of T2B. Write bus cycles do not have this clock cycle.

TBW Cycle A burst bus cycle can be extended by one wait clock cycle, named TBW. This is done according to WBR in SZCFGn register. The address is changed at the start of TBW. Write bus cycles do not have this clock cycle.

T_{hold} Cycle Hold cycles are added after T2 or T2B (if there is a burst bus cycle) or T3 (according to HOLD in SZCFGn or IOCFG register); the address and data (during a write bus cycle) are always valid during these cycles. The data bus is put in TRI-STATE after the last T_{hold}.

Other Clock Cycles

Special T_{idle} Cycle During T_{idle} cycles, one of the $\overline{\text{SEL0-1}}$ signals and the $\overline{\text{RD}}$ signal may be activated for one clock cycle. This happens due to special activity on the internal core bus. To avoid contention on the memory bus, it is guaranteed that this clock cycle is followed by a sufficient number of T_{idle} cycles before the next T1 cycle is performed.

The number of T_{idle} cycles that follows is at least the number required by the selected zone as configured in HOLD field in SZCFGn register.

4.0 Embedded Controller Modules (Continued)

Burst Read Cycles A read bus cycle consisting of the basic bus cycle plus additional clock cycles called “burst bus cycles”. The burst bus cycles occur if the bus is burstable (BRE in SZCFGn register is 1), the configured bus width is eight bits and the core attempts to read a word. When the bus is not burstable (BRE in SZCFGn register is 0), the BIU issues two separate read bus cycles. Write bus cycles are never burstable, and the BIU always issues two separate write bus cycles.

Control Signals

The write bus cycles use byte write qualifiers on $\overline{\text{WR0-1}}$ pins:

- They access an 8-bit wide memory on D0-7 data lines.
  One byte is accessed on basic bus cycles. Only the $\overline{\text{WR0}}$ pin is used as the byte write qualifier.
- They access a 16-bit wide memory on D0-15 data lines.
  Either one or two bytes are accessed on basic bus cycles. The $\overline{\text{WR0}}$ pin is used as an even byte (D0-7) write qualifier and $\overline{\text{WR1}}$ pin is used as an odd byte (D8-15) write qualifier.

4.1.4 Early Write Bus Cycle

If EWR in BCFG register is 1, the BIU uses early write bus cycles. This allows removal of the $\overline{\text{RD}}$ signal from the memory device interface. The basic early write bus cycle takes three clock cycles.

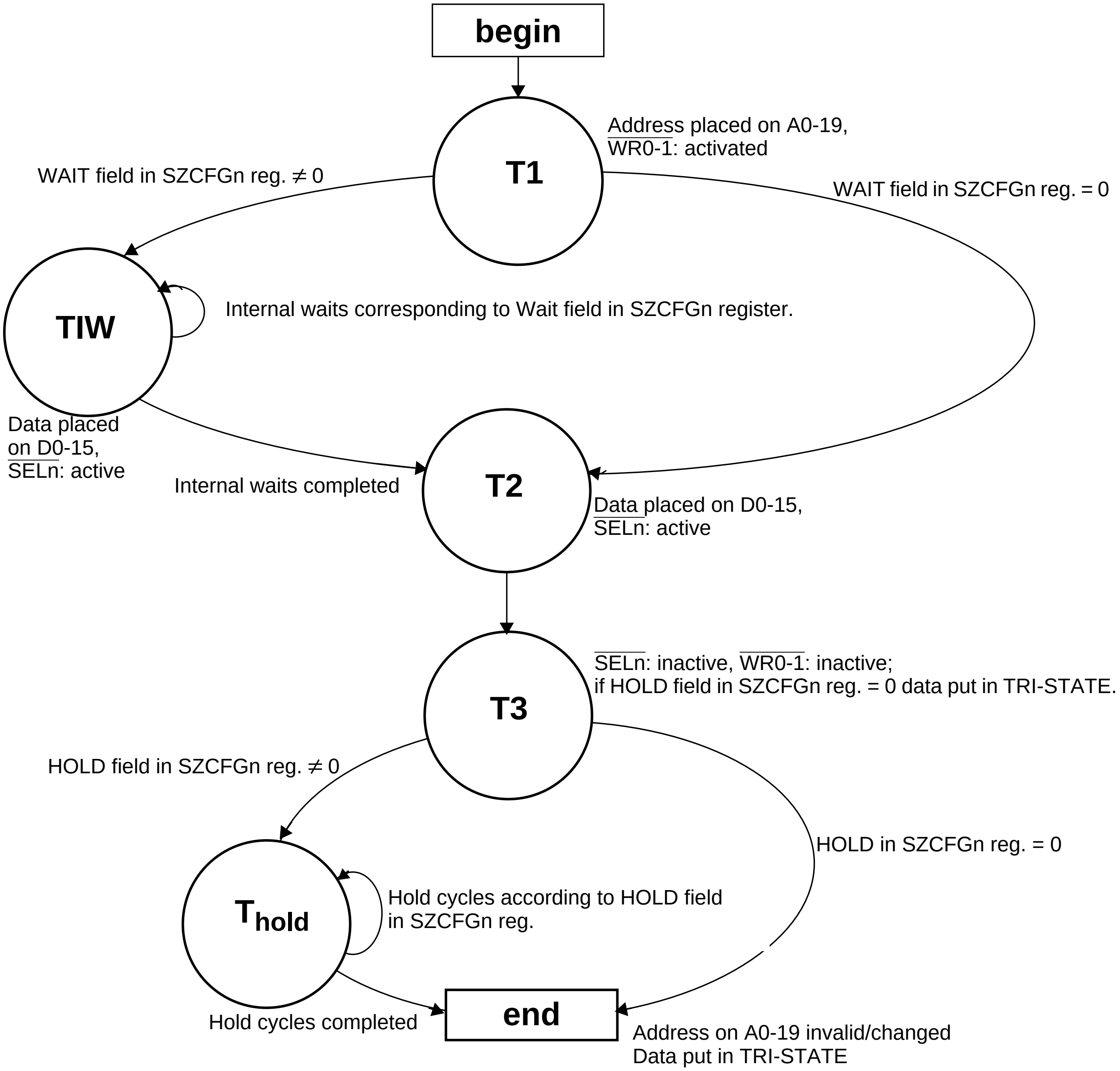
The cycle starts at T1; at this point, the data bus is in TRI-STATE, the address is placed on the address bus and $\overline{\text{RD}}$ is inactive. indicating that this is a write bus cycle. Then, $\overline{\text{WR0-1}}$ are activated.

At the first TIW or T2 (when there are no TIW cycles), the data is placed on the data bus and the $\overline{\text{SELn}}$ (or $\overline{\text{SELIO}}$) is activated. The bus transaction is terminated at T3; at this point, $\overline{\text{SELn}}$ (or $\overline{\text{SELIO}}$) becomes inactive. Then $\overline{\text{WR0-1}}$ become inactive and the data bus is put in TRI-STATE. The address remains valid until T3 is complete.

T_{hold} clock cycles may follow T3, according to HOLD in SZCFGn or IOCFG registers (may be 0). The address and data remain valid until the end of the last T_{hold} cycle. The data is put in TRI-STATE in the clock cycle after the last T_{hold} or T3 (if no T_{hold} cycle is configured); see Figures 12, 13 and 14.

If a read bus cycle immediately follows an Early Write bus cycle, an idle cycle is added between the two.

4.0 Embedded Controller Modules (Continued)


begin
T1
Address placed on A0-19,
WR0-1: activated
WAIT field in SZCFGn reg. ≠ 0
WAIT field in SZCFGn reg. = 0
TIW
Internal waits corresponding to Wait field in SZCFGn register.
Data placed
on D0-15,
SELn: active
Internal waits completed
T2
Data placed on D0-15,
SELn: active
T3
SELn: inactive, WR0-1: inactive;
if HOLD field in SZCFGn reg. = 0 data put in TRI-STATE.
HOLD field in SZCFGn reg. ≠ 0
HOLD in SZCFGn reg. = 0
Thold
Hold cycles according to HOLD field
in SZCFGn reg.
Hold cycles completed
end
Address on A0-19 invalid/changed
Data put in TRI-STATE

Note: References to SZCFGn also apply to the IOCFG register.
References to SELn also apply to the SELIO signal.

Figure 12. Early Write Bus Cycle

4.0 Embedded Controller Modules (Continued)

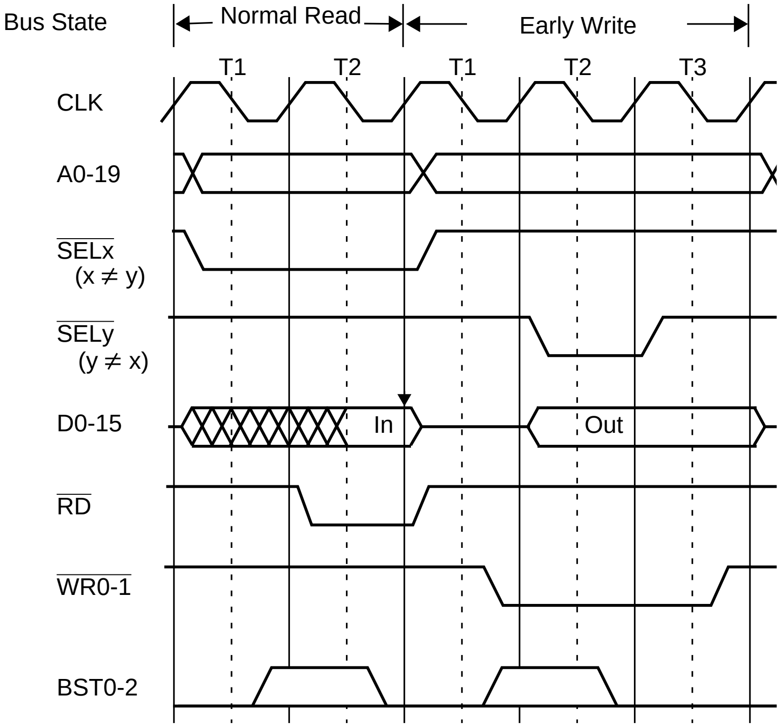

Bus State
Normal Read
Early Write
T1
T2
T1
T2
T3
CLK
A0-19
$\overline{\text{SELx}}$ (x ≠ y)
$\overline{\text{SELy}}$ (y ≠ x)
D0-15
In
Out
$\overline{\text{RD}}$
$\overline{\text{WR0-1}}$
BST0-2

Figure 13. Early Write Following Normal Read with 0 Wait

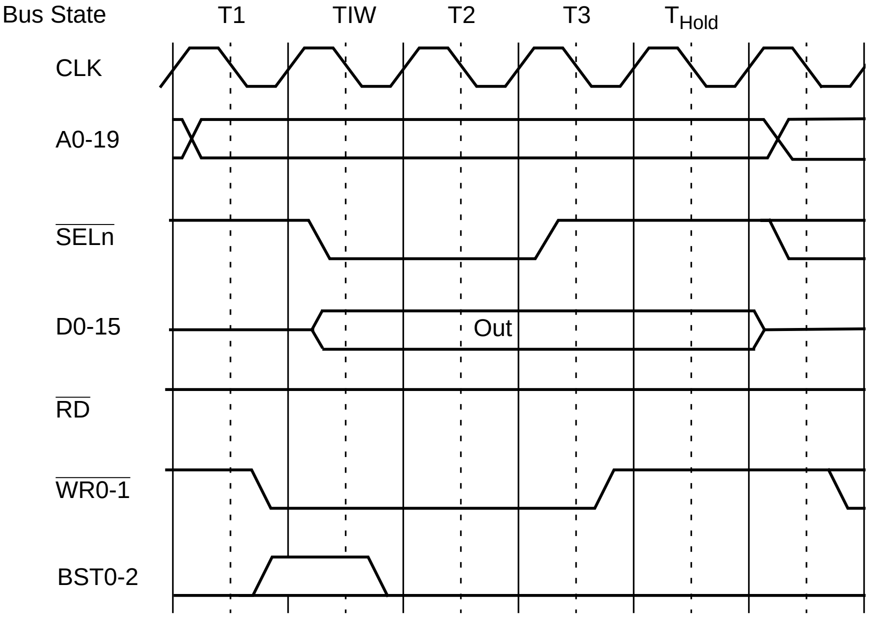


Figure 14. Early Write Bus Cycle with 1 Internal Wait and 1 Hold

4.1.5 Late Write Bus Cycle

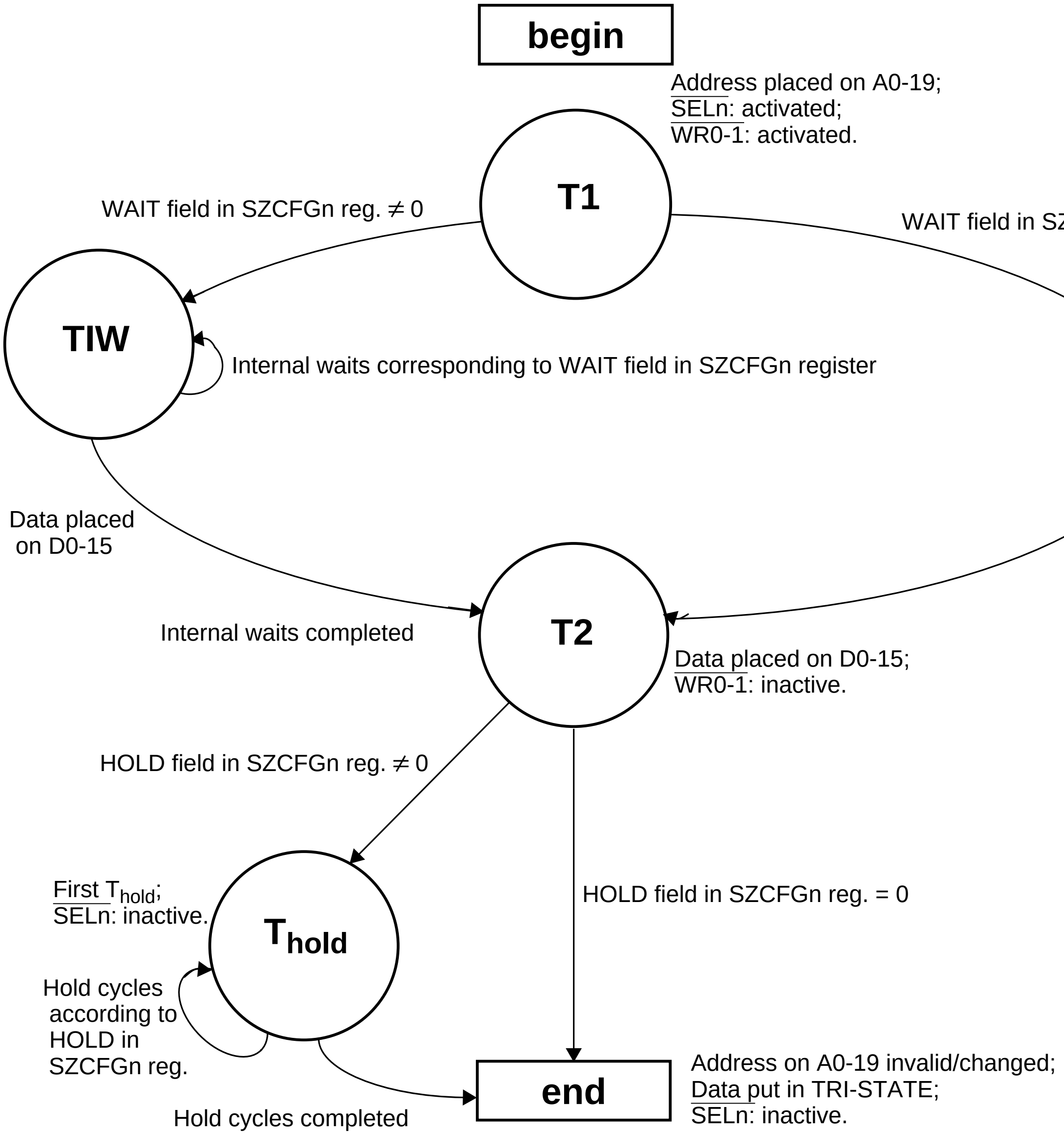
If EWR in BCFG register is 0, the BIU uses the late write bus cycle. The basic late write bus cycle takes two clock cycles. This write bus cycle requires the $\overline{\text{RD}}$ signal in the memory device interface.

A write bus cycle starts at T1. When the data bus is in TRI-STATE, the address is placed on the address bus and $\overline{\text{SELn}}$ (or $\overline{\text{SELIO}}$) is activated. Next, $\overline{\text{WR0-1}}$ are activated. $\overline{\text{RD}}$ is inactive to indicate this is a write transaction.

4.0 Embedded Controller Modules (Continued)

At the first TIW or T2 (when there are no TIW cycles), the data is placed on the data bus. The bus cycle is completed at T2; at this point, WR0-1 are deactivated. The address and data remain valid until T2 is completed.

After T2, the number of T_{hold} cycles specified by HOLD in SZCFGn register (may be 0) is added to the transaction. When T_{hold} cycles are added, the address and data remain valid until the end of the last T_{hold} cycle. SELn (or SELIO) is deactivated on the first T_{hold} cycle. When no T_{hold} cycles are specified, SELn (or SELIO) is deactivated in the clock cycle after T2 unless another read or write from the same zone follows. The data is put in TRI-STATE in the clock cycle after the last T_{hold} or T2 (if no T_{hold} cycle is configured); see Figures 15, 16 and 17.


begin
Address placed on A0-19;
SELn: activated;
WR0-1: activated.
T1
WAIT field in SZCFGn reg. ≠ 0
WAIT field in SZCFGn reg. = 0
TIW
Internal waits corresponding to WAIT field in SZCFGn register
Data placed on D0-15
Internal waits completed
T2
Data placed on D0-15;
WR0-1: inactive.
HOLD field in SZCFGn reg. ≠ 0
HOLD field in SZCFGn reg. = 0
First Thold;
SELn: inactive.
Thold
Hold cycles according to HOLD in SZCFGn reg.
Hold cycles completed
end
Address on A0-19 invalid/changed;
Data put in TRI-STATE;
SELn: inactive.

Note: References to SZCFGn also apply to the IOCFG register.
References to SELn also apply to the SELIO signal.

Figure 15. Late Write Bus Cycle

4.0 Embedded Controller Modules (Continued)

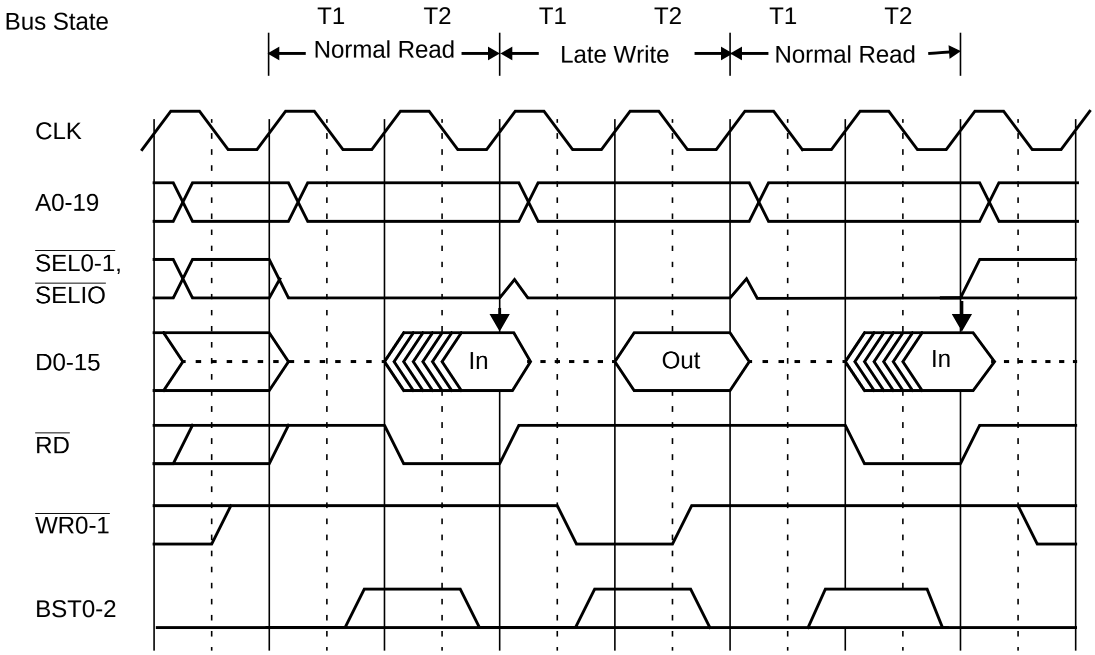


Figure 16. Late Write Bus Cycle Between Normal Read Bus Cycles with 0 Wait

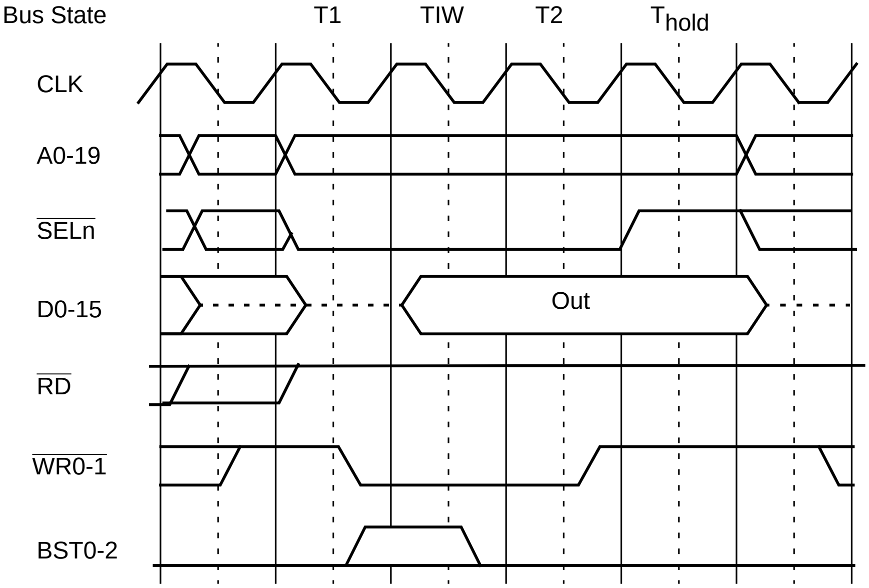


Figure 17. Late Write Bus Cycle with 1 Internal Wait and 1 Hold

4.1.6 Normal Read Bus Cycle

A read bus cycle starts at T1; at this point, the address is placed on the address bus, $\overline{\text{SELn}}$ (or $\overline{\text{SELIO}}$) is activated and $\overline{\text{WR0-1}}$ are inactive, indicating that this is a read bus cycle. The $\overline{\text{RD}}$ signal is activated on the first TIW or T2 (when there are no TIW cycles).

At the end of T2, the BIU samples the data on D0-7 or D0-15, according to BW signal in SZCFGn register. After T2, the number of T_{hold} cycles specified by HOLD in SZCFGn register (may be 0) is added. $\overline{\text{SELn}}$ and $\overline{\text{RD}}$ are deactivated on the first T_{hold} cycle. The address remains valid until the end of the last T_{hold} cycle.

4.0 Embedded Controller Modules (Continued)

When no T_{hold} cycles are specified, $\overline{SELn}$ is deactivated in the clock cycle that follows T2, unless another read from the same zone follows. The $\overline{RD}$ signal is always deactivated in the clock cycle following T2; see Figures 18, 19 and 20.

A burst bus cycle supplements the basic read bus cycle if the core attempts to access more bytes (i.e., a word) than the configured bus width (and BRE in SZCFGn register is set to 1). The burst bus cycle (T2B) follows T2 before the T_{hold} cycles (if configured). A wait clock cycle (TBW) is added between T2 and T2B if WBR in SZCFGn register is set to 1.

The address of the burst bus cycle is changed on TBW (if configured) or T2B (if no TBW). At the end of T2B, data is sampled. The $\overline{RD}$ signal is activated during the burst bus cycle and is deactivated in the clock cycle following T2B; see Figures 21 and 22.

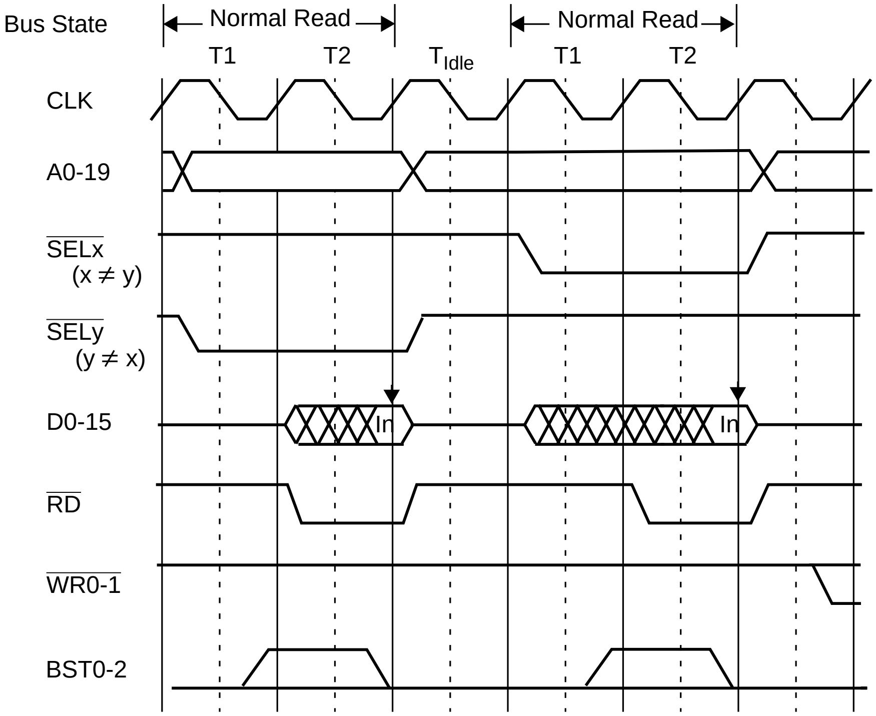


Figure 18. Two Basic Normal Read Bus Cycles with Idle In Between (IPST Bit in SZCFGy Register = 1, IPRE Bit in SZCFGx Register = 1)

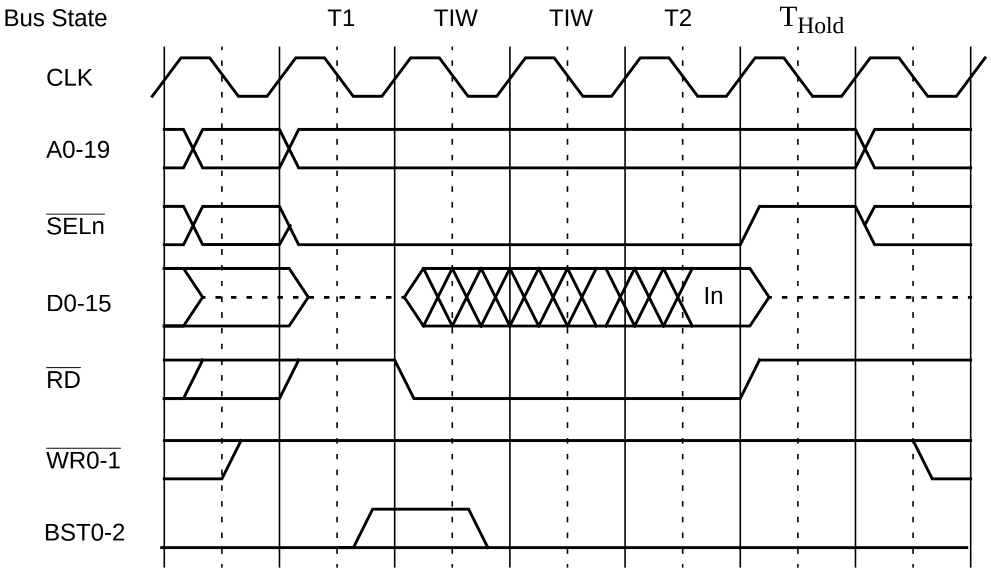


Figure 19. Normal Read Bus Cycle with 2 Internal Waits and 1 Hold

4.0 Embedded Controller Modules (Continued)

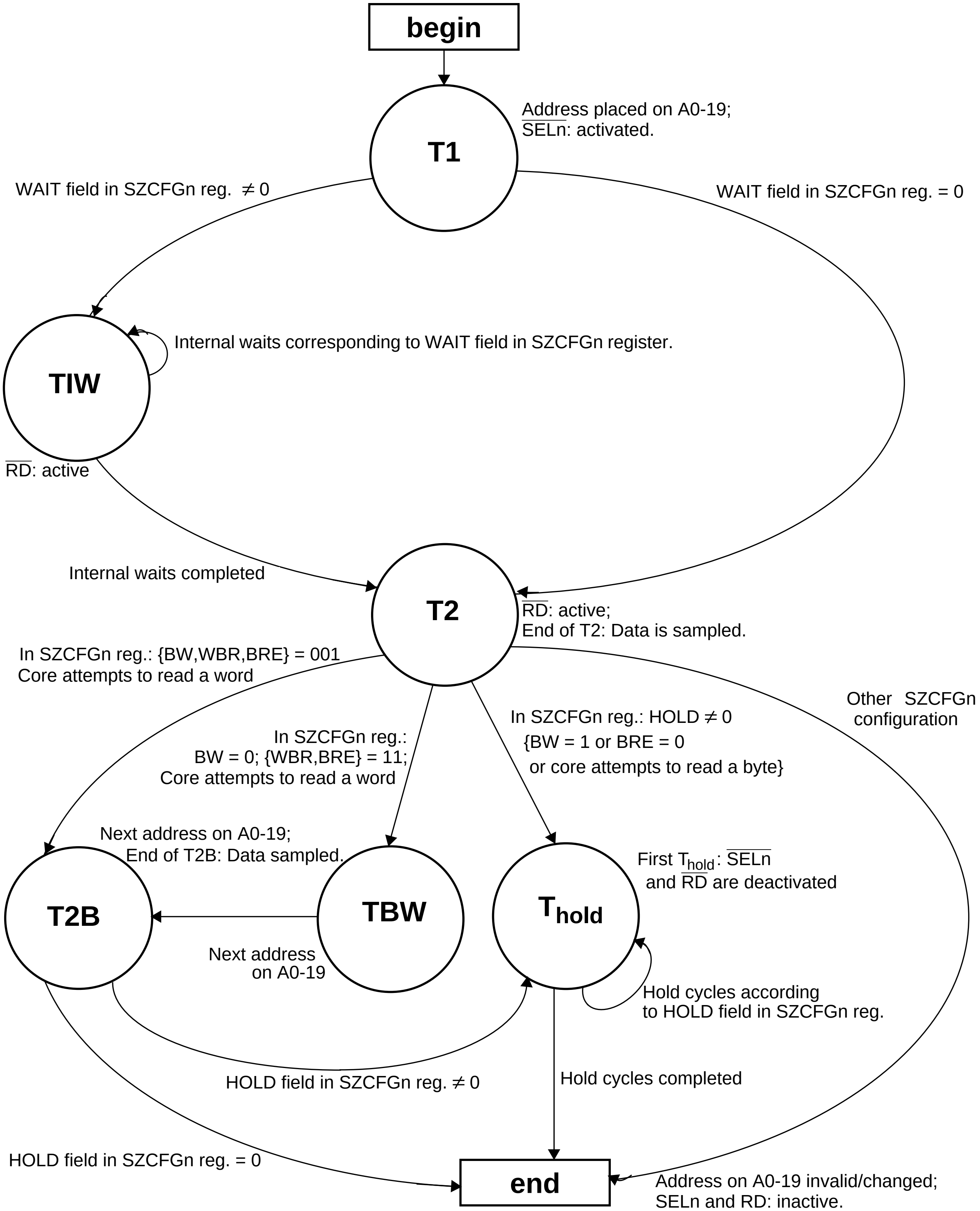

begin
T1
Address placed on A0-19;
$\overline{\text{SELn}}$: activated.
WAIT field in SZCFGn reg. ≠ 0
WAIT field in SZCFGn reg. = 0
TIW
Internal waits corresponding to WAIT field in SZCFGn register.
$\overline{\text{RD}}$: active
Internal waits completed
T2
$\overline{\text{RD}}$: active;
End of T2: Data is sampled.
In SZCFGn reg.: {BW,WBR,BRE} = 001
Core attempts to read a word
In SZCFGn reg.:
BW = 0; {WBR,BRE} = 11;
Core attempts to read a word
In SZCFGn reg.: HOLD ≠ 0
{BW = 1 or BRE = 0
or core attempts to read a byte}
Other SZCFGn configuration
Next address on A0-19;
End of T2B: Data sampled.
T2B
TBW
T_{hold}
First T_{hold}: $\overline{\text{SELn}}$
and $\overline{\text{RD}}$ are deactivated
Next address on A0-19
Hold cycles according to HOLD field in SZCFGn reg.
HOLD field in SZCFGn reg. ≠ 0
Hold cycles completed
HOLD field in SZCFGn reg. = 0
end
Address on A0-19 invalid/changed;
SELn and RD: inactive.

Figure 20. Normal Read Bus Cycle

Note: References to SZCFGn also apply to the IOCFG register.
References to $\overline{\text{SELn}}$ also apply to the $\overline{\text{SELIO}}$ signal.
TBW and T2B states do not exist in bus cycles of the IO zone.

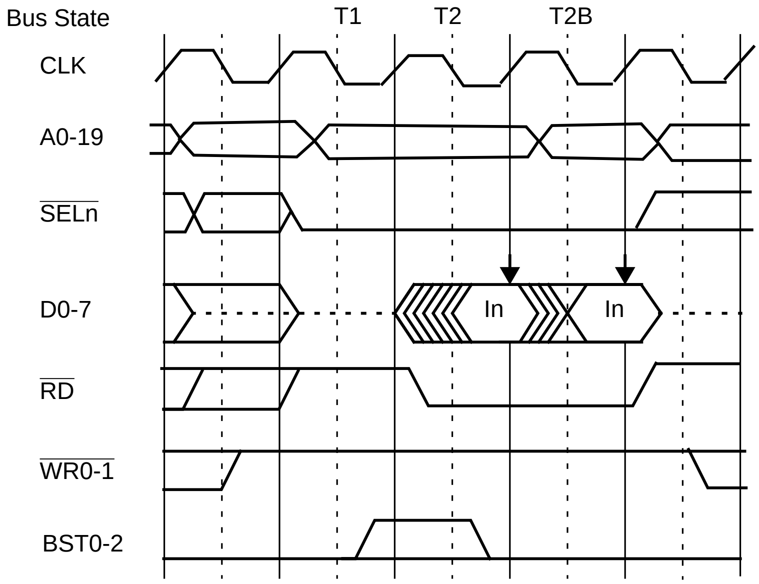


Figure 21. Normal Read Bus Cycle with 0 Wait on Burst

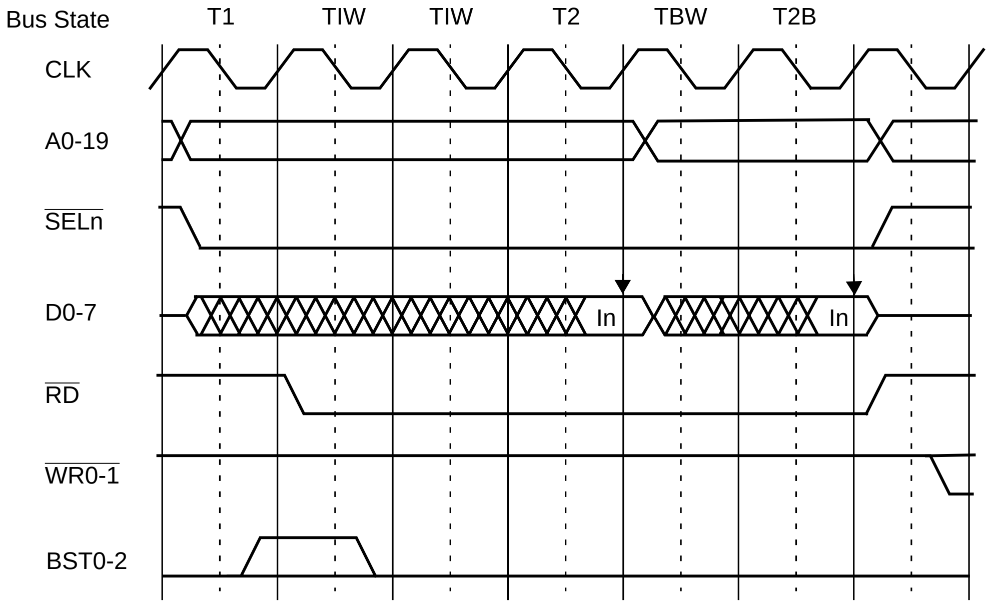


Figure 22. Normal Read Bus Cycle with 2 Internal Waits and 1 Wait on Burst

4.1.7 Fast Read Bus Cycle

When FRE bit in SZCFGn register is 1, the fast read bus cycle is enabled for zone n. The fast read bus cycle takes one clock cycle.

At the start of the T1-2 clock cycle, the address is placed on the address bus, and $\overline{\text{SELn}}$ and $\overline{\text{RD}}$ are activated. $\overline{\text{WR0-1}}$ are inactive, indicating a read bus cycle. At the end of the clock cycle, the BIU samples the data. $\overline{\text{SELn}}$ and $\overline{\text{RD}}$ are deactivated in the following clock cycle, unless another read from the same zone follows. If a write to the same zone follows and late write is configured, $\overline{\text{SELn}}$ remains activated. The address remains valid until the start of the clock cycle after the T1-2 clock cycle.

4.0 Embedded Controller Modules (Continued)

The fast read bus cycle cannot be extended by adding wait cycles (WAIT field in SZCFGn register is ignored during this bus cycle). Additionally, hold cycles cannot be added (HOLD field in SZCFGn register is also ignored). When a write bus cycle immediately precedes, in sequence, a fast read bus cycle, an idle clock cycle is forced between the two; see Figure 23.

Note that when the core attempts to access more bytes than the configured bus width (i.e., a word), the transaction is broken up into “basic” (T1-2) bus cycles.

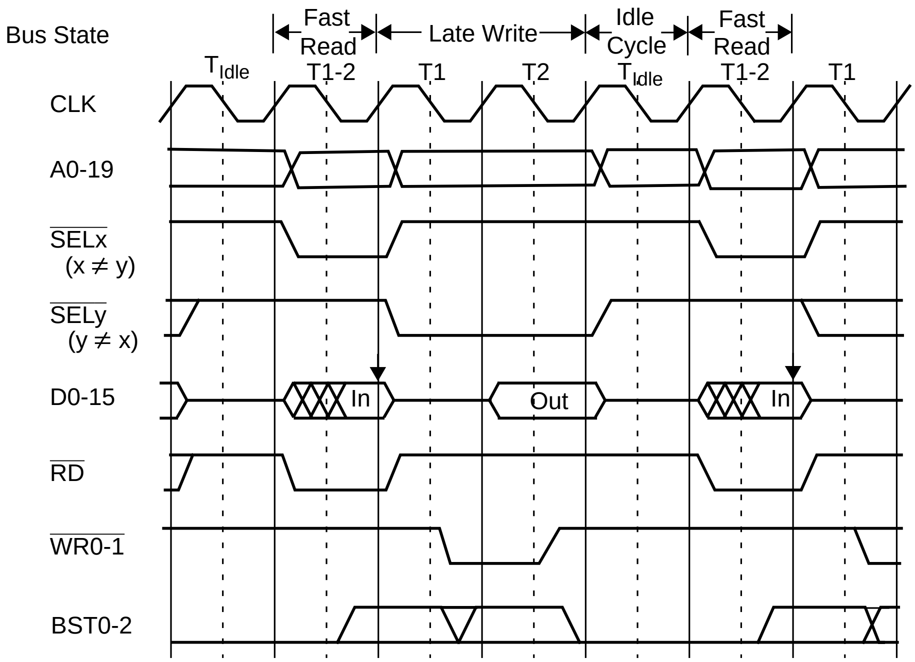


Figure 23. Fast Read Bus Cycle

4.1.8 I/O Expansion Bus Cycles

The I/O expansion bus cycles support the implementation of on-chip I/O port functionality (when the pins of the on-chip I/O ports are used to support DEV environments) and/or additional ports, using off-chip external logic.

I/O expansion bus cycles access the off-chip I/O device using the following signals:

- SELIO.
- Address lines A0-7.
- The RD and WR0-1 signals may be used.

The design minimizes the off-chip logic required to implement the I/O ports. It is costly to implement a port with pins individually configured for input or output. Implementing ports for input only or output only reduces expenses.

I/O expansion bus cycle is not generated during an access to a port register if one of the following conditions occurs:

- A port pin is available on-chip.
- All port pins are inputs and the port is being written.

4.0 Embedded Controller Modules (Continued)

I/O Expansion Read/Write Bus Cycle

These cycles are always preceded by a T_{idle} clock cycle (see Figure 24). The I/O zone is not burstable.

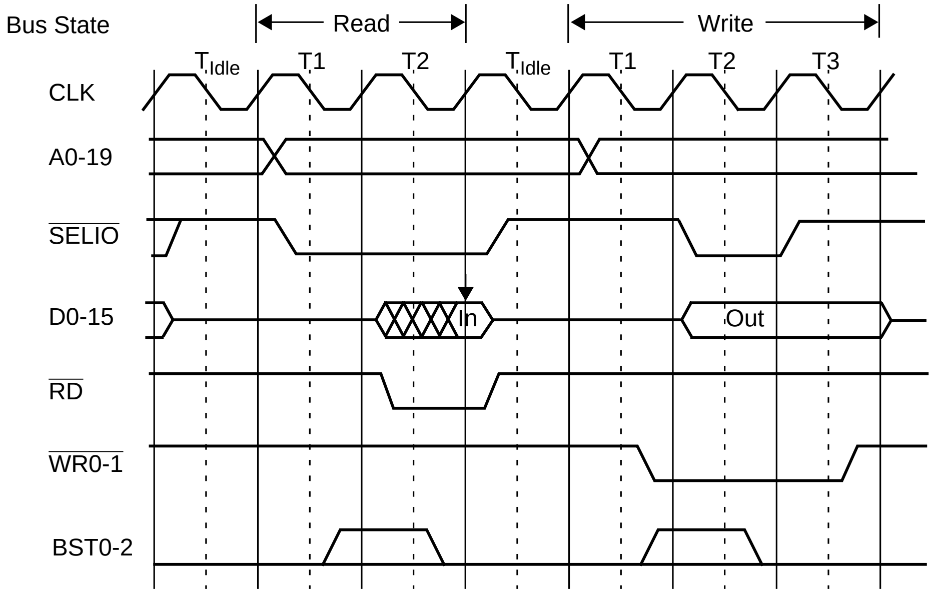
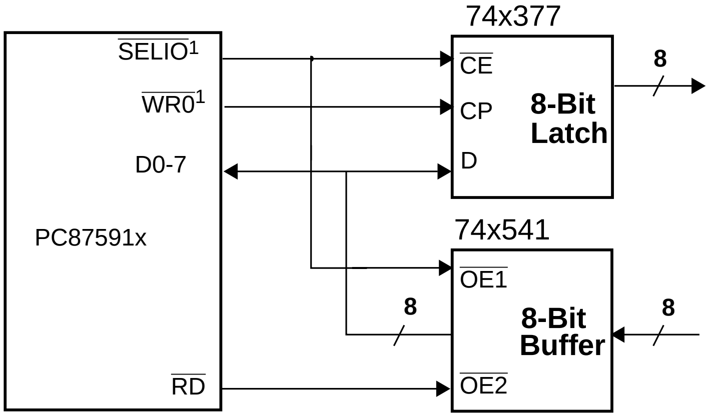


Figure 24. I/O Expansion Bus Cycles (EWR bit in BCFG Register = 1)

I/O Expansion Example

Figure 25 shows an example of how two ports can be implemented off-chip, using I/O expansion. This example implements two 8-bit ports by connecting the $\overline{\text{SELIO}}$, $\overline{\text{RD}}$ and $\overline{\text{WR0}}$ pins to the latch/buffer controls.



1. This routing is for late write. If early write, $\overline{\text{SELIO}}$ is routed to CP and $\overline{\text{WR0}}$ to $\overline{\text{CE}}$. All other routing is unchanged.

Figure 25. Example of an Implementation of Two Ports Using I/O Expansion

4.1.9 Development Support

The BIU provides the following support for development systems.

Bus Status Signals

The Bus Status BST0-2 signals indicate whether a transaction on the core bus was issued; they also indicate the transaction type; see Table 41 on page 273.

4.0 Embedded Controller Modules (Continued)

Core Bus Monitoring

The core bus monitoring cycle is a non-data transfer bus cycle. It takes a single clock cycle - T1. On this cycle:

- The address pins display the address of the internal device accessed on the core bus.
- CBRD indicates the direction of the access (read or write).
- BE0-1 indicate which data bus bytes are accessed (lower or upper).
- BST0-2 display the core bus status.

The core bus monitoring cycle, as shown in Figure 26, is generated only when bit 1 (OBR) in BCFG register is 1.

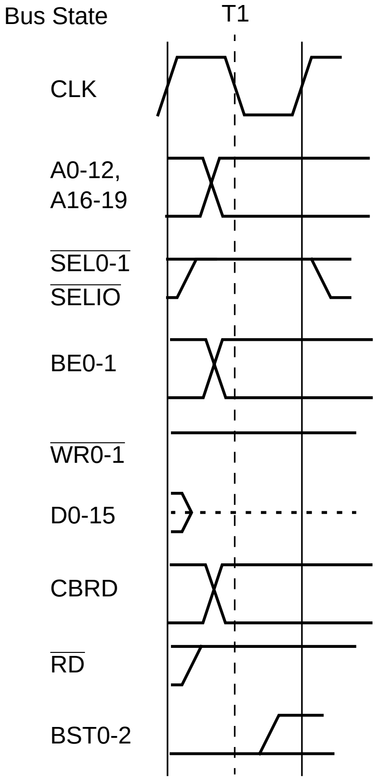

Bus State
T1
CLK
A0-12,
A16-19
$\overline{\text{SEL0-1}}$
$\overline{\text{SELIO}}$
BE0-1
$\overline{\text{WR0-1}}$
D0-15
CBRD
$\overline{\text{RD}}$
BST0-2

Figure 26. Core Bus Monitoring Bus Cycle

4.0 Embedded Controller Modules (Continued)

4.1.10 BIU Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

BIU Register Map

Mnemonic	Register Name	Type
BCFG	BIU Configuration	R/W
IOCFG	I/O Zone Configuration	R/W
SZCFGn	Static Zone Configuration	R/W

BIU Configuration Register (BCFG)

The BCFG register controls the configuration of common features to all zones. On reset, this register is initialized to 07_{16}.

Location: Index 00 $F980_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	Reserved					ISTL	OBR	EWR
Reset	0	0	0	0	0	1	1	1

Bit	Description
0	**EWR (Early Write).** 0: Late write 1: Early write
1	**OBR (Observability).** Determines if the address and status of internal accesses are observable. 0: Not observable (no toggle of external buses) 1: Observable (external buses toggle)
2	**ISTL (Internal Stall).** Determines if the internal bus is stalled while the BIU is busy. 0: Internal bus activity not stalled when BIU is busy 1: Stall internal bus activity when the external bus is busy
7-3	**Reserved.**

4.0 Embedded Controller Modules (Continued)

I/O Zone Configuration Register (IOCFG)

The IOCFG register controls the configuration of the I/O zone. On reset, it is initialized to $069F_{16}$.

Location: 00 $F982_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**						**IPST**	**Res**	**BW**	**Reserved**		**HOLD**		**WAIT**		
Reset	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1	1

Bit	Description
2-0	**WAIT.** Number of TIW clock cycles that extend the bus cycle. **Bits** **2 1 0 Number** 0 0 0: None 0 0 1: One 0 1 0: Two 0 1 1: Three 1 0 0: Four 1 0 1: Five 1 1 0: Six 1 1 1: Seven
4-3	**HOLD.** Number of T_{hold} clock cycles. **Bits** **4 3 Number** 0 0: None 0 1: One 1 0: Two 1 1: Three
6-5	**Reserved.**
7	**Bus Width.** Sets the external bus width used for the I/O zone. It is initialized during reset to its default value. 0: 8-bit bus 1: 16-bit bus (default)
8	**Reserved.**
9	**IPST (Idle After Bus Cycle).** Determines if an idle cycle follows the current bus cycle when the next bus cycle is in a different zone. 0: No idle cycle inserted 1: Idle cycle inserted
15-10	**Reserved.**

4.0 Embedded Controller Modules (Continued)

Static Zone Configuration Register (SZCFGn)

The SZCFGn register (where n = 0, 1 or 2) controls the configuration of zone n. On reset, SZCFGn is initialized to $069F_{16}$.

Location: Zone 0 - 00 $F984_{16}$
Zone 1 - 00 $F986_{16}$
Zone 2 - 00 $F988_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**				**FRE**	**IPRE**	**IPST**	**Res**	**BW**	**WBR**	**BRE**	**HOLD**		**WAIT**		
Reset	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1	1

Bit	Description
2-0	**WAIT.** Number of TIW clock cycles that extend the bus cycle. Ignored when bit 11 (FRE) of this register is set to 1. **Bits** **2 1 0 Number** 0 0 0: None 0 0 1: One 0 1 0: Two 0 1 1: Three 1 0 0: Four 1 0 1: Five 1 1 0: Six 1 1 1: Seven
4-3	**HOLD.** Number of T_{hold} clock cycles. **Bits** **4 3 Number** 0 0: None 0 1: One 1 0: Two 1 1: Three
5	**BRE (Burst Read Enable).** Ignored when bit 11 (FRE) of this register is set to 1. 0: Disabled 1: Enabled
6	**WBR (Wait on Burst Read).** Determines if a wait state (TBW) is added on a burst read transaction. 0: No TBW 1: TBW
7	**BW (Bus Width).** Sets the external bus width used for the I/O zone. It is initialized during reset to its default value. 0: 8-bit bus 1: 16-bit bus (default)
8	**Reserved.**
9	**IPST (Idle After Bus Cycle).** Determines if an idle cycle follows the current bus cycle when the next bus cycle is in a different zone. 0: No idle cycle inserted 1: Idle cycle inserted

Bit	Description
10	**IPRE (Idle Before Bus Cycle).** Inserts an idle cycle before the current bus cycle when this bus cycle is in a new zone. 0: No idle cycle inserted 1: Idle cycle inserted
11	**FRE (Fast Read Enable).** 0: Disabled - Normal read bus cycle takes at least two clock cycles 1: Enabled - Normal read bus cycle takes one clock cycle
15-12	**Reserved.**

4.1.11 Usage Hints

The following usage hints help configure the BIU to maximize PC87591x performance and avoid contention on the data bus.

1. In IRE environment, access time to the internal flash can use zero wait and zero hold cycles or fast reads depending on the CLK frequency set for the core and its peripherals (see “Flash Read, Write and Erase Time” on page 222).

   Flash sections 0 and 1 (fast zone) and Section 2 (slow zone) can use a fast read bus cycle through the operation frequency of the PC87591x; therefore, program SZCFG1 fields to be: WAIT=000, HOLD=00, BRE=0, WBE=0, BW=1, FRE=1.

   When Section 2 (slow zone) can operate with a fast read bus cycle, program SZCFG2 fields to be: WAIT=000, HOLD=00, BRE=0, WBE=0, BW=1, FRE=1.

   When Section 2 (slow zone) needs to operate with normal read and zero wait, program SZCFG2 fields to be: WAIT =000, HOLD=00, BRE=0, WBE=0, BW=1, FRE=0.

2. To avoid contention on the data bus when a read bus cycle (no T_{hold} clock cycles) in one zone is followed by a read bus cycle in another zone, program IPST and IPRE in the different memory (I/O) zones as follows:

Zone	IPRE	IPST
Zone 0	0	0
Zone 1	1	0
Zone 2	0/1[1]	0
Zone I/O	1[2]	0

[1] Set IPRE when the zone is configured for fast read.

[2] An IPRE is forced always for Zone I/O.

4.2 DMA CONTROLLER (DMAC)

The DMAC transfers blocks of data between memory and I/O devices along four independent channels, with minimal intervention by the core. The source and destination addresses and the block size to be transferred may be defined for each of the channels.

4.2.1 Features

- Four Independent Direct Memory Access (DMA) channels.
- Direct (fly-by) and indirect (memory-to-memory) transfer types.
- Single-buffer, double-buffer and auto initialize operation modes.
- Fixed address (I/O device) or updated (memory device).
- Address update (increment or decrement) independent of the number of transferred bytes.
- Interrupt line for each channel.
- Programmable bus policy for each channel: continuous or intermittent use of the bus.
- Software DMA request for each channel.
- Maximum throughput in direct (fly-by) transfer:
  - Intermittent:Every three clock cycles.
  - Continuous:On internal core bus - every clock cycle.
  - Otherwise:Every two clock cycles.
- Maximum throughput in indirect (memory-to-memory) transfer:
  - Intermittent:Every five clock cycles.
  - Continuous:On internal core bus - every two clock cycles.
  - Otherwise:Every four clock cycles.

4.2.2 Functional Description

When transferring blocks of data using the DMAC, the source and destination addresses, as well as the block size and type of operation, are set up in advance by programing the appropriate control registers. Actual data transfers are handled by the DMAC channel in response to DMA transfer requests. On receiving a DMA transfer request ($\overline{\text{DMRQn}}$), if the channel is enabled, the DMAC performs the following operations:

1. Acquires control of the core bus according to the DMAC priority on the core bus.
2. Determines priority among the DMAC channels, one clock cycle before T1 of the DMAC transfer cycle. (T1 is the first clock cycle of the bus cycle.) Priority among the DMAC channels is fixed in descending order, with Channel 0 receiving the highest priority.
3. Executes data transfer bus cycle(s) according to the values stored in the control registers of the channel being serviced and according to the accessed memory address. It acknowledges the request during the bus cycle that accesses the requesting device.
4. If the transfer of a block is terminated, the DMAC does the following:
   - Updates the termination flags.
   - Generates an interrupt if enabled.
   - Goes to step 6.
5. If $\overline{\text{DMRQ}}$n is still active and the Bus Policy is “continuous”, returns to step 3.
6. Relinquishes the internal core bus.

Each DMAC channel can be programed for direct (fly-by) or indirect (memory-to-memory) data transfer. Once a DMAC transfer cycle is in process, the next transfer request is sampled when the DMAC acknowledge is deactivated and subsequently, on the rising edge of each clock cycle.

The configuration of either address freeze or address update (increment or decrement) is independent of the number of transferred bytes, transfer direction or number of bytes in each DMAC transfer cycle. All these can be configured for each channel by programing the appropriate control registers.

Each DMAC channel has eight control registers. DMAC channels are described hereafter with the suffix “n”, where n represents the channel number in the register name (n = 0 to 3).

4.2.3 Channel Assignment in PC87591x

Table 15 shows the assignment of the DMA channels to different tasks in the PC87591x. If a channel is used for memory block transfers, other resources must be disabled.

Embedded Controller Modules (Continued)

Table 15. DMA Channel Assignment

Channel	Usage	Mode	Enable Control	Interrupt
0	USART Receive	Indirect	USART registers	DMA_INT0
1	USART Transmit	Indirect	USART registers	DMA_INT1
2	Reserved	Indirect		
3	Reserved	Indirect		

4.2.4 Transfer Types

The DMAC uses two data transfer modes, direct (fly-by*)* and indirect (memory-to-memory). The choice of mode depends on the correlation between the source and destination bus lengths, the required bus performance and the peripheral structure (as indicated in Table 15).

Direct (Fly-By) Transfers

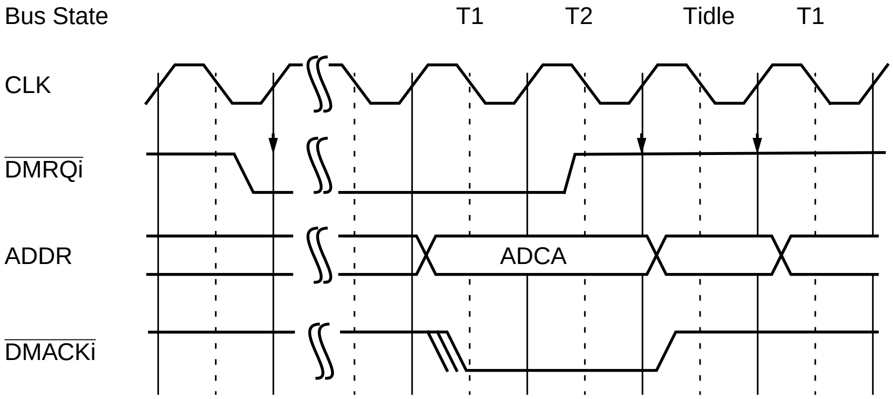


Figure 27. DMAC Direct Bus Cycle Followed by a Core Bus Cycle

In Direct mode, each data item is transferred using a single bus cycle without reading the data into the DMAC. It provides the fastest transfer rate, but it requires identical source and destination bus widths.

Data transfer cannot occur between two memory elements. One of the elements must be the I/O device that requested the DMA transfer. This device is referred to as the implied I/O device. The other element can be either memory or another I/O device and is referred to as the addressed I/O device.

The appropriate DMA acknowledge signal for each channel is asserted during the bus cycle.

If the bus policy is “intermittent”, maximum throughput is one transaction every three clock cycles. If bus policy is “continuous”, maximum throughput on the internal core bus is one transaction every two clock cycles.

Since only one address is required in Direct mode, this address is taken from the corresponding ADCAn counter. The DMAC Channel generates either a read or a write bus cycle according to the setting of DIR bit in DMACNTLn register.

When DIR bit is 0, a read bus cycle from the addressed device is performed, and the data is written to the implied I/O device. When DIR bit is 1, a write bus cycle to the addressed device is performed, and the data is read from the implied I/O device.

The configuration of either address freeze or address update (increment or decrement) is independent of the number of transferred bytes, transfer direction, or number of bytes in each DMAC transfer cycle. All these can be configured for each channel by programing the appropriate control register.

The number of bytes transferred in each cycle is taken from TCS bit in DMACNTLn register. After the data item has been transferred, the BLTCn counter is decremented by one. The ADCAn counter are updated according to INCAn field and ADA bit in DMACNTLn register.

Embedded Controller Modules (Continued)

Indirect (Memory-to-Memory) Transfers

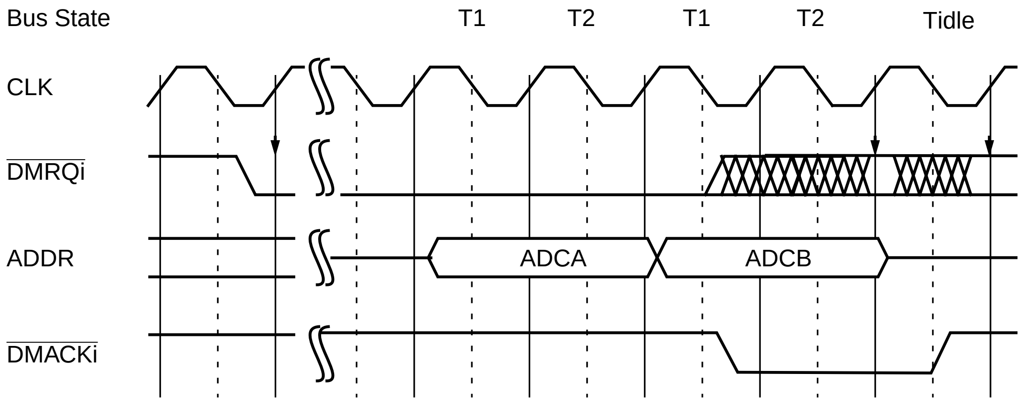


Figure 28. Indirect Bus Cycle (DIR=0) - External Bus

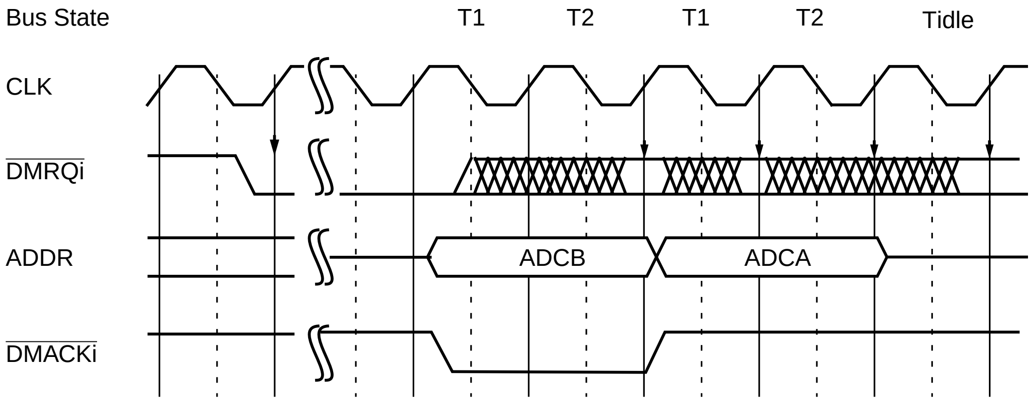


Figure 29. Indirect Bus Cycle (DIR=1) - External Bus

In Indirect (Memory-to-Memory) mode, data transfers use two consecutive bus cycles. The data is first read into a temporary register and subsequently written into the destination. This mode is slower than Direct (Fly-By) mode, but it provides support for different source and destination bus widths.

Indirect mode also facilitates block transfers between two memory elements. Each element is an addressed device and can be either memory or an I/O device.

The appropriate DMA acknowledge signal for each channel is asserted during the Device B bus cycle.

If the bus policy is “intermittent”, maximum, throughput is every five clock cycles. If the bus policy is “continuous”, maximum throughput is every two clock cycles on the internal core bus (otherwise, it uses four clock cycles).

When DIR bit is 0, the first bus cycle reads data from the source using the ADCAn counter, and the second bus cycle writes the data into the destination using the ADCBn counter. When DIR bit is 1, the first bus cycle reads data from the source using the ADCBn counter, and the second bus cycle writes the data into the destination using the ADCAn counter.

The number of bytes transferred in each cycle is taken from TCS bit in DMACNTLn register. After the data item has been transferred, the BLTCn counter is decremented by one. The ADCAn and ADCBn counters are updated according to INCAn, INCBn, ADA and ADB fields in DMACNTLn register.

Note:

For transfer operations between two memory areas, see “Software DMA Request” on page 92.

Embedded Controller Modules (Continued)

4.2.5 Bus Policy

Intermittent Operation Mode

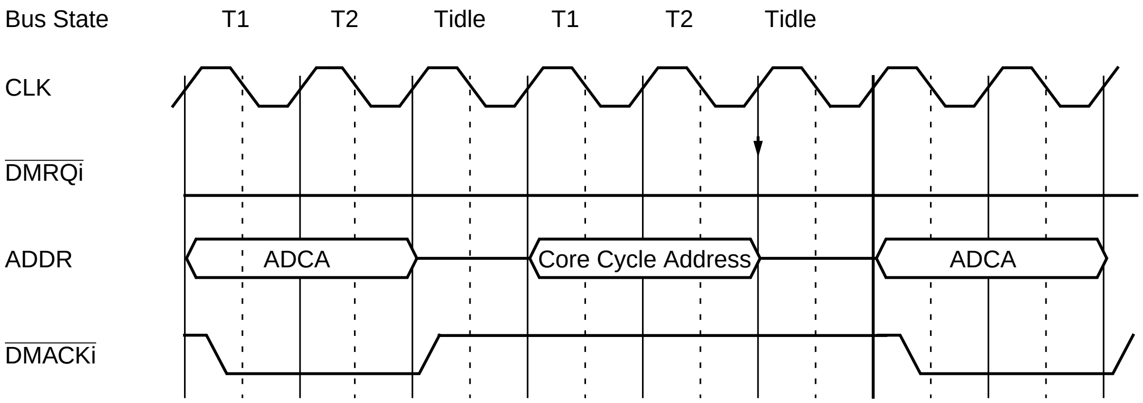


Figure 30. DMAC Direct Bus Cycles in Intermittent Mode, $\overline{\text{DMRQ}}$ Asserted Constantly.

When BPC bit in DMACNTLn is 0, channel n is in Intermittent mode. In this mode, the DMAC channel relinquishes the bus after each transaction, regardless of the state of its DMA request input. In this way, the DMAC gives the core (and other DMA channels) a chance to use the bus, even if a DMA device needs the bus for multiple transfers.

Continuous Operation Mode

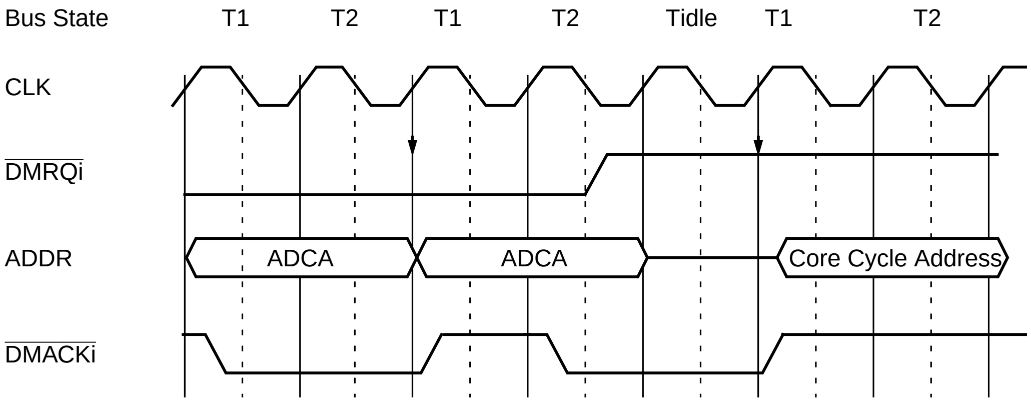


Figure 31. DMAC Direct Bus Cycles in Continuous Mode - External Bus

When BPC bit in DMACNTLn register is 1, channel n is in Continuous mode. In this mode, a DMAC channel uses the bus continuously, as long as its request is active and BLTCn > 0. This allows the channel to utilize the full bandwidth of the bus.

The activity of this channel cannot be interrupted by any other internal bus master, including higher priority DMAC channels. It is the system designer’s responsibility to limit the duration of the DMA request to prevent bus starvation.

4.2.6 Operation Modes

The DMAC operates in three different block transfer modes - single transfer, double buffer and auto-initialize. Select the appropriate mode according to the character of the block transfer.

Single Transfer Operation

This mode provides the simplest way to accomplish a single block data transfer.

Initialization

1. Write the two block transfer addresses and byte count into the corresponding ADCAn, ADCBn and BLTCn counters, respectively. The BLTCn counter should be written last.
2. Program the OT bit for Non-Auto-Initialize mode, and clear EOVR bit in DMACNTLn register to 0. Clear to 0 VLD bit in DMASTATn register.
3. Set CHEN bit in DMACNTLn register to 1; the channel activates and responds to DMAC transfer requests.

Termination

When the BLTCn counter reaches 0:

- The transfer operation terminates.
- TC bit in DMASTATn register is set to 1, and CHAC is cleared to 0.
- A level interrupt is generated (if enabled by ETC bit in DMACNTLn register).

Double Buffer Operation

This mode allows the software to set up the next block transfer specification while the current block transfer proceeds. This mode is used for preparing the next buffer for use in a multi-buffer operation (e.g., the alternate buffer in a double-buffer scheme).

Initialization

1. Write the two block transfer addresses and byte count into the ADCAn, ADCBn and BLTCn counters, respectively. The BLTCn counter should be written last.
2. Program OT bit in DMACNTLn register for Non-Auto-Initialize mode.
3. Set CHEN bit in DMACNTLn register to 1; the channel activates and responds to DMAC transfer requests.
4. While the current block transfer proceeds, write the addresses and byte count for the next block into the ADRAn, ADRBn and BLTRn registers. The BLTRn register should be written last.

Continuation / Termination

When the BLTCn counter reaches 0:

- TC bit in DMASTATn register is set to 1.
- A level interrupt is generated (if enabled by ETC bit in DMACNTLn register).
- The DMAC channel checks the value of VLD bit.

If VLD bit is 1:

- The channel copies the ADRAn, ADRBn and BLTRn values into ADCAn, ADCBn and BLTCn. The BLTCn counter should be written last.
- Clears VLD bit to 0.
- Becomes ready to start the next block transfer (on the next DMA request).

If VLD bit is 0:

- The transfer operation terminates.
- The channel sets OVR bit in DMASTATn register to 1.
- Clears CHAC bit to 0.
- A level interrupt is generated (if enabled by EOVR bit in DMACNTLn register).

Note:

ADCB and ADRBn are used only in indirect (memory-to-memory*)* transfer. In Direct (Fly-By) mode, the DMAC does not use them and therefore does not copy ADRBn into ADCBn.

Auto-Initialize Operation

This mode allows the DMAC to continuously fill the same memory area without software intervention.

Initialization

1. Write the two block addresses and byte count into the ADCAn, ADCBn and BLTCn counters, respectively (the BLTCn counter should be written last); also write them to the ADRAn, ADRBn and BLTRn registers, respectively (the BLTRn counter should be written last).
2. Program OT bit in DMACNTLn register for Auto-Initialize mode.
3. Set CHEN bit in DMACNTLn register to 1; the channel activates and responds to DMAC transfer requests.

Continuation

When the BLTCn counter reaches 0:

- The contents of the ADRAn, ADRBn and BLTRn registers are copied to the ADCAn, ADCBn and BLTCn counters, respectively. The BLTCn counter should be written last.
- The DMAC channel checks the value of TC bit.

If TC bit is 1:

- OVR bit in DMASTATn register is set to 1.
- A level interrupt is generated (if enabled by EOVR bit in DMACNTLn register).
- The operation is repeated.

If TC bit is 0:

- TC bit in DMASTATn register is set to 1.
- A level interrupt is generated (if enabled by ETC bit in DMACNTLn register).
- The operation is repeated.

4.2.7 Software DMA Request

In addition to the $\overline{\text{DMRQn}}$ signals, a DMA transfer request can also be initiated by software. The software DMA transfer request is used for memory-to-memory block copying (in indirect transfers).

When SWRQ bit in DMACNTLn register is 1, the corresponding DMA channel receives a DMA transfer request. When SWRQ bit is 0, the software DMA transfer request of the corresponding channel is inactive.

For each channel, use the software DMA transfer request, only when the corresponding $\overline{\text{DMRQn}}$ signal is inactive.

4.2.8 DMAC Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

DMAC Register Map

Mnemonic	Register Name	Type
ADCAn	Device A Address Counter Register	R/W
ADRAn	Device A Address Register	R/W
ADCBn	Device B Address Counter Register	R/W
ADRBn	Device B Address Register	R/W
BLTCn	Block Length Counter Register	R/W
BLTRn	Block Length Register	R/W
DMACNTLn	DMA Control Register	R/W
DMASTATn	Status Register	R/W

Note:

Register names with the suffix n, where n = 0 to 3, are duplicated for each channel.

Unless stated otherwise, bits 21 to 31 are reserved in each of the following registers. Double-word (32-bit) registers may be accessed word-by-word (word aligned).

Embedded Controller Modules (Continued)

Device A Address Counter Register (ADCAn)

A double-word, read/write register. Holds the current address of either the source data item or the destination location, according to DIR bit in DMACNTLn register. ADCAn is updated after each transfer cycle by INCAn, INCBn, ADA and ADB in DMACNTLn register.

Location: Channel 0 - 00 FA00$_{16}$
Channel 1 - 00 FA20$_{16}$
Channel 2 - 00 FA40$_{16}$
Channel 3 - 00 FA60$_{16}$

Type: R/W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Device A Address Counter**																															

Device A Address Register (ADRAn)

A double-word, read/write register. Holds the starting address of either the next source data block, or the next destination data area, according to DIR bit in DMACNTLn register.

Location: Channel 0 - 00 FA04$_{16}$
Channel 1 - 00 FA24$_{16}$
Channel 2 - 00 FA44$_{16}$
Channel 3 - 00 FA64$_{16}$

Type: R/W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Device A Address**																															

Device B Address Counter Register (ADCBn)

A double-word, read/write register. Holds the current address of either the source data item, or the destination location, according to DIR bit in DMACNTLn register. ADCBn is updated after each transfer cycle by INCAn, INCBn, ADA and ADB in DMACNTLn register. In Direct (Fly-By) mode, this register is not used.

Location: Channel 0 - 00 FA08$_{16}$
Channel 1 - 00 FA28$_{16}$
Channel 2 - 00 FA48$_{16}$
Channel 3 - 00 FA68$_{16}$

Type: R/W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Device B Address Counter**																															

Device B Address Register (ADRBn)

A double-word, read/write register. Holds the starting address of either the next source data block or the next destination data area, according to DIR bit in DMACNTLn register. In Direct (Fly-By*)* mode, this register is not used.

Location: Channel 0 - 00 FA0C$_{16}$
Channel 1 - 00 FA2C$_{16}$
Channel 2 - 00 FA4C$_{16}$
Channel 3 - 00 FA6C$_{16}$

Type: R/W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Device B Address**																															

Embedded Controller Modules (Continued)

Block Length Counter Register (BLTCn)

A double-word, read/write register. Holds the current number of DMA transfers to be executed in the current block. BLTCn is decremented by one after each transfer cycle. A DMA transfer may consist of one or two bytes according to TCS bit in DMACNTLn register.

Location: Channel 0 - 00 FA10$_{16}$
Channel 1 - 00 FA30$_{16}$
Channel 2 - 00 FA50$_{16}$
Channel 3 - 00 FA70$_{16}$

Type: R/W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**																**Block Length Counter**															

Note: Writing 0 to Block Length Counter field of BLTCn initializes the DMA for 2^{21}-1 transfers.

Block Length Register (BLTRn)

A double-word, read/write register. Holds the number of DMA transfers to be executed in the next block. Writing this register, sets VLD bit in DMASTATn register to 1.

Location: Channel 0 - 00 FA14$_{16}$
Channel 1 - 00 FA34$_{16}$
Channel 2 - 00 FA54$_{16}$
Channel 3 - 00 FA74$_{16}$

Type: R/W

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**																**Block Length**															

Note: writing 0 to Block Length field of BLTRn initializes the DMA for 2^{21}-1 transfers.

DMA Control Register (DMACNTLn)

A word-wide, read/write register that synchronizes the channel's operation with the programing of the block transfer parameters. On reset, the implemented bits are initialized to 0.

The format of the DMACNTLn register is shown below.

Location: Channel 0 - 00 FA1C$_{16}$
Channel 1 - 00 FA3C$_{16}$
Channel 2 - 00 FA5C$_{16}$
Channel 3 - 00 FA7C$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8
Name	**Reserved**	**INCB**		**ADB**	**INCA**		**ADA**	**SWRQ**
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	**BPC**	**OT**	**DIR**	**IND**	**TCS**	**EOVR**	**ETC**	**CHEN**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**CHEN (Channel Enable).** Must be set to enable DMA operation on this channel 0: Channel disabled 1: Channel enabled If CHEN bit in DMACNTLn register is cleared in all channels, the DMA clock is disabled to preserve power.

Embedded Controller Modules (Continued)

Bit	Description
1	**ETC (Enable Interrupt on Terminal Count).** Enables a level interrupt, when TC bit is set. 0: Interrupt masked 1: Interrupt enabled
2	**EOVR (Enable Interrupt on OVR).** Enables a level interrupt, when OVR bit is set. 0: Interrupt masked 1: Interrupt enabled
3	**TCS** (**Transfer Cycle Size).** Specifies the number of bytes transferred in each DMA transfer cycle. In Direct (Fly-By) mode, undefined results occur if TCS is not equal to the addressed memory bus width. 0: Byte wide transfer 1: Word-wide (16-bit) transfer
4	**IND (Direct/Indirect Transfer).** Transfer Type. 0: Direct (Fly-By) 1: Indirect (Memory-to-Memory)
5	**DIR (Transfer Direction).** Specifies the direction of the transfer relative to Device A. 0: Device A (pointed to by ADCAn) is the source. In Fly-By mode, a read transaction is initialized. 1: Device A (pointed to by ADCAn) is the destination. In Fly-By mode, a write transaction is initialized.
6	OT **(Operation Type).** 0: Single-Buffer mode or Double-Buffer mode enabled 1: Auto-Initialize mode enabled
7	**BPC (Bus Policy Control).** Intermittent (cycle stealing) or continuous (burst). 0: Intermittent operation. DMAC channel n relinquishes the bus after each transaction even if the request is still asserted. 1: Continous operation. DMAC channel n uses the bus continuously as long as the request is asserted. This mode can only be used for SW DMA requests (i.e., when SWRQ is set). On HW DMA requests, BPC must be set to 0.
8	**SWRQ (Software DMA Request).** 0: Software DMA request is inactive 1: Software DMA request is active
9	**ADA (Device A Address Control).** Enables Update of Device A Address. 0: ADCAn address unchanged 1: ADCAn address incremented or decremented, according to INCA field
11-10	**INCA (Increment/Decrement ADCAn).** **Bits** **11 10 Description** 0 0: Increment ADCAn register by 1 0 1: Increment ADCAn register by 2 1 0: Decrement ADCAn register by 1 1 1: Decrement ADCAn register by 2
12	**ADB (Device B Address Control).** Enables Update of Device B Address. 0: ADCBn address unchanged 1: ADCBn address incremented or decremented, according to INCB field
14-13	**INCB (Increment/Decrement ADCBn).** **Bits** **14 13 Description** 0 0: Increment ADCBn register by 1 0 1: Increment ADCBn register by 2 1 0: Decrement ADCBn register by 1 1 1: Decrement ADCBn register by 2
15	**Reserved.**

Embedded Controller Modules (Continued)

DMA Status Register (DMASTATn)

A byte-wide, read with write 1 to clear register that holds the status information for the DMAC channel. On reset, the implemented bits are initialized to 0. The reserved bits always return zero when read.

The format of the DMASTATn register is shown below.

Location: Channel 0 - 00 FA1E$_{16}$
Channel 1 - 00 FA3E$_{16}$
Channel 2 - 00 FA5E$_{16}$
Channel 3 - 00 FA7E$_{16}$

Type: R/W1C

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**				**VLD**	**CHAC**	**OVR**	**TC**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**TC**[1] **(Terminal Count).** When set to 1, indicates that the transfer was completed by a terminal count condition (BLTCn register reached 0).
1	**OVR**[1] (**Channel Overrun).** • In double buffered operation (OT bit in DMACNTLn register is 0): OVR is set to 1 when the present transfer is completed (BLTC = 0), but the parameters for the next transfer (address and block length) are not valid. • In auto initialize operation: (OT bit in DMACNTLn register is 1) OVR is set to 1 when the present transfer is completed (BLTC = 0), but TC bit is still set to 1 (e.g., the software did not serve the last interrupt). • In single buffer operation: This bit is ignored.
2	**CHAC (Channel Active).** Continuously reflects the active or inactive status of the channel and is therefore read only. Data written to CHAC bit is ignored. 0: Indicates that the channel is inactive 1: Indicates that the channel is active (CHEN bit in DMACNTLn register is 1 and BLTC > 0)
3	**VLD**[1] **(Transfer Parameters Valid).** Specifies whether the transfer parameters for the next block to be transferred are valid. Writing to the BLTRn register sets this bit to 1. It is cleared to 0 in the following cases: • The present transfer is completed, and the ADRAn, ADRBn (Indirect mode only) and BLTR registers are copied to the ADCAn, ADCBn (Indirect mode only) and BLTCn registers, respectively. • Writing 1 to VLD bit; (writing zero has no effect).
7-4	**Reserved.**

1. The VLD, OVR and TC bits are sticky (once set by the occurrence of the specific condition, they remain set until explicitly cleared by software). These bits can be cleared individually by writing a value into the DMASTATn register with the bit positions to be cleared set to 1; writing 0 to these bits has no effect.

4.2.9 Usage Hints

- Do not write to ADCAn, ADCBn or BLTCn and do not change the value of TCS, IND, DIR, OT, ADA, INCA, ADB and INCB fields of DMACNTLn register while the associated channel is active (CHAC bit in DMASTATn register is 1). When initializing these registers, write to BLTCn register last, since writing to BLTCn register activates the channel immediately (if CHEN bit in DMACNTLn register is set to 1).
- The ADRAn, ADRBn and BLTRn registers store transfer parameters (source address, destination address and block length) for the next data block to be transferred, for either Auto-Initialize or Double-Buffer modes of operation. When initializing these registers, write the BLTRn register last, since this validates the next block's parameters (VLD bit in DMASTATn register is set to 1).
- The TCS bit in DMACNTLn register is programed according to the bus width of the devices. It determines how many bytes are transferred in each DMA bus cycle.
- The DMAC does not support non-aligned transfers. The values written to ADCAn, ADRAn, ADCBn and ADRBn must be multiples of the Transfer Cycle Size (as defined by TCS bit in DMACNTLn register).

4.3 INTERRUPT CONTROL UNIT (ICU)

The ICU has 31 channels. It interfaces between the different modules' interrupt requests and external interrupt requests and also generates the core interrupt. It generates both maskable and non-maskable interrupts. The ICU has a predetermined scheme that allocates priority.

4.3.1 Features

Non-Maskable Interrupts (NMI)

- Gathers all edge-triggered non-maskable interrupt sources
  - — External Power Fail (PFAIL) interrupt source
- Holds the status of the current pending NMI requests
- Generates non-maskable interrupt (NMI) to the core

Maskable Interrupts

- 31 active high-level or edge-triggered interrupt sources
- Core vectored interrupt mode
- Fixed priority among interrupt sources
- Individual enable/disable for each interrupt source
- Polling support by an interrupt status register
- Clear registers for edge-triggered interrupts

4.3.2 Non-Maskable Interrupt (NMI)

The Interrupt Control Unit (ICU) gathers external non-maskable interrupt (NMI) sources and generates an NMI interrupt to the core when required.

External NMI Inputs

The ICU processes the $\overline{\text{PFAIL}}$ signal to send to the CR16B NMI input.

Non-Maskable Interrupt Processing

The CR16B core performs an "Interrupt Acknowledge" bus cycle when beginning to process a non-maskable interrupt. The address associated with this core bus cycle is within the internal core address space and may be monitored as a Core Bus Monitoring (CBM) clock cycle. For additional details, see "Core Bus Monitoring" on page 82 and Section 4.20.6 on page 273.

After reset, NMI is disabled and must remain disabled until the software initializes the interrupt table, interrupt base and the interrupt mode.

The PFAIL interrupt is enabled by setting ENLCK bit and remains enabled until a reset occurs. This allows the external NMI feature to be enabled only after the Interrupt Base Register (IMASK) and the Interrupt Stack Pointer (ISP), in the core, have been set up.

Alternatively, the external PFAIL interrupt can be enabled by setting EN bit, which remains enabled until an interrupt event or a reset occurs. The NMISTAT register holds the status of the current pending NMI request. When the bit in NMISTAT is set to 1, an NMI request to the core is issued. NMISTAT is cleared each time its contents are read. NMI handlers must read the NMISTAT register to allow new NMI events to occur.

Note that PFAIL status bit in NMISTAT register may be set as a result of transient enable conditions on $\overline{\text{PFAIL}}$. To avoid an interrupt to the core, after configuring the $\overline{\text{PFAIL}}$ input for operation, read the NMISTAT register and then enable $\overline{\text{PFAIL}}$ by writing 1 to EN bit in PFAIL register.

$\overline{\text{PFAIL}}$ Input

The $\overline{\text{PFAIL}}$ signal is an asynchronous input with Schmitt trigger characteristics and an internal synchronization circuit; therefore, no external synchronizing circuit is needed. The $\overline{\text{PFAIL}}$ signal generates an interrupt on its falling edge.

4.3.3 Maskable Interrupts

The ICU receives level or edge-triggered interrupt request signals from 31 sources and generates a vectored interrupt to the CR16B core when required. Priority among the interrupt sources (named INT1 to INT31) is fixed. Each interrupt source can be individually enabled or disabled under software control through:

- ICU interrupt enable registers
- Interrupt enable bits in the peripherals that request the interrupts.

Embedded Controller Modules (Continued)

Pending interrupts, enabled or disabled, can be polled using the Status registers. The CR16B core supports INT0, but the ICU reserves INT0 so that it is not connected to any interrupt source.

Maskable Interrupt Vectors

Interrupt vector numbers are always positive and are in the range 10_{16} to $2F_{16}$. The IVCT register contains the interrupt vector of the enabled and pending interrupt with the highest priority. The interrupt vector 10_{16} corresponds to INT0 with the lowest priority; the vector $2F_{16}$ corresponds to INT31 with the highest priority.

The CR16B core performs an “Interrupt Acknowledge” bus cycle on receiving an enabled maskable interrupt request from the ICU. During the interrupt acknowledge cycle, a byte is read from address 00 $FE00_{16}$ (IVCT register). The byte read is used as an index in the Dispatch Table to determine the address of the interrupt handler.

Although INT0 is not connected to any interrupt source, the IVCT register can return the value 10_{16}. This happens, for example, when the interrupt request is removed before the interrupt acknowledge cycle. The entry in the Dispatch Table should point to a default interrupt handler that handles this error condition.

Pending Interrupts

Edge-triggered interrupts are latched by the Interrupt Status register. A pending edge-triggered interrupt is cleared by writing a ‘1’ to the respective bit in the Edge Interrupt Clear register, IECLR0 or IECLR1.

A pending level-triggered interrupt is cleared when the interrupt source is not active; note that the interrupt should be cleared at the device/module that drives it by clearing the event status bit.

Interrupt mask bits (IENAM register bits) and pending interrupt bits (ISTAT register bits), should be cleared to 0 only when interrupts are disabled; i.e., when bits I and/or E in PSR register (a core register) are 0. Bits in IENAM may be set at any time.

Interrupt Priorities

The priorities of the maskable interrupts are hard-wired and thus fixed. The interrupts are named INT0 to INT31, where INT0 has the lowest priority and INT31 has the highest priority.

Power-Down Modes

Interrupt sources that may generate unexpected interrupts in Idle mode should be masked before switching to the power-down mode.

External Interrupt Inputs

When an MIWU input is disabled, and the respective WKOxx output at the MIWU is connected to the ICU, the MIWU input is fed directly to the ICU. In this case, the interrupt inputs are asynchronous. They are recognized by the PC87591x during cycles in which the input setup and hold time requirements are satisfied. To use an external interrupt which is shared with an I/O port, configure the I/O port to its alternate function (see Section 2.4 on page 54).

Interrupt Assignment

Table 16 shows the mapping of the ICU maskable interrupts to different functions. For information on mask bits and the clear mechanism for the status bit in internal level interrupts, refer to descriptions of the module that is the interrupt source.

Table 16. ICU Interrupt Assignments

INT Number	Source	Type	Details	Priority
INT0	-	-	Error condition occurred (spurious interrupt)	Lowest
INT1	External/MIWU	Level-High	External Interrupt EXWINT20 through the MIWU[1]	
INT2	Internal	Level-High	Host I/F Keyboard/Mouse channel Output Buffer Empty	
INT3	Internal	Level-High	Host I/F Power Management channel 1 or channel 2 Output Buffer Empty	
INT4	Internal	Level-High	High-Frequency Clock Generator	
INT5	Internal	Level-High	MIWU PSWUINT or WKINTD	
INT6	External/MIWU	Level-High	External Interrupt EXWINT23 through the MIWU[1]	
INT7	Internal	Level-High	MFT16 1 Interrupt (INT1 ORed with INT2)	

Embedded Controller Modules (Continued)

Table 16. ICU Interrupt Assignments (Continued)

INT Number	Source	Type	Details	Priority
INT8	Internal	Level-High	ADC interrupt (ADCI)	
INT9	Internal	Level-High	ACCESS.bus 1 interrupt	
INT10	Internal	Level-High	ACCESS.bus 2 interrupt	
INT11	MIWU	Level-High	Internal Keyboard Scan Interrupt (KBSINT from MIWU) or ACM Interrupt	
INT12	Internal/MIWU	Level-High	MSWC interrupt through the MIWU	
INT13	Internal/MIWU	Edge Rising	TWM system tick (T0OUT), through the MIWU[2]	
INT14	External/MIWU	Level-High	SWIN input[1], through the MIWU	
INT15	Reserved			
INT16	Internal	Level-High	USART Interrupt (TX Int OR RX int)	
INT17	External/MIWU	Level-High	External Interrupt EXWINT24 through the MIWU[1]	
INT18	Internal	Edge Falling	PS/2 shift mechanism (PSINT3)	
INT19	Internal	Edge Falling	PS/2 shift mechanism (PSINT2)	
INT20	Internal	Level High / Edge Falling[3]	PS/2 shift mechanism (PSINT1)	
INT21	External/MIWU	Level-High	External Interrupt EXWINT22 through the MIWU[1]	
INT22	Internal	Level-High	MFT16 2 Interrupt (INT1 ORed with INT2)	
INT23	Internal	Level-High	Flash I/F interrupt or Shared Memory Interrupt	
INT24	Internal	Level-High	Host I/F Keyboard/Mouse channel Input Buffer Full	
INT25	Internal	Level-High	Host I/F Power Management channel 1 or channel 2 Input Buffer Full	
INT26	Reserved			
INT27	Internal	Level-High	DMA Channel 0	
INT28	Internal	Level-High	DMA Channel 1	
INT29	Reserved			
INT30	Reserved			
INT31	External/MIWU	Level-High	External Interrupt EXWINT21 through the MIWU[1]	Highest

1. To enable the external interrupt, set the pin to its alternate function. When used as I/O port signals the External interrupt input is forced to 0
2. When in Active mode, it is advised to disable the T0OUT channel of the MIWU, this saves the need to clear the pending bit in the MIWU on each interrupt.
3. Falling Edge is enabled for this input when the PS/2 channel ‘Disabled Shift Mechanism Interrupts’ are enabled (DSMIE in PSIEN register = 1)

4.3.4 ICU Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

ICU Register Map

Mnemonic	Register Name	Type
IVCT	Interrupt Vector	RO
NMISTAT	NMI Status	RO
PFAIL	Power Fail Interrupt Control and Status	R/W
ISTAT0	Interrupt Status 0	RO
ISTAT1	Interrupt Status 1	RO
IENAM0	Interrupt Enable and Mask 0	R/W
IENAM1	Interrupt Enable and Mask 1	R/W
IECLR0	Edge Interrupt Clear 0	WO
IECLR1	Edge Interrupt Clear 1	WO

Interrupt Vector Register (IVCT)

The IVCT register holds the vector number of the interrupt vector. IVCT is set to 10_{16} on reset.

Location: 00 $FE00_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**0**	**0**	**INTVECT**					
Reset	0	0	0	1	0	0	0	0

Bit	Description
5-0	**INTVECT (Interrupt Vector).** Contains the encoded value of the enabled pending interrupt with the highest priority; the valid values range from 10_{16} to $2F_{16}$. Valid during an interrupt acknowledge core bus cycle in which IVCT is read. It may contain invalid data when INTVECT is updated.
7-6	These bits return 0 when read.

NMI Status Register (NMISTAT)

The NMISTAT register holds the status of the current pending Non-Maskable Interrupt (NMI) requests. This register is cleared on reset and each time its contents are read. Refer to the description of the PFAIL register below for additional details.

Location: 00 $FE02_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**							**EXT**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**EXT (External Non-Maskable Interrupt Request).** 0: No external non-maskable interrupt request occurred 1: External non-maskable interrupt request occurred
7-1	**Reserved.**

Embedded Controller Modules (Continued)

Power Fail Interrupt Control and Status Register (PFAIL)

The PFAIL register holds the current value of the PFAIL signal and controls the NMI interrupt generation based on a falling edge of the PFAIL signal. EN and ENLCK are cleared on reset. When writing to this register, all reserved bits must be written with 0 for the device to function properly.

Location: 00 FE04$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**ENLCK**	**PIN**	**EN**
Reset	x	x	x	x	0	0	x	0

Bit	Description
0	**EN (PFAIL In**terr**upt Enable).** An NMI interrupt is generated when this bit is set to 1 and the PFAIL signal changes its value from high to low. The bit is cleared by hardware on reset and whenever the interrupt occurs (i.e., when EXT bit in NMISTAT register is set). It can be set and cleared by software; however, software can set this bit only when EXT is cleared. This bit is ignored when ENLCK is set. 0: No NMI interrupt generated 1: NMI interrupt generated
1	**PIN (PFAILPin Value).** Contains the current (non-inverted) PFAIL signal value. This bit is read only; data written to it is ignored.
2	**ENLCK PFAIL Interrupt Enable Lock.** When this bit is set to 1, the external PFAIL feature is enabled and locked; it cannot be cleared by software and can only be cleared by hardware on reset. After setting this bit, an NMI interrupt is generated every time the PFAIL signal changes its value from high to low. Note that when ENLCK is set, EN bit is ignored. 0: External PFAIL feature disabled 1: External PFAIL feature enabled and locked
7-3	**Reserved.**

Interrupt Status Register 0 (ISTAT0)

This register indicates which maskable interrupts are pending regardless of the state of the corresponding IENA bits. ISTAT0 is cleared on reset.

Location: 00 FE0A$_{16}$

Type: RO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IST15-0**															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description
15-0	**IST15-0 (Interrupt Status).** Each bit indicates if an interrupt event was sent to the ICU; IST15 to IST0 correspond to INT15 to INT0, respectively. Since INT0 is not used, IST0 always reads 0. Each bit is encoded as follows: 0: Interrupt input to ICU not pending 1: Interrupt input to ICU pending

Embedded Controller Modules (Continued)

Interrupt Status Register 1 (ISTAT1)

This register indicates which maskable interrupts are pending regardless of the state of the corresponding IENA bits.

Location: 00 FE0C$_{16}$

Type: RO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IST31-16**															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description
15-0	**IST31-16 (Interrupt Status).** Each bit indicates if an interrupt event was sent to the ICU; IST31 to IST16 correspond to INT31 to INT16, respectively. Each bit is encoded as follows: 0: Interrupt input to ICU not pending 1: Interrupt input to ICU pending

Interrupt Enable and Mask Register 0 (IENAM0)

This register controls the enable/disable of the maskable interrupt sources INT0 to INT15. The register is cleared (0000$_{16}$) on reset.

Location: 00 FE0E$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IENA15-0**															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description
15-0	**IENA15-0 (Interrupt Enable).** Each bit enables or disables the corresponding interrupt request INT0 to INT15; e.g. IENA15 controls INT15. Since INT0 is not used, IENA0 has no effect on the operation of the ICU. 0: Interrupt disabled 1: Interrupt enabled

Interrupt Enable and Mask Register 1 (IENAM1)

This register controls the enable/disable of the maskable interrupt sources INT16 to INT31. The register is cleared (0000$_{16}$) on reset.

Location: 00 FE10$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IENA31-16**															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description
15-0	**IENA31-16 (Interrupt Enable).** Each bit enables or disables the corresponding interrupt request INT16 to INT31; e.g. IENA16 controls INT16. 0: Interrupt disabled 1: Interrupt enabled

Embedded Controller Modules (Continued)

Edge Interrupt Clear Register 0 (IECLR0)

The IECLR register is used to clear pending, edge-triggered interrupts.

Location: 00 FE12$_{16}$

Type: WO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IEC15-1**															**Res.**

Bit	Description
0	**Reserved.**
15-1	**IEC15-1 (Edge Interrupt Clear).** Each bit clears the corresponding bit in ISTAT0 register. Writing to the bit positions of level-triggered interrupts has no effect. Read always returns FFFF$_{16}$. IEC15 to IEC1 correspond to INT15 to INT1, respectively. Each bit is encoded as follows: 0: No effect 1: Pending edge-triggered interrupt cleared

Edge Interrupt Clear Register 1 (IECLR1)

The IECLR register is used to clear pending, edge-triggered interrupts.

Location: 00 FE14$_{16}$

Type: WO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IEC31-16**															

Bit	Description
15-0	**IEC31-16 (Edge Interrupt Clear).** Each bit clears the corresponding bit in ISTAT1 register. Writing to the bit positions of level-triggered interrupts has no effect. Read always returns FFFF$_{16}$. IEC31 to IEC16 correspond to INT31 to INT16, respectively. Each bit is encoded as follows: 0: No effect 1: Pending edge-triggered interrupt cleared

4.3.5 Usage Hints

Initializing

The recommended initialization sequence is:

1. Initialize both the INTBASE register and the interrupt stack pointer of the core.
2. Prepare the interrupt routines of the interrupts used.
3. Clear edge interrupt used.
4. Set relevant bits of the peripherals.
5. Set relevant bits in IENAM register.
6. Set PFAIL register.
7. Enable core interrupt.

Clearing

Clearing an interrupt request before it is serviced may cause a spurious interrupt (i.e., when the core detects an interrupt not reflected by IVCT). Clear interrupt requests only when interrupts are disabled. Clear IENAM bits and ISTAT bits while the core interrupts are disabled (i.e., bits I and/or E in PSR register are cleared).

Nesting

The IENAM registers can be used in interrupt handlers to allow interrupt nesting. When the core acknowledges an interrupt, it disables maskable interrupts by clearing bit I in PSR register and executes the interrupt service routine. This routine can enable nested interrupts by setting bit I in PSR register and can use the IENAM registers to control which interrupts are allowed.

4.4 MULTI-INPUT WAKE-UP (MIWU)

The Multi-Input Wake-Up Unit (MIWU) allows the PC87591x to exit Idle mode. In addition, it provides signal conditioning and grouping of external interrupt sources. It supports a total of 32 internal and/or external wake-up sources.

4.4.1 Features

- Supports up to 32 internal and/or external wake-up inputs
- Generates a wake-up signal
- Generates interrupt signals for:
  - — each of the 32 inputs
  - — one interrupt for each group of eight inputs e.g., I/O ports
- User-selectable trigger condition on each input:
  - — positive edge
  - — negative edge
- Individual enable and pending bits for each input
- Programmable bypass mode connects inputs to ICU without MIWU involvement

4.4.2 Operation

Overview

The Multi-Input Wake-Up Unit (MIWU) detects a valid software-selectable trigger condition on any of its inputs. On detection of a valid trigger condition, the MIWU generates a wake-up request and/or an interrupt request. The wake-up request is connected to the Power Management module (PMC) and may be utilized to exit the Idle mode and return to Active mode. The interrupt requests are used to signal to the Interrupt Control Unit (ICU) that either an edge-triggered external or internal interrupt condition has occurred. Figure 32 shows a block diagram of the Multi-Input Wake-Up module.

Note that the MIWU module is active while in Idle mode. Note, however, that because all device clocks are stopped in Idle mode, the detection of a trigger condition on an input, and the resulting set of the pending flag, are not synchronous to the system clock.

Table 17 lists the MIWU sources and interrupts used in the PC87591x.

Table 17. Input Assignments

Source		Destination	
Name	**MIWU Input**	**Interrupt Name**	**MIWU Output**
PSCLK1	WUI10	PSWUINT	WKINTA
PSCLK2	WUI11		
PSCLK3	WUI12		
PSDAT1	WUI13		
PSDAT2	WUI14		
PSDAT3	WUI15		
PSCLK4	WUI16		
PSDAT4	WUI17		
EWXINT20	WUI20	INT1	WKO20
EXWINT21	WUI21	INT31	WKO21
EXWINT22	WUI22	INT21	WKO22
EXWINT23	WUI23	INT6	WKO23
EXWINT24	WUI24	INT17	WKO24
SWIN	WUI25	INT14	WKO25
MSWC Wake-Up	WUI26	INT12	WKO26
T0OUT[1]	WUI27	T0OUTINT	WKO27

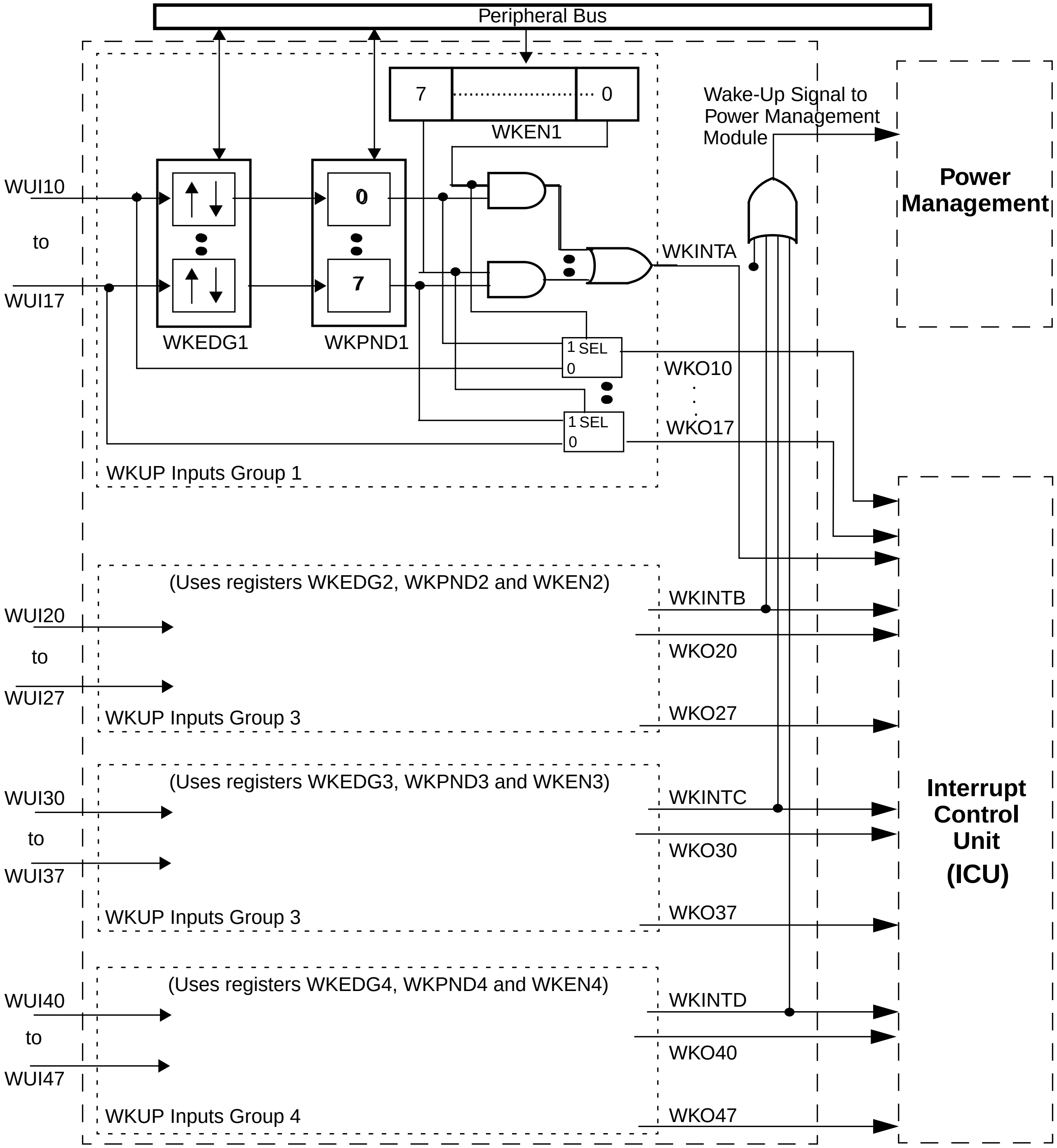
Embedded Controller Modules (Continued)

Table 17. Input Assignments (Continued)

Source		Destination	
Name	**MIWU Input**	**Interrupt Name**	**MIWU Output**
KBSIN0	WUI30	KBSINT	WKINTC
KBSIN1	WUI31		
KBSIN2	WUI32		
KBSIN3	WUI33		
KBSIN4	WUI34		
KBSIN5	WUI35		
KBSIN6	WUI36		
KBSIN7	WUI37		
EXWINT40	WUI40	MIWU2	WKINTD
Reserved	WUI41		
Host Access Wake-Up[2]	WUI42		
ACCESS.bus 1 Wake-Up[1]	WUI43		
ACCESS.bus 2 Wake-Up[1]	WUI44		
EXWINT45	WUI45		
EXWINT46	WUI46		
Analog Comparators (ACMI)	WUI47		

1. Program the input to detect the rising edge of the input event.
2. The wake-up input is triggered by a host access to one of a selected set of devices in the PC87591x. See “Core Interrupt” on page 343. Configure the WUI26 for rising edge detection.

Embedded Controller Modules (Continued)



Note: Not all MIWU-ICU connections are implemented; for details, see ICU interrupt assignments in Table 17 on page 105.

Figure 32. Multi-Input Wake-Up Block Diagram

Trigger Conditions

Through the WKEDGx registers, the trigger condition on the selected input signal can be selected as either positive edge (low-to-high transition) or negative edge (high-to-low transition).

Embedded Controller Modules (Continued)

Pending Flags

An occurrence of a trigger condition for the Multi-Input Wake-Up input is latched into the respective pending bit in WKPNDx register. The respective bits of WKPNDx are set on an occurrence of the selected trigger edge on the corresponding input signal.

Since the WKPNDx register holds a pending wake-up condition until it is cleared, the device does not enter the IDLE mode if any wake-up bit is both enabled and pending. Consequently, the pending flags must be cleared before an attempt to enter the Idle mode.

Input Enable

The MIWU utilizes multiple Multi-input Wake-Up signals. Setting the appropriate bits in the WKENAx registers select which particular wake-up signal causes the device to exit Idle mode.

Interrupts

The combined output of all pending and enabled channels of the MIWU module generates the wake-up signal, which is fed into both the Power Management Control module (PMC) and the Interrupt Control Unit (ICU). Therefore, each wake-up of the device can be followed by a wake-up interrupt. Since the device can not enter the power reduction mode without having the core execute a “WAIT” instruction, a wake-up interrupt is needed to terminate the “WAIT” instruction on wake-up.

MIWU outputs WKO10 through WKO37 may be connected to the Interrupt Control Unit (ICU) to generate an interrupt associated with the specific MIWU output. The WKOxx behaves as follows:

- When the respective WKENxx bit is cleared, the WUIxx is connected to the ICU directly (bypassing the edge detectors and pending bits). The ICU can be configured to use the signal as a level or edge triggered interrupt.
- When the respective WKENxx bit is enabled, the output of the pending bit, WKPDxx, is connected to WKOxx.

Note: To enable and disable ICU interrupts generated by their associated WKOxx output of the MIWU, use the ICU mask register in the ICU.

In addition, the MIWU provides four interrupt request lines: WKINTA, WKINTB, WKINTC and WKINTD (see Figure 32 on page 107). These are routed to the ICU (except WKINTB) and can request an interrupt if a valid trigger condition occurred on any of the enabled input sources within the group of eight inputs associated with the interrupt line. For a detailed summary of the interrupts available, see Table 17 on page 105.

4.4.3 MIWU Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

MIWU Register Map

Mnemonic	Register Name	Type
WKEDG1	Edge Detection Register	R/W
WKEDG2	Edge Detection Register	R/W
WKEDG3	Edge Detection Register	R/W
WKEDG4	Edge Detection Register	R/W
WKPND1	Pending Register	R/W
WKPND2	Pending Register	R/W
WKPND3	Pending Register	R/W
WKPND4	Pending Register	R/W
WKEN1	Enable Register	R/W
WKEN2	Enable Register	R/W
WKEN3	Enable Register	R/W
WKEN4	Enable Register	R/W
WKPCL1	Pending Clear Register	WO
WKPCL2	Pending Clear Register	WO
WKPCL3	Pending Clear Register	WO
WKPCL4	Pending Clear Register	WO

Edge Detection Register (WKEDG1)

Byte-wide read/write register that configures the trigger condition of the input signals WUI10 to WUI17. The register is cleared on reset. This configures all associated input signals to be triggered on a rising edge.

Location: 00 FFC0$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**WKED17-WKED10**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Edge Selection.** For inputs WUI10 through WUI17. Each bit is associated with one of eight inputs. 0: Low-to-High transition. 1: High-to-Low transition.

Embedded Controller Modules (Continued)

Edge Detection Register (WKEDG2)

Byte-wide read/write register that configures the trigger condition of the input signals WUI20 to WUI27. The functionality of the register is identical to the WKEDG1 register described above.

Location: 00 FFC2$_{16}$

Type: R/W

Edge Detection Register (WKEDG3)

Byte-wide read/write register that configures the trigger condition of the input signals WUI30 to WUI37. The functionality of the register is identical to the WKEDG1 register described above.

Location: 00 FFC4$_{16}$

Type: R/W

Edge Detection Register (WKEDG4)

Byte-wide read/write register that configures the trigger condition of the input signals WUI40 to WUI47. The functionality of the register is identical to the WKEDG1 register described above.

Location: 00 FFC6$_{16}$

Type: R/W

Pending Register (WKPND1)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI10 to WUI17. On reset, the WKPND1 register is cleared (0). This indicates that no occurrence of the selected trigger condition is pending.

Note: Only software can set the register bits; only the WKPCL1 register can clear them. Writing a 0 to any of the bits leaves their values unchanged. The WKPND1 register format is shown below:

Location: 00 FFC8$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**WKPD17-WKPD10**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Wake-Up Pending.** If set, (1) indicates that a valid trigger condition occurred on the associated input.

Pending Register (WKPND2)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI20 to WUI27. For a detailed description of the register see the above description of the WKPND1 register.

Location: 00 FFCC$_{16}$

Type: R/W

Pending Register (WKPND3)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI30 to WUI37. For a detailed description of the register see the above description of the WKPND1 register.

Location: 00 FFD0$_{16}$

Type: R/W

Embedded Controller Modules (Continued)

Pending Register (WKPND4)

Byte-wide read/write register that latches the occurrence of a selected trigger condition associated with the input signals WUI40 to WUI47. For a detailed description of the register see the above description of the WKPND1 register.

Location: 00 FFD4$_{16}$

Type: R/W

Enable Register (WKEN1)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI10 to WUI17. On reset, WDENA1 is cleared (0); this disables the associated input signals. The WKENA1 register format is shown below:

Location: 00 FFD8$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**WKEN17-WKEN10**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Wake-Up Enable.** If set (1), a valid trigger condition on the associated input generates a wake-up signal or EXTINTx interrupt request

Enable Register (WKEN2)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI20 to WUI27. For a detailed description of the register, see the above description of the WKEN1 register.

Location: 00 FFDA$_{16}$

Type: R/W

Enable Register (WKEN3)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI30 to WUI37. For a detailed description of the register, see the above description of the WKEN1 register.

Location: 00 FFDC$_{16}$

Type: R/W

Enable Register (WKEN4)

Byte-wide read/write register that enables the wake-up function of the associated input signal, WUI40 to WUI47. For a detailed description of the register, see the above description of the WKEN1 register.

Location: 00 FFDE$_{16}$

Type: R/W

Pending Clear Register (WKPCL1)

Byte-wide write-only register that controls the clearing (0) of the pending bits associated with the WUI10 through WUI17 inputs. This avoids potential hardware/software collisions during read-modify-write operations. The WKPCL1 register format is shown below:

Location: 00 FFCA$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**WKCL17-WKCL10**							

Bit	Description
7-0	**Clear Pending Flag.** If a 1 is written to any bit, the associated pending flag located in WKPND1 is cleared (0). Writing a 0 to any bit leaves the value of the corresponding pending flag unchanged.

Embedded Controller Modules (Continued)

Pending Clear Register (WKPCL2)

Controls the clearing (0) of the pending bits associated with the WUI20 through WUI27 inputs. For a detailed description of the register see, the above description of the WKPCL1 register.

Location: 00 $FFCE_{16}$

Type: WO

Pending Clear Register (WKPCL3)

Controls the clearing (0) of the pending bits associated with the WUI30 through WUI37 inputs. For a detailed description of the register, see the above description of the WKPCL1 register.

Location: 00 $FFD2_{16}$

Type: WO

Pending Clear Register (WKPCL4)

Controls the clearing (0) of the pending bits associated with the WUI40 through WUI47 inputs. For a detailed description of the register, see the above description of the WKPCL1 register.

Location: 00 $FFD6_{16}$

Type: WO

4.4.4 Usage Hints

1. When changing an edge select, perform the following steps to avoid a spurious wake-up condition, which may occur as a result of the edge change:
   a. Clear the associated WKENxx bit.
   b. Select the required the edge in the WKEDGx register.
   c. Clear the associated WKPDxx bit.
   d. Re-enable the associated WKENxx bit.
2. The correct use of the Multi-Input Wake-Up circuit, which avoids false triggering of a wake-up condition, requires the following sequence of actions. Use the same procedure following a Reset since the wake-up inputs are left floating, producing unknown data on the MIWU input signals.
   a. If the input originates from an I/O port, write to the port alternate function and, if required, direction register to set the pin to interrupt inputs.
   b. Clear the WKENAx register or, if a WKOxx interrupt is used, disable the interrupt via the ICU.
   c. Write the WKEDGx register to select the desired type of edge sensitivity for each of the pins used.
   d. Clear the WKPNDx register to cancel any pending bits.
   e. Either set the WKENxx bits associated with the pins to be used, thus enabling them for the wake-up/interrupt function, or re-enable the interrupt via the ICU.
3. On Reset, the WKEDGx registers are configured to select positive edge sensitivity for all wake-up inputs. To change the edge sensitivity of an input signal while preventing the false triggering of a wake-up/interrupt condition, use the following procedure.
   a. Clear the WKENxx bit associated with the WUIxx input to disable that input.
   b. Write to the WKEDGx register to select the new type of edge sensitivity for the specific input.
   c. Clear the WKPDxx bit associated with the WUIxx input.
   d. Set the WKENxx bit associated with the WUIxx input to re-enable it.

4.5 GENERAL-PURPOSE I/O (GPIO) PORTS

The PC87591x includes four types of General-Purpose I/O (GPIO) ports: Px, Py, Pz and Pw.

- Px signals: Each signal is bidirectional and can be configured as input or output. An internal weak pull-up is provided to hold the pin high when used as an input or in an open-drain scheme.
- Py signals: Each signal is input only. An internal weak pull-up is provided to hold the pin high.
- Pz signals: Each signal is output only. It may be configured to work as totem-pole or in an open-drain scheme.
- Pw signals: Each signal is bidirectional and can be configured for input or output. The Pw pins may be shared with development system functions. These ports can be implemented off-chip in DEV environment using external logic.

The GPIO signals are organized in ports. Each port is either 8-bits or 16-bits wide. In ports where not all eight bits are used, some of the register's bits are reserved. Some GPIO signals share their pins with one or more alternate functions. A configuration bit selects which function is active (see Section 2.4 on page 54).

Note: Some GPIO signals are only available in a 176-pin package.

GPIO Port Functionality

The PC87591x provides 117 GPIO pins in the 176-pin package and 84 GPIO pins in the 128-pin package. They are subdivided into the following groups:

- Ports IOPA(7-0), IOPB(7-0), IOPC(7-0), IOPD(7-0) and IOPF(7-0)
  These ports are on-chip, General-Purpose Input/Output (GPIO) ports (type Px).

  IOPC0 is reserved for power supply control use. Bit 0 of PCALT, PCDIR, PCWPU and PCDOUT are reset on V_{CC} Power-Up reset and WATCHDOG reset only.

  PB5 and PB6 have an option for automatic TRI-STATE based on $\overline{\text{LPCPD}}$. See "MSWC Control Status Register 3 (MSWCTL3)" on page 330 for the enable function.

  PB6 is selected to its alternate function, by default (i.e., bit 6 is set to 1).

  Ports IOPA4-0, IOPB2-0, IOPC0 and IOPD3 have the option to echo the value of the associated input. For the exact echo matrix specifications, see Section 2.4.3 on page 59.

  Bit 5 of PBALT register, bit 0 of PCALT register and bits 4-7 of PDALT register are read only (RO) and return a value of zero.
- IOPE(7-0) and KBSIN(7-0)
  These are General-Purpose Input (GPI) ports (type Py). IOPE(3-0) and IOPE5 do not implement the pull-up function, and the respective bits in PEWPU are reserved. KBSIN has no alternate function; thus it has no PyALT register.
- KBSOUT(15-0)
  This is a 16-bit General-Purpose Output (GPO) port (type Pz). Since KBSOUT has no alternate functions, its alternate function register is not implemented. The reset value of KBSOUT register is $FFFF_{16}$. KBSOUT has open-drain output drivers.
- Ports IOPH(7-0), IOPI(7-0), IOPJ(7-0), IOPK(7-0),IOPL(4-0) and IOPM(7-0)
  These ports are on-chip, General-Purpose Input/Output (GPIO) ports (type Pw).
- When the analog function is enable, the Read function, for GPIO signals that are multiplexed with analog functions, is disabled; this affects signals KBSIN(7-0) and IOPE(3-0).
  When the BIU function is enabled, the Read function, for GPIO signals that are multiplexed with BIU signals, is disabled; this affects signals PH(7-0), PI(7-0), PJ(1-0), PK(7-0), PL(7-0) and PM(7-0).

4.5.1 Features

- General-Purpose Input/Output (GPIO) Port (Px).
  - — Each pin functions as input or output signal.
  - — Direction register controls the port direction.
  - — Weak pull-up.
  - — Read-back on all registers.
- General-Purpose Input (GPI) Port (Py).
  - — All port pins function as input signals.
  - — Weak pull-up.
  - — Read-back on all control registers.
- General-Purpose Output (GPO) Port (Pz).
  - — All pins function as output signals.
  - — Read-back on all registers.
- General-Purpose Input/Output (GPIO) Port (Pw).
  - — DEV environment support.
    - — Pins shared with DEV environment signals.
    - — Off-chip implementation supported in DEV environments.
    - — Binary and cycle-by-cycle compatibility between environments.
  - — Each pin functions as input or output signal.
  - — Direction register controls the port direction.
  - — Read-back on all registers.
- Each I/O pin can be configured as a GPIO port or as an alternate function.
  - — Some I/O or input pins can provide interrupt functions.

4.5.2 GPIO Port Px

Bidirectional Port with Alternate Function

The GPIO port enables access to input and output pins. It also controls the pin usage either as an I/O port or in its alternate function. Figure 33 illustrates this functionality.

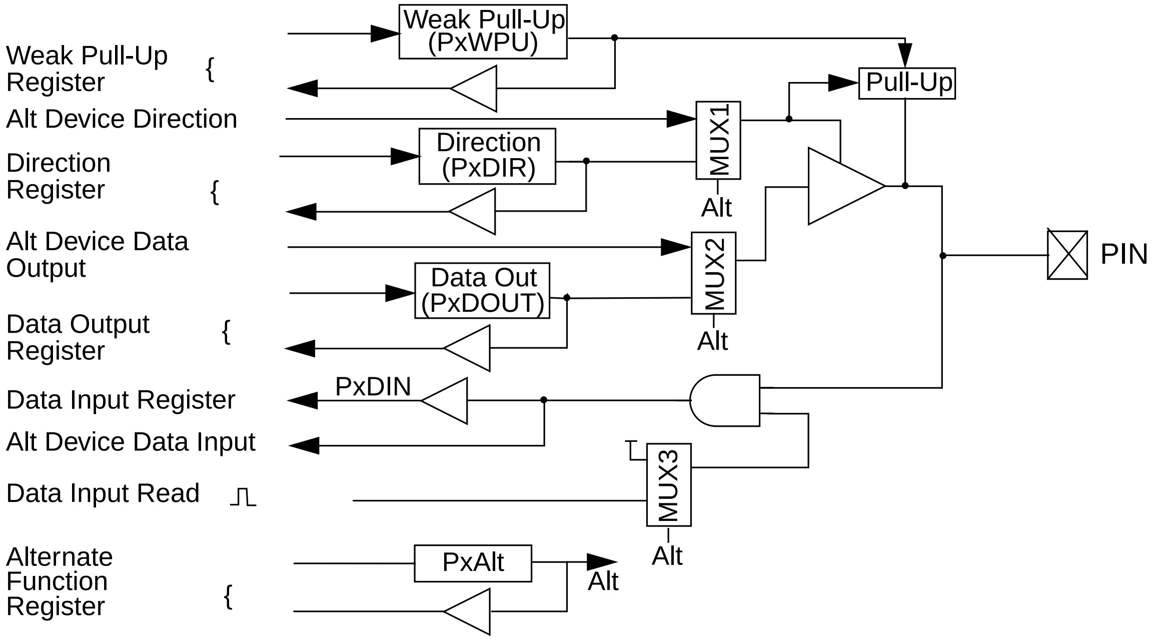


Figure 33. GPIO Port Px Schematic Diagram

Embedded Controller Modules (Continued)

Output Buffer

The output buffer is a TRI-STATE buffer. The output type (i.e., CMOS or TTL) and its driving capabilities are described in Section 2.2 on page 41.

Input Buffer

The I/O port input buffer characteristics are defined in Section 2.2 on page 41.

The input buffer has an enable input. When enabled, the buffer inputs the pin's logic level to the on-chip modules. When disabled, the input is blocked to prevent supply leakage currents.

Weak Pull-Up

The weak pull-up is enabled when the corresponding bit of PxWPU is set (1) and the pin is configured as an input port. When the pin is configured for input, this pull-up can prevent the input from being in an undefined state. When the pin is configured as an output port, this pull-up is disabled.

Alternate Function

The PxALT controls the use of each of the port pins for GPIO or for the pin's respective alternate function.

When PxALT bit is cleared (0):

- The corresponding pin is used as a GPIO pin.
- The output buffer is controlled by the Direction and Data Output registers.
- The input buffer is routed to the Data Input register.
  In this case, the input buffer is blocked, except when the buffer is actually being read.
- The pull-up is enabled when both the PxWPU is set and the device puts the output buffer in TRI-STATE.

When a bit in PxALT is set (1):

- The corresponding pin is used for an alternate function (i.e., a signal from/to some other PC87591x module).
- The output buffer data and TRI-STATE are controlled by signals from the alternate module.
- The input buffer is always enabled when the alternate function is an input; therefore, to minimize current consumption, the signal should be held above V_{CC}–0.2 or below GND+0.2V.
- The pull-up is enabled when PxWPU is set and the device puts the output buffer in TRI-STATE.

Port Direction

The Port Direction register (PxDIR) controls the direction of the port. If set (1), each bit in the register causes the corresponding port signal to serve as an output port, thus enabling the output buffer.

When cleared, the port serves as an input port signal, thus putting the output buffer in TRI-STATE.

If the corresponding bit in PxWPU is set, it also enables the pull-up.

Data Output

The Data Output (PxDOUT) register holds the data to be driven onto the pin, when the respective pin is configured as GPIO and its direction is set as output.

Data Input

The Data Input (PxDIN) register returns the current value/state of the pin. This register can always be read.

Open Drain

To use the GPIO pin as an inverting open-drain output buffer, the software should clear the corresponding bit in PxDOUT register and then use PxDIR register to set the value to the port pin.

When the signal direction is set as output (1), a value of 0 is forced. When the direction is set for input (0), the signal is in TRI-STATE and is not forced low.

The internal weak pull-up can pull the signal high when it is not forced low, by writing (1) to the corresponding bit of PxWPU.

Input Echo Function

Some of the Px pins can echo the value of an input pin. Figure 34 illustrates the modified structure of Px port pins that support the echo function. The input echo function may be used when the Px pin is configured to operate as a General-Purpose output pin.

Table 9 on page 60 defines pairs of input ports (Pi) and output ports (Po) and the Echo Enable bit associated with each pair. When the pair's Echo Enable bit is set, and if Po is configured as output, the value from the input bit (Pi) is output to the respective output port (Po). When the pair's Echo Enable bit is cleared, and if Po is configured as output, the value in PxDOUT is output to the respective output port.

Embedded Controller Modules (Continued)

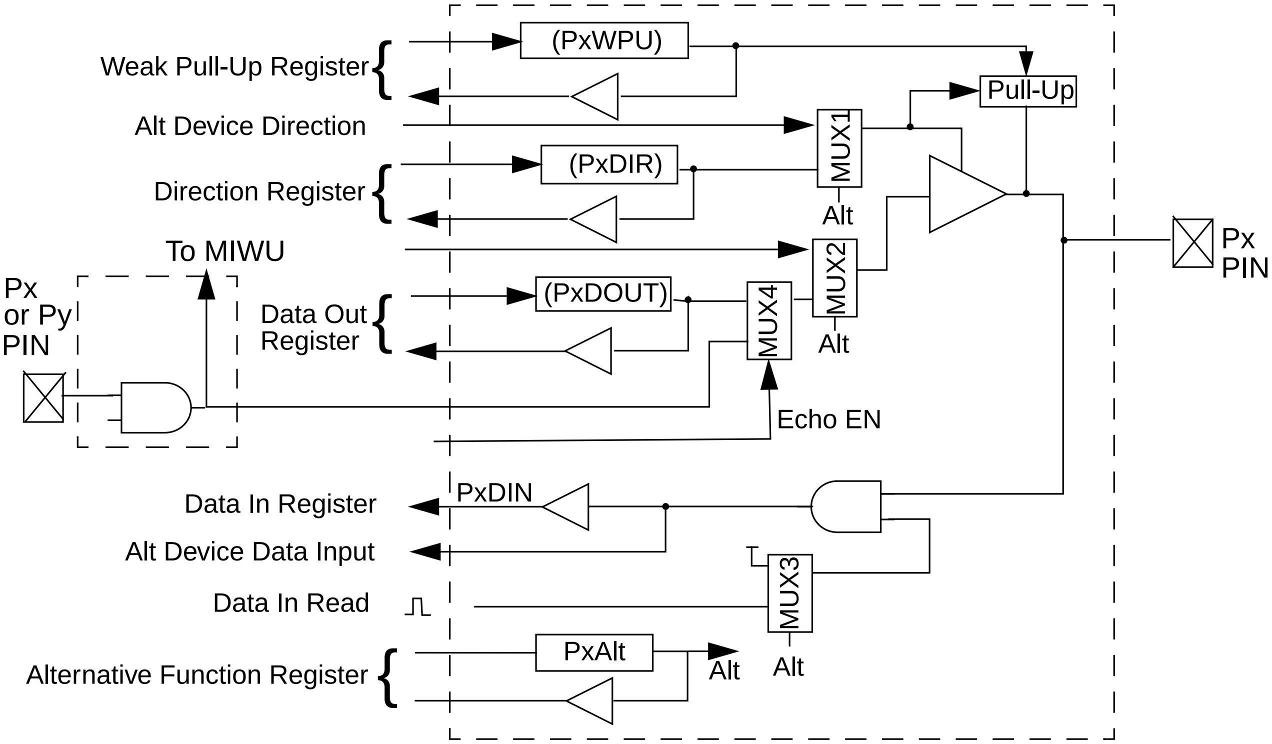


Figure 34. GPIO Port Px Output with Echo Schematic Diagram

4.5.3 GPI Port Py

Input Only Port with Alternate Function

The General-Purpose Input (GPI) port (Py) contains a subset of the GPIO functions. It can be used as a GPI port or as an input signal for an alternate function. Figure 35 illustrates its functionality.

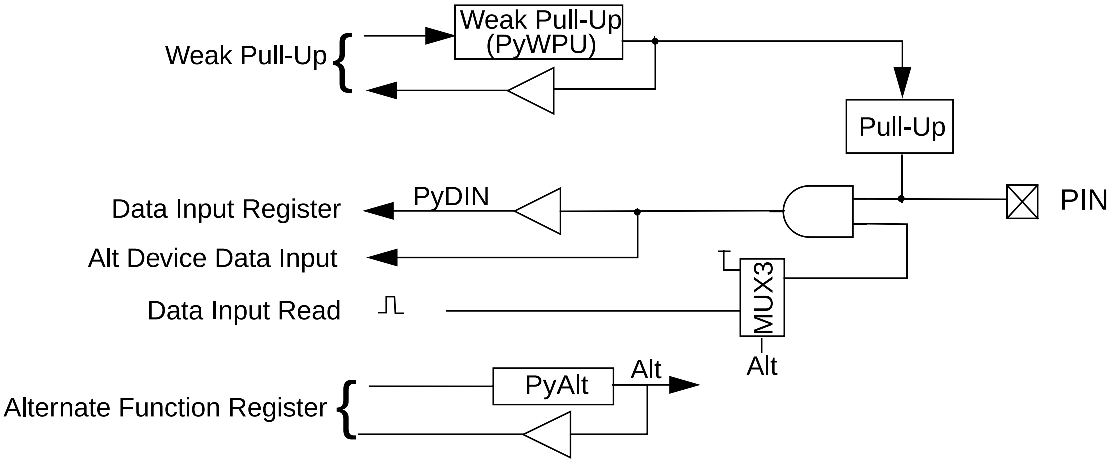


Figure 35. GPI Port Py Schematic Diagram

Embedded Controller Modules (Continued)

Input Buffer

The input buffer characteristics are defined in Section 2.2 on page 41. The input buffer has an enable input. When enabled, the buffer inputs the pin's logic level to the on-chip modules. When disabled, the input is blocked to prevent supply leakage currents.

Weak Pull-Up

The weak pull-up is enabled when the corresponding bit of PyWPU is set (1). This pull-up can prevent the input from being in an undefined state.

Alternate Function

PyALT controls the use of each of the port pins for GPI or for the pin's respective alternate function.

When a PxALT bit is cleared (0):

- The corresponding pin is used as a GPI pin.
- The input buffer is routed to the Data Input register.
  In this case, the input buffer is blocked, except when the buffer is actually being read.
- The pull-up is enabled when both the PyWPU is set and the output buffer is put in TRI-STATE.

When a bit in PxALT is set (1):

- The corresponding pin is used for an alternate function (i.e., a signal to some other PC87591x module).
- The input buffer is always enabled; therefore, to minimize current consumption, the signal should be held above V_{CC}–0.2 or below GND+0.2V.)
- The pull-up is enabled when both the PyWPU is set and the output buffer is put in TRI-STATE.

Data Input

The Data Input (PyDIN) register returns the current value/state of the pin. This register can always be read.

4.5.4 GPO Port Pz

Output Only Port with Alternate Function

The General-Purpose Output (GPO) port (Pz) is a subset of the GPIO functions. It enables the port to be used as a GPO port or as an output signal for an alternate function. Figure 36 illustrates its functionality.

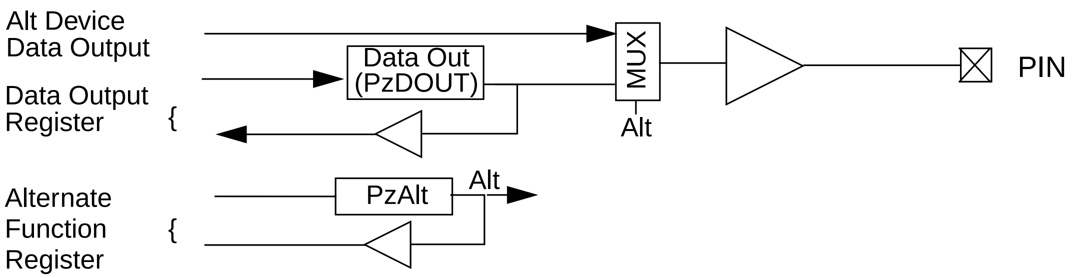


Figure 36. GPO Port Pz Schematic Diagram

Output Buffer

The output buffer is a TRI-STATE buffer. The output type (i.e., CMOS or TTL) and its driving capabilities are described in Section 2.2 on page 41.

Alternate Function

The PzALT controls the use of each of the port pins for GPO or for the pin's respective alternate function.

When a PzALT bit is cleared (0):

- The corresponding pin is used as a GPO pin.
- The output buffer is controlled by the Data Output register.

When a bit in PzALT is set (1):

- The corresponding pin is used for an alternate function (i.e., a signal from or to some other module of the PC87591x).
- The output buffer data is controlled by signals coming from the alternate module.

Embedded Controller Modules (Continued)

Data Output

The Data Out (PzDOUT) register holds the data to be driven onto the pin.

4.5.5 GPIO Port Pw

GPIO Port Signals Shared with Development System Signals

The Pw GPIO port enables access to input and output pins. Depending on the operating environments, some pins of the I/O ports may be dedicated to functions other than input or output. In this case, low-cost external logic can be used to perform the I/O functions with binary and cycle-by-cycle compatibility (see Section 4.1.8 on page 80 for details of the Expansion I/O protocol). The Alternate Functions Table (Table 8 on page 54) defines the environment in which each port is implemented off-chip and the pin that performs its alternate function.

The I/O Expansion protocol is used to access the off-chip implementation of the port's registers (PwDIR, PwDOUT, PwDIN), when the port pins are used by DEV environment. This enables binary compatibility between all environments.

To enable cycle-by-cycle compatibility in all environments, the access time to any of the registers is identical for on-chip and off-chip implementation of the ports (i.e., as configured for the BIU I/O zone).

Figure 37 illustrates its functionality.

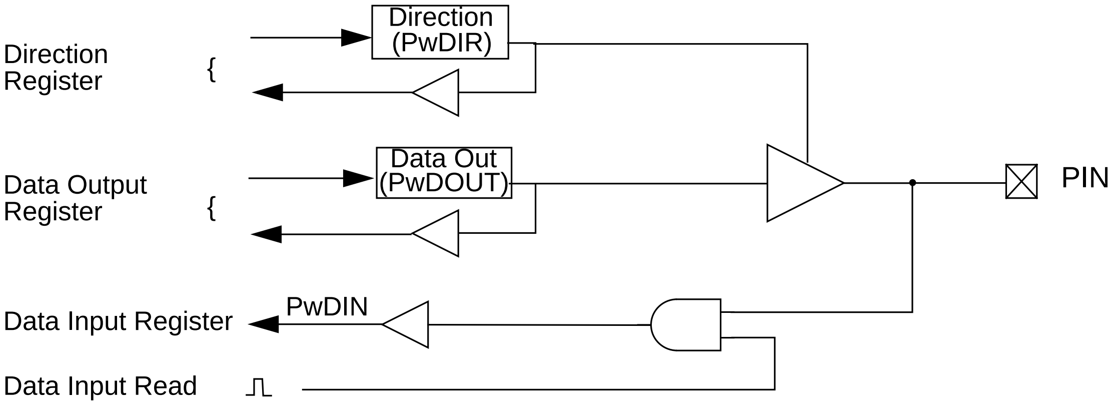


Figure 37. GPIO Port Pw Schematic Diagram

Output Buffer

The output buffer is a TRI-STATE buffer. Its output type (i.e., CMOS or TTL) and its driving capabilities are described in Section 2.2 on page 41.

Input Buffer

The I/O port input buffer characteristics are defined in Section 2.2 on page 41. The input buffer has an enable input. When enabled, the buffer inputs the pin's logic level to the on-chip modules. When disabled, the input is blocked to prevent supply leakage currents.

Port Direction

The Port Direction register (PwDIR) controls the direction of the port. When set (1), each bit in PwDIR register causes the corresponding port signal to serve as an output port, thus enabling the output buffer. When cleared, the port serves as an input port signal, thus putting the output buffer in TRI-STATE.

Data Output

The Data Output (PwDOUT) register holds the data to be driven onto the pin when the corresponding pin is set as GPIO and its direction is set as output.

Data Input

The Data Input (PwDIN) register returns the current value/state of the pin. This register can always be read.

Open Drain

To use the GPIO pin as an inverting open-drain output buffer, the software should clear the corresponding bit in Data Output (PwDOUT) register and then use the Direction register to set the value to the port pin.

When the signal’s direction is set as output (1), a value of 0 is forced. When the direction is set for input (0), the signal is in TRI-STATE and is not forced low.

4.5.6 GPIO Port Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

GPIO Register Map

Mnemonic	Register Name	Type
PxALT	Port Px Alternate Function	R/W
PyALT	Port Py Alternate Function	R/W
PzALT	Port Pz Alternate Function	R/W
PxDIR	Port Px Direction	R/W
PwDIR	Port Pw Direction	R/W
PxDOUT	Port Px Data Output	R/W
PzDOUT	Port Pz Data Output	R/W
PwDOUT	Port Pw Data Output	R/W
PxDIN	Port Px Data Input	RO
PyDIN	Port Py Data Input	RO
PwDIN	Port Pw Data Input	RO
PxWPU	Px Weak Pull-Up	R/W
PyWPU	Py Weak Pull-Up	R/W

Port Alternate Function Registers (PxALT, PyALT and PzALT)

These registers control the use of each of the Px, Py and Pz pins, respectively, as GPIO ports or as alternate functions.

- When cleared (0), each bit in PxALT, PyALT or PzALT enables the corresponding pin as a GPIO signal.
- When set (1), each bit enables the corresponding pin as an alternate function.

These registers are cleared (0) on reset, except when otherwise noted in Appendix A on page 408.

Location: See Appendix A

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Px Pins Alternate Function Enable**							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	**Py Pins Alternate Function Enable**							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	**Pz Pins Alternate Function Enable**							
Reset	0	0	0	0	0	0	0	0

Embedded Controller Modules (Continued)

Port Direction Registers (PxDIR and PwDIR)

These registers configure the direction of the Px and Pw pins.

- When cleared (0), each bit in PxDIR or PwDIR defines the corresponding pin as input.
- When set (1), each pin is defined as output.

PxDIR and PwDIR are cleared (0) on reset except when otherwise noted in Appendix A. Clearing these registers configures all the pins in port Px and Pw as input. Some specific reset values differ, as described in Appendix A on page 408.

Location: See Appendix A

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Px Port Direction**							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	**Pw Port Direction**							
Reset	0	0	0	0	0	0	0	0

Port Data Output Register (PxDOUT, PzDOUT and PwDOUT)

Writing to PxDOUT, PzDOUT or PwDOUT registers sets the values of the output pins in ports Px, Pz and Pw, respectively. Reading from one of these registers returns the last value written to the register.

Location: See Appendix A

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Px Port Output Data**							

Bit	7	6	5	4	3	2	1	0
Name	**Pz Port Output Data**							

Bit	7	6	5	4	3	2	1	0
Name	**Pw Port Output Data**							

Port Data Input Registers (PxDIN, PyDIN and PwDIN)

Reading from PxDIN, PyDIN or PwDIN returns the current value of the pins in port Px, Py and Pw, respectively.

Location: See Appendix A

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Px Port Input Data**							

Bit	7	6	5	4	3	2	1	0
Name	**Py Port Input Data**							

Bit	7	6	5	4	3	2	1	0
Name	**Pw Port Input Data**							
Reset								

Port Weak Pull-Up Registers (PxWPU, PyWPU)

These registers control the pull-up for the related pin, when it used either as GPIO or in its alternate function. The pull-up is enabled when the corresponding bit of PxWPU or PyWPU is set and the port buffer is in TRI-STATE. Otherwise, the pull-up is disabled (i.e., high impedance).

On reset, PxWPU or PyWPU is cleared (0), disabling all pull-ups.

Location: See Appendix A

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Px Port Weak Pull-Up Enable**							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	**Py Port Weak Pull-Up Enable**							
Reset	0	0	0	0	0	0	0	0

4.6 PS/2 INTERFACE

The PS/2 protocol is an industry-standard, PC-AT-compatible interface for keyboards. It uses a two-wire bidirectional TTL interface for data transmission. Several vendors also supply PS/2 mouse products and other pointing devices that employ the same type of interface.

The PC87591x provides four PS/2 data transfer channels. Each channel has two quasi-bidirectional signals that serve as direct interfaces to an external keyboard, mouse or any other PS/2-compatible pointing device. Since the four channels are identical, the connector ports are interchangeable.

4.6.1 Features

- Four PS/2 channels
- Enable/Disable for each of the four channels
- Automatic hardware shift mechanism
- Hardware support for PS/2 auxiliary device protocol
- Processor interrupts at the beginning and end of data transfer
- Optional software-based PS/2 implementation

4.6.2 General Description

In the previous generation of keyboard controllers, firmware executed the PS/2 device interface by toggling the interface signals. The PC87591x (and its predecessor, the PC87570) supports this bit toggling mode via either polling or interrupt-driven clock edge detection.

PS/2 devices' firmware is significantly simplified through the use of a hardware accelerator mechanism. The accelerator includes an 8-bit shift register, a state-machine and control logic that handle both the incoming and outgoing data. It reduces code overhead, performance requirements and reduces the overall interrupt latency from the core firmware. The hardware is designed to meet the PS/2 device interface as defined in *Keyboard and Auxiliary Device Controller (Types 1 and 2), August 1988.*

Section Naming conventions

- In this section, the term “channel” describes the interface to one of the PS/2 devices and its two associated signals (clock and data).
- The term “shift mechanism” refers to the hardware accelerator.
- The term “PS/2 interface” refers to the entire mechanism.

Interface Signals

The PS/2 interface includes eight external signals (PSCLK4-1 and PSDAT4-1) and six registers.

Module Block Diagram

A schematic description of the PS/2 interface appears in Figure 38. The interface to the three channels is symmetric and only channel 1 is detailed in the figure.

Embedded Controller Modules (Continued)

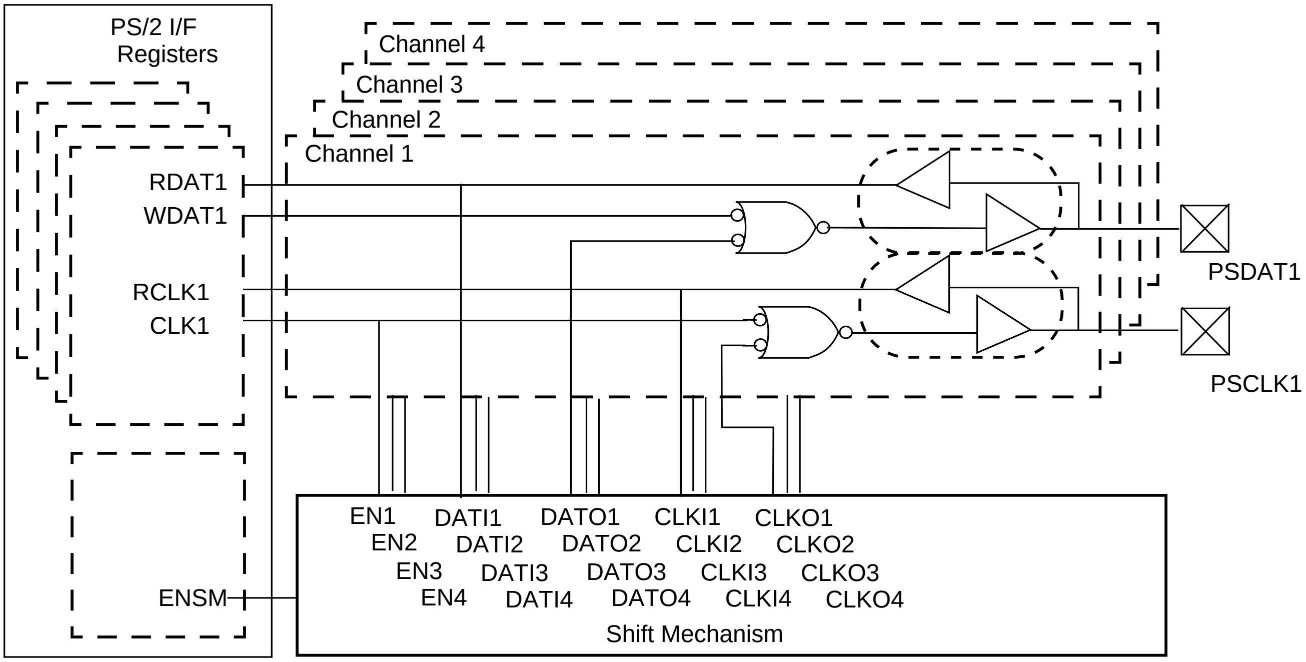


Figure 38. PS/2 Interface Functional Diagram

Quasi-Bidirectional Drivers

The quasi-bidirectional drivers have an open-drain output (Q2), an internal pull-up (Q3) and a low-impedance pull-up(Q1). Q2 pulls the signal low whenever the output buffer data is “0”. The weak pull-up (Q3) is active whenever the output buffer data is “1” and WPUEN in PSCON register is set (1). The low impedance pull-up is active whenever the PC87591x changes the output data buffer from “0” to “1”, thereby reducing the low-to-high transition time. The length of time that the low-impedance pull-up is active is determined by HDRV field in PSCON register. A schematic description of this output driver appears in Figure 39.

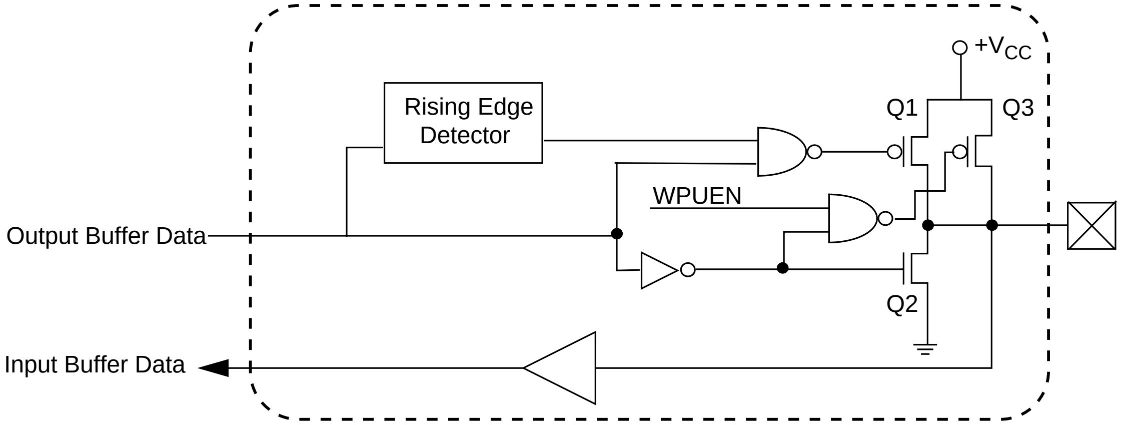


Figure 39. Quasi-Bidirectional Buffer

Interrupt Signals

The firmware can use an interrupt-driven scheme to implement the PS/2 device interface. When the shift mechanism is not in use, four interrupts are available (PSINT4-1), one for each channel. (PSINT4 is not connected to the ICU as a separate input, the MIWU PSCLK4 input should be used instead). When the shift mechanism is in use, only one interrupt signal is used (PSINT1). More details on the use of the interrupts are provided in Section 4.6.4 on page 125. Figure 40 illustrates the interrupt scheme with the associated enable bits.

Embedded Controller Modules (Continued)

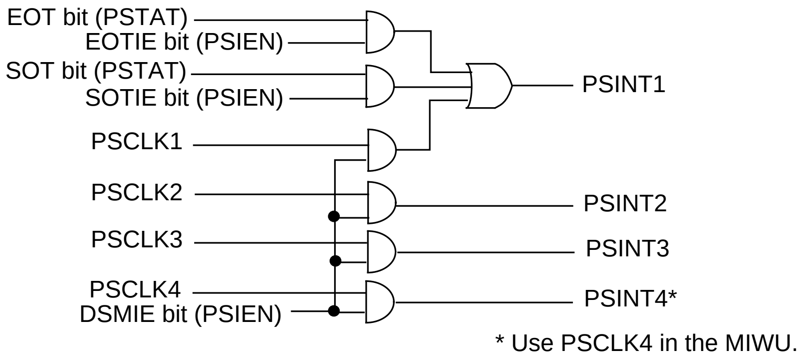


Figure 40. PS/2 Interface Interrupt Signals

Power Modes

The PS/2 interface is active only when the PC87591x is in Active mode. The shift mechanism should be disabled before entering Idle mode. In Idle mode, the state of output signals cannot be changed (i.e., the firmware cannot write to PSOSIG register, and the shift mechanism does not function).

When the PC87591x needs to wake up on a Start bit detection by the MIWU, the PS/2 channels that may serve as wake-up event sources must be enabled before entering Idle mode. To enable them, set to 1 their corresponding CLK bits in PSOSIG register.

The MIWU module can be used to identify a start bit in Idle mode and to return the PC87591x to Active mode. The MIWU receives PSCLK4-1 and PSDAT4-1 signals as inputs (see Table 17 on page 105). The MIWU should be programed to identify a falling edge on the clock or data lines of the enabled channels. In this configuration, a start bit causes the PC87591x to switch from Idle mode to Active mode. Once Active mode is reached, the firmware should cancel the transaction just started and then enable re-transmission of the information by the device.

PS/2 Interface Operation

The PS/2 interface has two basic operating methods: with the shift mechanism disabled and with the shift mechanism enabled. The following sections describe how to use the PS/2 interface with each of these operating methods.

4.6.3 Operating With the Shift Mechanism Disabled

The shift mechanism is disabled when EN bit in PSCON register is cleared (0). In this state, the PS/2 clock and data signals are controlled by the firmware, which performs the PS/2 protocol by manipulating the PS/2 clock and data signals.

Clock Signal Control

CLK4-1 bits in PSOSIG register control the value of the respective clock signals (PSCLK4-1). When one of these bits is cleared (0), the relevant pin is held low. When set (1), the open-drain output is open and the respective clock signal is either floating or held high by the pull-up. In this case, an external device can force the respective clock signal low.

When reading PSISIG register, bits RCLK4-1 indicate the current state of the corresponding clock signal.

Data Signal Control

WDAT4-1 bits in PSOSIG register control the value of the respective data signals (PSDAT4-1). When one of these bits is cleared (0), the relevant data signal is held low. When set (1), the open-drain output is open and the respective data signal is held high by the pull-up. In this case, an external device can force the respective data signal low.

When reading PSISIG register, bits RDAT4-1 indicate the current state of the corresponding data signal.

Interrupt Generation

When DSMIE bit in PSIEN register is set (1), the clock input signals are connected to the Interrupt Control Unit (ICU) for an interrupt driven PS/2 protocol. The four interrupts that are generated are PSINT4-1 for channels 4-1, respectively.

The ICU should be programed to detect a falling edge on each of the clock signals. Disabling a channel by writing 0 to the clock control signals (CLK4-1) may cause a falling edge on a clock signal. When such an interrupt is not desired, clear the clock control bit (0); then clear the respective pending bit in the ICU (or in the MIWU, for PSINT4). This should be done while interrupts are disabled. For more details about the ICU, see Section 4.3 on page 98.

4.6.4 Operating With the Shift Mechanism Enabled

The shift mechanism is designed to off load the bit level handling of the data transfer from the firmware to a hardware scheme; this improves system tolerance to interrupt latency. The mechanism includes a shift register and a state machine that controls the PS/2 protocol.

Figure 41 illustrates the shift mechanism PS/2 data transfer sequence. There are three basic modes: Disabled, Receive and Transmit. Different states in each mode define the progress of the data transfer. The rest of this section details the use of the shift mechanism for implementing a PS/2 data transfer.

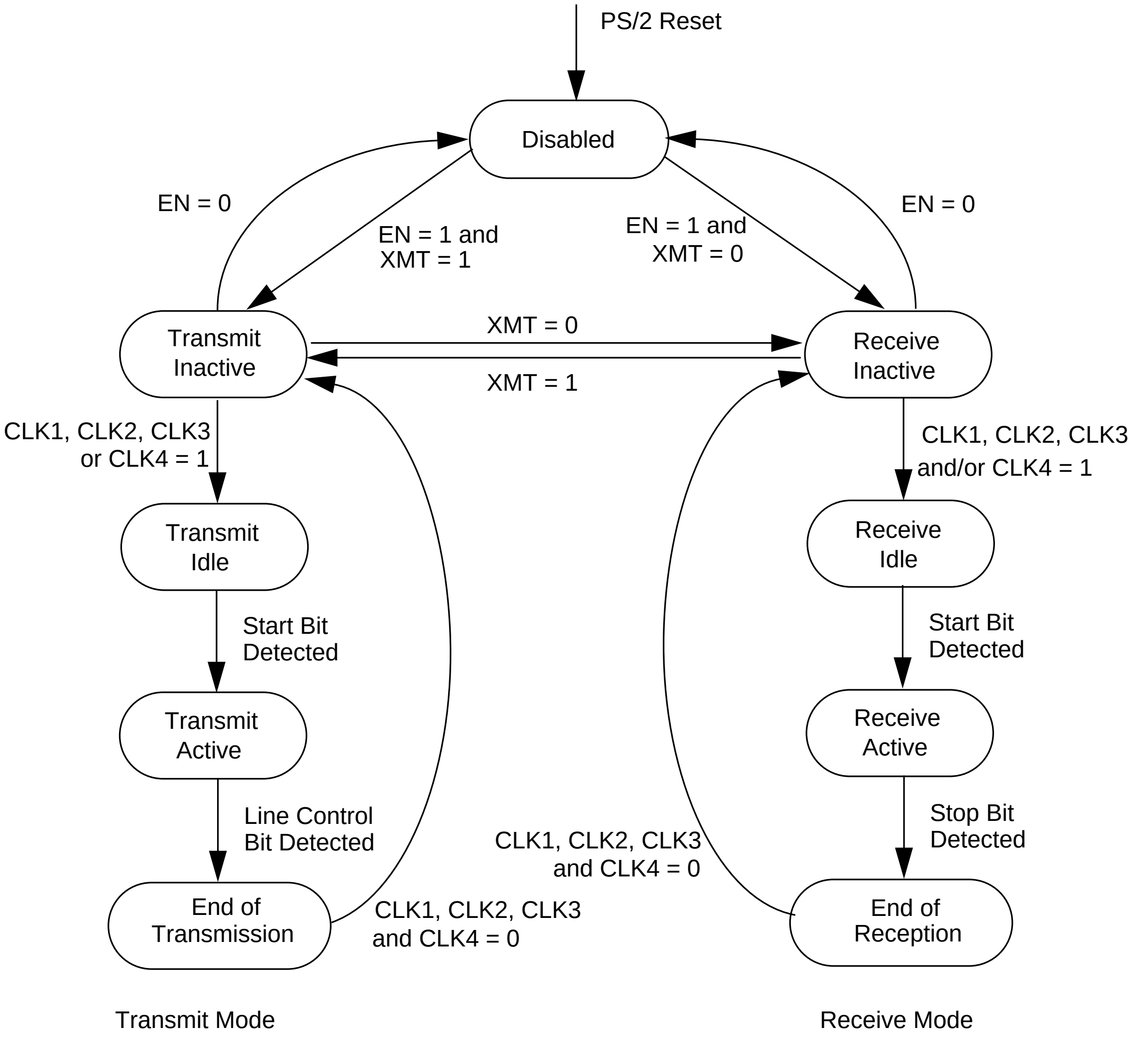


Figure 41. Shift Mechanism State Diagram

Reset the Shift Mechanism

Clearing either the shift mechanism enable bit (EN = 0 in PSCON register) or all the channels' clock bits (CLK4-1 = 0) resets the shift mechanism. In this state, PSTAT register is cleared (00_{16}), and the state of the PS/2 clock and data signals (PSCLK4-1 and PSDAT4-1) is set according to the value of their control bits (CLK4-1 and WDAT4-1, respectively).

When the shift mechanism is reset while in an unknown state or while in Transmit Idle state, the firmware should set (1) WDAT4-1 before the shift mechanism is reset.

Before disabling the shift mechanism, the software should clear (0) CLK4-1 to prevent glitches on the clock signals.

Enable the Shift Mechanism

To enable the shift mechanism, verify that PSOSIG register is set to 47_{16} and then set (1) EN bit in PSCON register. This puts the shift register state machine in Receive Inactive or Transmit Inactive state (XMT is 0 or 1, respectively, in PSCON register). In either of these states, the clock signals (PSCLK4-1) are low and the data signals PSDAT4-1 are either floating or pulled high.

Embedded Controller Modules (Continued)

Shift Status

The PSTAT register indicates the current status of the shift mechanism. The data transfer process may be in one of the following three states:

- Shifter Empty:
  The shift mechanism is in Receive Inactive, Receive Idle, Transmit Inactive or Transmit Idle state. The PSTAT is cleared because none of the enabled devices has sent a start bit.
- Start Bit Detected:
  The shift mechanism is in Receive Active or Transmit Active state. This indicates that a start bit was identified for at least one of the channels and the shift process has begun. SOT bit in PSTAT register indicates the detection of the start bit and ACH field in PSTAT register indicates the active channel (the channel on which the start bit was detected).
- End of Transaction:
  The shift mechanism is in End-of-Reception or End-of-Transmission state. This indicates that the last bit of the transfer sequence was detected (and the data can therefore be read from PSDAT register) or that the data transmission was completed (for receive and transmit, respectively). EOT bit in PSTAT register indicates transfer completion. If a parity error was detected in the received data, PERR bit in PSTAT register is set. If a stop bit was detected low instead of high, RFERR bit in PSTAT register is set.

Input Signal Debounce

The PC87591x performs a debounce operation on the clock input signal before determining its logical value. IDB field in PSCON register determines for how many clock cycles the input signal must be stable to define a change in its value.

Interrupt Generation

The PSINT1 is an interrupt signal generated by the shift mechanism to allow an interrupt driven interface with the firmware.

The ICU should be programed to detect high-level interrupts on the PSINT1 interrupt. See Section 4.3 on page 98 for details on the ICU. SOTIE and EOTIE bits in PSIEN register mask the interrupt signaling for SOT and EOT bits, respectively, in PSTAT register.

Receive Mode

Receive Inactive

When the shift mechanism is enabled and bit XMT=0 in PSCON register, the shift mechanism enters Receive mode in the Receive Inactive state. Receive Idle state is entered when one (or more) of the channels is enabled, by setting the channel enable bit (CLK4-1 for channels 4-1, respectively). In this state, the shift-mechanism sets the clock and data lines of the enabled channels high (1) and waits for a start bit.

Receive Idle

In the Receive Idle state, the PS/2 interface waits for input from any one of the enabled channels. The first of the enabled channels to send a start bit is selected for handling by the shift mechanism. The other two channels are disabled by forcing “0” on their clock lines.

Start Bit Detection

The start bit is identified by a falling edge on the clock signal while the data signal is low (0).

If the start bit is identified simultaneously in more than one channel, one channel is selected for receive, while the other channel’s transfer is aborted. The channel with the lower number is selected (i.e., channel 1 has priority over channels 2, 3 and 4, channel 2 has priority over channel 3 and 4 and channel 3 has priority over channel 4). The data transfer in the other channels is aborted before 10 data bits have been sent (by forcing the clock signal to 0), and the transmitting PS/2 device resends its data when its interface is enabled again by the firmware. This mechanism ensures that no incoming data is lost.

When the hardware sets (1) SOT bit and designates the selected channel in ACH field, this indicates receipt of the start bit in PSTAT register. In addition, if SOTIE is set in PSIEN register, an interrupt signal to the ICU is set high. The firmware may use this interrupt to start a time-out timer for the data transfer.

Receive Active

After identifying the start bit, the shift mechanism enters the “Receive-Active” state. In this state the clock signal of the selected device (PSCLK1, PSCLK2, PSCLK3 or PSCLK4) sets the data bit rate. On each falling edge of the clock, new data is sampled on the data signal of the active channel (i.e., PSDAT1 PSDAT2, PSDAT3 or PSDAT4).

Following the start bit, eight bits of data are received (clocks 2 through 9); a parity bit follows (10th clock) and then a stop bit (11th clock). The stop bit is indicated by a falling edge of the clock with the data signal high (1). If the 11th clock is identified with data low, the receive frame error bit (RFERR in PSTAT register) is set but the clock is treated as the stop bit.

After the parity is received, the shift mechanism checks the incoming data for parity errors. If there are eight data bits with a value of 1 and the parity bit is even, PERR bit in PSCON register is set, indicating a parity error.

Embedded Controller Modules (Continued)

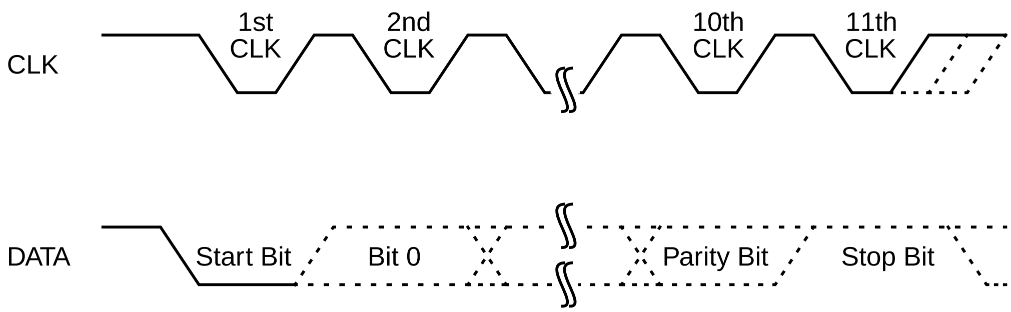


Figure 42. PS/2 Receive Data Byte Timing

End of Receive

When the stop-bit is detected, the shift mechanism enters the “End-Of-Reception” state. In this state, the shift mechanism:

- Disables all the clock signals by forcing them low
- Sets End-Of-Transaction status bit (EOT = 1 in PSTAT register)
- If EOTIE bit in PSTAT register is set, it asserts (1) the interrupt signal to the ICU.

The shift mechanism stays in this state until it is reset.

Figure 42 illustrates the receive byte sequence, as defined by the PS/2 standard.

Transmit Mode

Transmit Inactive

When the shift mechanism is enabled and XMT bit in PSCON register is set (1):

- The shift mechanism enters Transmit mode in Transmit Inactive state with all clock signals low and data signals high (PSOSIG = 47_{16}).
- The firmware writes the data to be transmitted to the PS/2 data register (PSDAT).
- The data line of the channel to be transmitted is forced low by the firmware clearing its data bit (WDAT4-1 for channels 4-1, respectively).

Transmit Idle

The Transmit Idle state can be entered by setting the channel enable bit (CLK4-1 for channel 4-1, respectively). This enables the channel to be used for transmission. In this state, the shift-mechanism sets the clock of the enabled channel high (1) while the data line of that channel is held low and waits for a start bit. When a PS/2 device senses the clock signal high with the data signal low, it identifies a transmit request from the PC87591x.

The three channels not in use are disabled by forcing “0” on their clock lines.

Start Bit Detection

The start bit is identified by a falling edge on the clock signal while the data signal is low (0).

When a start bit is detected, data transmission begins by outputting bit 0 (LSB) of the transmitted data and setting data bits WDAT4-1 in PSOSIG register. This allows bit 0 of the transmitted data to be output on the PS/2 data signal (PSDAT1, PSDAT2, PSDAT3 or PSDAT4, according to the active channel).

In addition, the hardware sets the SOT bit (to 1) and stores the active channel number in ACH field, indicating transmission of the start bit in PSTAT register. Note that if SOTIE bit in PSIEN register is set, an interrupt signal to the ICU is set high. The firmware can use this interrupt to start a time-out timer for the data transfer.

Transmit Active

After identifying the start bit, the shift mechanism enters the Transmit Active state. The clock signal of the selected device (PSCLK1, PSCLK2, PSCLK3 or PSCLK4) sets the data bit rate.

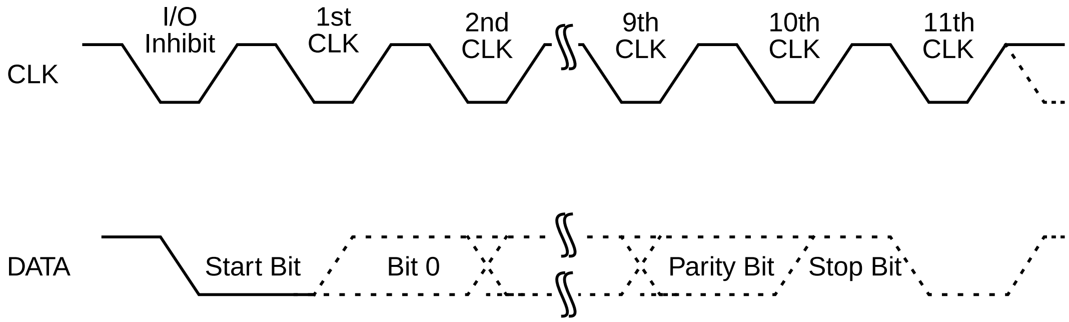


Figure 43. PS/2 Transmit Data Byte Timing

After each of the next seven falling edges of the clock line, one more data bit (bits 1 through 7) is driven on the data line of the active channel (either PSDAT1, PSDAT2, PSDAT3 or PSDAT4).

On the ninth falling edge of the clock, the parity bit is output. The parity bit is high (1) if the number of bits with a value of 1 in the transmitted data was even (i.e., odd parity).

The tenth falling edge causes a 1 to be output as a stop bit. The data signal remains high to allow the PS/2 device to send the line control bit.

The auxiliary device then completes the transfer by sending the line-control bit. The line-control bit, is identified by the data signal being low after the 11th falling edge of the clock.

End of Transmission

The End-Of-Transmission state is entered when the line-control bit is detected. In response, the shift mechanism holds all clock signals low, and if the internal pull-up is enabled, all data signals are pulled high by the internal pull-up.

The End-Of-Transaction flag (EOT in PSTAT register) is set to indicate that the transmit operation was completed; in addition, if EOTIE bit in PSIEN register is set, the interrupt signal to the ICU is set high.

The shift mechanism stays at this state until being reset.

Figure 43 illustrates the transmit byte sequence, as defined by the PS/2 standard.

Transfer Abort

At each stage of a receive or transmit operation, the transaction can be aborted by clearing all four channel enable bits (CLK4-1) in PS/2 Output Signal register (PSOSIG) to 0. This resets the shifter state machine and puts it in the Enabled Inactive state. If the shift mechanism is in Transmit Inactive or Transmit Idle state, WDAT4-1 bits should also be set.

4.6.5 PS/2 Interface Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Embedded Controller Modules (Continued)

PS/2 Register Map

Mnemonic	Register Name	Type
PSDAT	PS/2 Data Register	R/W
PSTAT	PS/2 Status Register	RO
PSCON	PS/2 Control Register	R/W
PSOSIG	PS/2 Output Signal Register	R/W
PSISIG	PS/2 Input Signal Register	RO
PSIEN	PS/2 Interrupt Enable Register	R/W

PS/2 Data Register (PSDAT)

The PSDAT register is a byte-wide read/write register. In Receive mode, PSDAT holds the data received in the last message from the PS/2 device. In Transmit mode, the data to be shifted out is written to this register. When the PS/2 i/f is reset, the contents of this register become invalid.

On reset, the PS/2 interface is set to Receive mode. In this mode, PSDAT should be read only when EOT bit in PSTAT register is set to 1.

Setting the transmit enable bit in PSCON register to 1 (XMT = 1 in PSCON register) puts the PS/2 interface in Transmit mode. PSDAT should be written only when in Transmit mode and when all four channel enable bits CLK4-1 in PSOSIG register are cleared (0).

Location: 00 FE80$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Data**							

Bit	Description
7-0	**Data.** Contains the data received in the last message (or that is transmitted in the following transmission). Bit 0 is the first bit to be shifted (LSB).

PS/2 Status Register (PSTAT)

The PSTAT register is a byte-wide read-only register. It contains the status information on the data transfer on the PS/2 ports. All non-reserved bits of PSTAT are cleared (0) on reset when CLK1, CLK2 and CLK3 in PSOSIG are cleared and when EN bit in PSCON register is cleared. Reading PSTAT does not clear any of its bits.

Location: 00 FE82$_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**RFERR**	**ACH**			**PERR**	**EOT**	**SOT**
Reset	x	0	0	0	0	0	0	0

Bit	Description
0	**SOT (Start of Transaction).** When set to 1, indicates that a start bit was detected. The ACH field (bits 5-3 of this register) indicates which of the channels it was detected on.
1	**EOT (End of Transaction).** When set to 1, Indicates that a PS/2 data transfer was completed, i.e., a stop bit was detected at Receive mode or a line control bit was detected at Transmit mode.
2	**PERR (Parity Error).** When set to 1, indicates that a parity error was detected in the last data transfer.

Embedded Controller Modules (Continued)

Bit	Description
5-3	**ACH (Active Channel).** Defines which of the PS/2 channels is currently active (i.e., a start bit was detected). In case more than one channel become active simultaneously, only the one with the highest priority (lowest number) is flagged. **Bits** **5 4 3 Description** 0 0 0: None of the channels is active 0 0 1: Channel 1 0 1 0: Channel 2 1 0 0: Channel 3 1 0 1: Channel 4
6	**RFERR (Receive Frame Error).** When set to 1, indicates that the stop bit in a received frame was detected low instead of high.
7	**Reserved.**

PS/2 Control Register (PSCON)

The PSCON register is an 8-bit read/write register. It controls the operation of the PS/2 interface by enabling it and controlling the data transfer direction. On reset, PSCON is set to 00_{16}.

Location: 00 $FE84_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**WPUEN**	**IDB**			**HDRV**		**XMT**	**EN**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**EN (Shift Mechanism Enable).** 0: The hardware shift mechanism is disabled and the software controls and monitors the PS/2 signals using PSOSIG and PSISIG registers. 1: The hardware shift mechanism is enabled. The enabled channels are controlled by PSOSIG, and Transmit/Receive mode is controlled by the XMT bit.
1	**XMT (Transmit Enable).** 0: Receive mode. 1: Causes the PS/2 interface to enter Transmit mode.
3-2	**HDRV (High Drive).** Defines the quasi-bidirectional buffers’ behavior on transition from low to high. HDRV defines the period of time for which the output is pulled high with a low-impedance drive (when the PC87591x changes the output level from low to high). This period is a function of the PC87591x clock as follows: **Bits** **3 2 Description** 0 0: Disabled 0 1: Low-impedance drive for one clock cycle 1 0: Low-impedance drive for two clock cycles 1 1: Low-impedance drive for three clock cycles

Embedded Controller Modules (Continued)

Bit	Description
6-4	**IDB (Input Debounce).** Defines the number of PC87591x clock cycles during which the clock input is expected to be stable before the shift mechanism identifies its new value. This protects the shift mechanism from false edge detections. The number of PC87591x clock cycles for which the input should be stable before an edge is detected is as follows: **Bits** **6 5 4 Description** 0 0 0: One cycle 0 0 1: Two cycles 0 1 0: Four cycles 0 1 1: Eight cycles 1 0 0: 16 cycles 1 0 1: 32 cycles
7	**WPUEN (Weak Pull-Up Enable).** 0: The pull-up is disabled. In this state, the system must ensure that PS/2 interface signals are not floating, to enable proper PS/2 operation. 1: Enables the internal pull-up of the output buffer. The pull-up remains active as long as the buffer does not drive the signal to low level.

PS/2 Output Signal Register (PSOSIG)

The PSOSIG register is a byte-wide, read/write register. It allows setting the value of the PS/2 port signals. When the shift mechanism is enabled, the clock control bits in this register define the active channel(s). On reset, this register is set to 47_{16}.

Location: 00 $FE86_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**CLK4**	**WDAT4**	**CLK3**	**CLK2**	**CLK1**	**WDAT3**	**WDAT2**	**WDAT1**
Reset	0	1	0	0	0	1	1	1

Bit	Description
0	**WDAT1 (Write Data Signal Channel 1).** Controls the data output to channel 1 data signal (PSDAT1). Use of this bit depends on whether or not the shift mechanism is enabled. • When the shift mechanism is disabled (EN bit in PSCON register is set to 0), the data in WDAT1 is output to PSDAT1 signal. – If WDAT1 is cleared (0), the output buffer data is 0 (i.e., PSDAT1 is forced low). – If WDAT1 is set (1), the output buffer data is 1 (i.e., PSDAT1 is pulled high by the internal pull-up and may be pulled low by an external device). • When the shift mechanism is enabled (EN=1), WDAT1 should be set to 1, except when the shift mechanism is in Transmit mode. In this case, when in transmit-inactive and it is intended to transmit data to channel 1, the firmware should clear WDAT1 bit to force the transmit signaling (low) to the PS/2 device. **Note:** WDAT1 is set by the hardware after the PC87591x detected a start bit (i.e., on entering Transmit Active state). If a transmission is aborted before Transmit Active state, WDAT1 should be set (1) prior to disabling the channel.
1	**WDAT2 (Write Data Signal Channel 2).** Controls the data output to channel 2 data signal (PSDAT2). For more information, see the description of bit 1 (above).
2	**WDAT3 (Write Data Signal Channel 3).** Controls the data output to channel 3 data signal (PSDAT3). For more information, see the description of bit 1 (above).

Embedded Controller Modules (Continued)

Bit	Description
3	**CLK1 (Enable Channel 1)** 0: Forces the PSCLK1 pin low (0) and disables channel 0 of the shift mechanism. 1: Depends on whether or not the shift mechanism is enabled. • When the shift mechanism is enabled (EN bit in PSCON register is set to 1), channel 1 of the PS/2 ports is enabled. • When the shift mechanism is disabled (EN bit in PSCON register is set to 0), the clock line output buffer data is 1 (i.e., the signal is pulled high by the pull-up, if enabled, and may be pulled low by an external device).
4	**CLK2 (Enable Channel 2).** Same as bit 3 of this register (described above) but for channel 2.
5	**CLK3 (Enable Channel 3).** Same as bit 3 of this register (described above) but for channel 3.
6	**WDAT4 (Write Data Signal Channel 4).** Controls the data output to channel 4 data signal (PSDAT4). For more information, see the description of bit 1 (above).
7	**CLK4 (Enable Channel 4).** Same as bit 3 of this register (described above) but for channel 4.

Note: When CLK1, CLK2, CLK3 and CLK4 are all 0, this is interpreted as a shift mechanism reset. In this case, the PSTAT register and the shift state machine are reset to their initial state.

PS/2 Input Signal Register (PSISIG)

The PSISIG register is an 8-bit read only register. It provides the current value of the PS/2 port signals.

Location: 00 $FE88_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**RCLK4**	**RDAT4**	**RCLK3**	**RCLK2**	**RCLK1**	**RDAT3**	**RDAT2**	**RDAT1**

Bit	Description
0	**RDAT1 (Read Data Signal Channel 1).** The current value of the channel 1 data signal (PSDAT1).
1	**RDAT2 (Read Data Signal Channel 2).** The current value of the channel 2 data signal (PSDAT2).
2	**RDAT3 (Read Data Signal Channel 3).** The current value of the channel 3 data signal (PSDAT3).
3	**RCLK1 (Read Clock Signal Channel 1).** When read, returns the current value of the channel 1 clock signal (PSCLK1).
4	**RCLK2 (Read Clock Signal Channel 2).** When read, returns the current value of the channel 2 clock signal (PSCLK2).
5	**RCLK3 (Read Clock Signal Channel 3).** When read, returns the current value of the channel 3 clock signal (PSCLK3).
6	**RDAT4 (Read Data Signal Channel 4).** The current value of the channel 4 data signal (PSDAT4).
7	**RCLK4 (Read Clock Signal Channel 4).** When read, returns the current value of the channel 4 clock signal (PSCLK4).

PS/2 Interrupt Enable Register (PSIEN)

The PSIEN register is an 8-bit read/write register. It enables/disables the various interrupts generated by the PS/2 module. Bits in PSIEN register may be cleared to 0 only when interrupts are disabled (i.e., in the core, I or E bits in PSR register are 0) or when the corresponding interrupts in the ICU are masked. Bits in PSIEN register may be set to 1 at any time. On reset, non-reserved bits of PSIEN are cleared.

Location: 00 FE8A$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**DSMIE**	**EOTIE**	**SOTIE**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**SOTIE (Start of Transaction Interrupt Enable).** Used for enabling the interrupt generation on a transaction start detection. 0: SOT bit in PSTAT register does not affect the interrupt signal. 1: The interrupt signal (PSINT1) to the ICU is active (1) whenever SOT bit in PSTAT register is set. **Note:** Once set, SOT is not cleared until the shift mechanism is reset. Therefore SOTIE should be cleared on the first occurrence of an SOT interrupt. SOTIE should be set (1) when the PS/2 module is programed to handle the impending transfer.
1	**EOTIE (End of Transaction Interrupt Enable).** Used for enabling the interrupt generation on an End of Transaction detection. 0: EOT bit in PSTAT register does not affect the interrupt signal. 1: The interrupt signal (PSINT1) to the ICU is active (1) whenever EOT bit in PSTAT register is set. **Note:** Once set, EOT is not cleared until the shift mechanism is reset. Therefore EOTIE should be cleared on the first occurrence of an EOT interrupt. EOTIE should be set (1) when the PS/2 module is programed to handle the impending transfer.
2	**Disabled Shift Mechanism Interrupt Enable (DSMIE).** Used for enabling the interrupt generation when the shift mechanism is disabled. 0: The four interrupt signals are low. Note that PSINT1 may be activated (1) by other interrupt sources of the module. 1: The clock input signals are connected to the Interrupt Control Unit (ICU), to allow implementing an interrupt driven PS/2 protocol. The four interrupts generated are PSINT1, PSINT2, PSINT3 and PSINT4, for channels 1, 2, 3 and 4, respectively. Note that PSINT4 is connected to the MIWU and not directly to the ICU. **Note:** When the shift mechanism is disabled, no debounce is applied to the PSCLK inputs before producing the interrupt signals, except for local synchronization.
7-3	**Reserved.**

4.7 MULTI-FUNCTION 16-BIT TIMER (MFT16)

The PC87591x includes two Multi-Function Timer (MFT) modules. The registers of each module are prefixed with Tn and the signals are suffixed with an n (where n is the module's number, i.e., 1 or 2). Each MFT16 module consists of two functional units designed to satisfy a wide range of application requirements. The units include:

- Clock Source Unit that contains a pre-scaler with one clock source selector for each counter.
- Main timer/counter and action unit that contains two counters, two reload registers for PWM, Input Capture or Counter modes.
- Mode selector/control unit that defines the function of the I/O pins and the interrupts.

Figure 44 shows the contents of an MFT16 and the top-level interaction. The rest of the section describes an MFT16 module.

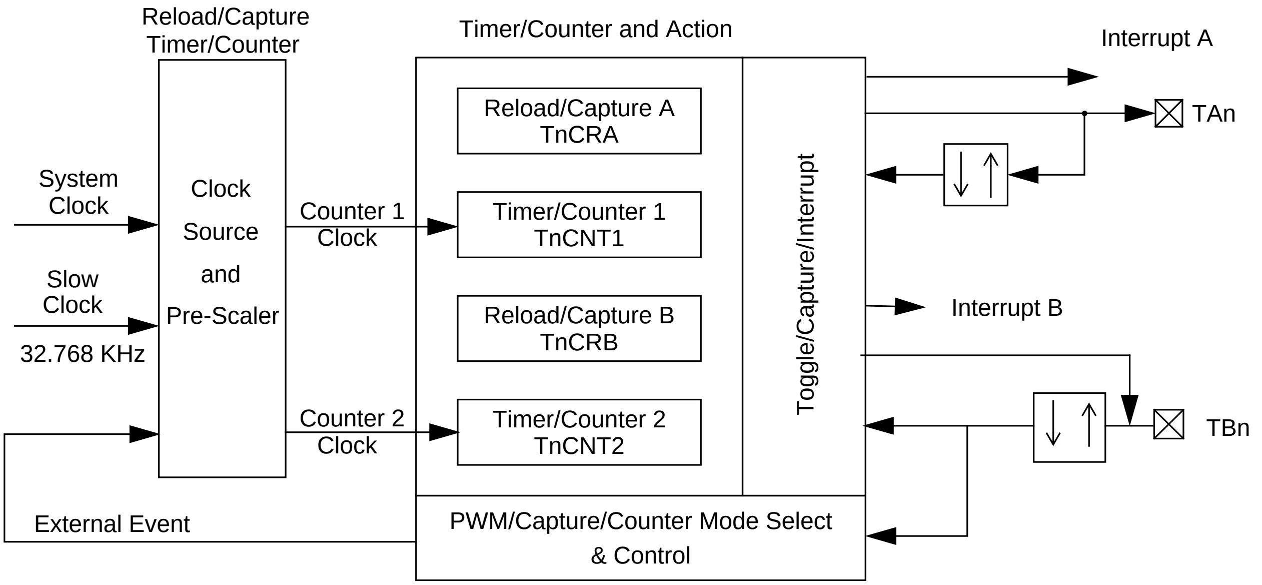


Figure 44. MFT16 Functional Diagram

4.7.1 Features

- Two 16-bit programmable timers/counters
- Two 16-bit reload/capture registers. These registers are used either as reload registers or capture registers, depending on the mode of operation.
- A 5-bit fully programmable clock pre-scaler
- Clock source selectors for each counter. These enable each counter to operate in:
  - — Pulse accumulate mode
  - — External event mode
  - — Prescaled system clock mode
  - — Slow speed clock input mode (where applicable)
- Two I/O pins (TAn and TBn), with programmable edge detection; these operate as:
  - — Capture inputs
  - — Capture and preset inputs
  - — External event (clock) inputs
  - — PWM outputs
- Two interrupts, one for each counter, that can be generated/ triggered by:
  - — Timer underflow
  - — Timer reload
  - — Input capture
- Four pending bits, which can be polled by software, are associated with the two interrupts.

4.7.2 Clock Source Unit

The clock source unit, as shown in Figure 45, contains two clock selectors for each counter and a 5-bit clock pre-scaler.

Pre-Scaler

The 5-bit clock pre-scaler consists of a pre-scaler register and a 5-bit counter, allowing the timer to run with a prescaled clock. The system clock is divided by the value contained in TnPRSC+1. The minimum counter clock frequency is thus the system clock divided by 32, and the maximum counter clock frequency is equal to the system clock. The pre-scaler register, TnPRSC, can be read or written by software at any time. The pre-scaler counter is a 5-bit down counter which can not be read or written by software. The 5-bit counter and the pre-scaler register TnPRSC are cleared on reset

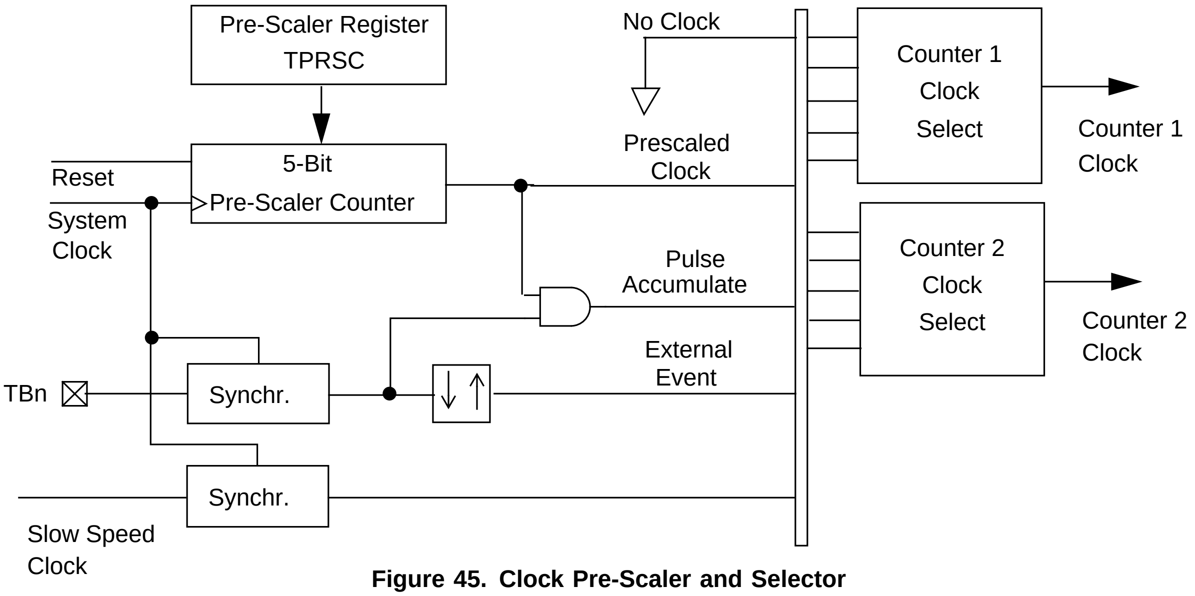


Figure 45. Clock Pre-Scaler and Selector

External Event Clock

The TBn I/O pin can be selected as an external event input clock source for either of the two 16-bit counters. The polarity of the input signal is programmable to trigger a count if either a rising or a falling edge is detected on TBn. The minimum pulse width of the external signal is one system clock cycle; thus the maximum frequency with which the counter can run in this mode is limited to half the system clock frequency.

Note: An External Event clock is not available in Dual Channel Capture mode because this mode requires TBn as an input.

Pulse Accumulate Mode

In Pulse Accumulate mode, the counter can also be clocked while an external signal on TBn is either high or low. In this configuration, the output of the pre-scaler is gated with an external signal applied on TBn input. This mode can be used to obtain a cumulative count of pre-scaler output clock pulses, as shown in Figure 46.

Note: Pulse Accumulate mode is not available in Dual Channel Capture mode because this mode requires TBn as an input.

Slow-Speed Clock

A slow-speed clock of 32.768 KHz can be used as a clock source for the two 16-bit counters. The MFT16 synchronizes the slow-speed clock with the system clock. Therefore, the maximum input frequency of the slow-speed clock is the system clock rate divided by four.

Some power save modes stop the system clock completely. When this occurs, the timer stops counting the slow-speed clock until the system clock resumes. While operating in a power-save mode that uses a slow system clock, the slow-speed clock source for the timer can only be used if it is at least four times faster than the slow- speed clock for the counter.

Embedded Controller Modules (Continued)

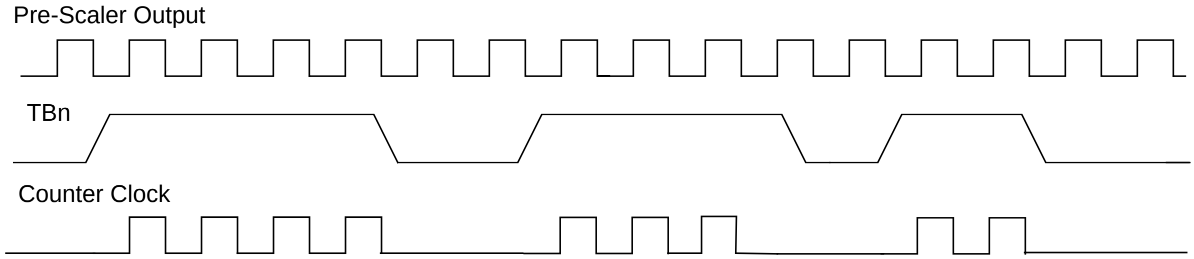


Figure 46. Pulse Accumulate Mode

Counter Clock Source Select

The clock source unit contains two clock source selectors that allow the clock source to be selected independently for each of the two 16-bit counters from one of the following sources:

- No clock, in which case the counter is stopped
- Prescaled system clock
- External Event count based on TBn
- Pulse Accumulate mode based on TBn
- Slow Speed Clock i.e., 32.768 KHz

4.7.3 Timer/Counter and Action Unit

The timer/counter and action unit consists of two 16-bit counters, TnCNT1 and TnCNT2, in addition to two 16-bit reload/capture registers, TnCRA and TnCRB. The timers are down counters capable of triggering events on underflow detection (count from 0000_{16} to $FFFF_{16}$). In addition, it contains the mode control logic which allows the timer to operate in any of four operation modes described below.

Different interrupts can be triggered on certain conditions, and the functionality of the I/O pins changes depending on the mode of operation. Therefore the interrupt control and the I/O control are an integral part of the timer/counter unit.

Operation Modes

The MFT16 can be configured to operate in any one of four modes, as summarized in Table 18 and described in this section.

Table 18. Operation Modes

Mode	Description	Timer/Counter 1 (TnCNT1)	Reload/Capture A (TnCRA)	Reload/Capture B (TnCRB)	Timer/Counter 2 (TnCNT2)
1	PWM and system timer or external event counter	Counter for PWM	Auto Reload A = PWM time 1	Auto reload = PWM time 2	System Timer or external event counter
2	Dual input capture and system timer	Capture A and B time base	Capture counter 1 value on TAn event	Capture counter 1 value on TBn event	System Timer
3	Dual independent timer	Time base for first timer	Reload register for timer/counter 1	Reload register for timer/counter 2	Time base for second timer
4	Input capture and timer	Time base for first timer	Reload register for timer/counter 1	Capture counter 1 value on TBn event	Capture B time base

Mode 1, PWM and Counter

PWM can be used to generate precise pulses of known width and duty cycle on the TAn pin. The timer is clocked by the instruction clock. An underflow causes the timer register to be reloaded alternately from the TnCRA and TnCRB registers and optionally causes TAn output to toggle. Thus, the values stored in TnCRA and TnCRB registers control the high and low time of the signal produced on TAn. In PWM mode, timer/counter 2 can be used either as a simple system timer or as an external event counter. The counter can be loaded by software with a specific value; the counter can then generate an interrupt after the pre-programed number of external events have been received on TBn input.

Embedded Controller Modules (Continued)

Figure 47 shows a block diagram of the timer operating in mode 1. In PWM mode, counter 1, TnCNT1, functions as the time base for the PWM timer. Counter 1 counts down at the clock rate selected via the counter 1 clock selector. When an underflow occurs, the timer register is reloaded alternately from the TnCRA and TnCRB registers, and counting proceeds downward from the loaded value. On reset, and every time this mode is entered, the first reload in this mode is from the TnCRA register. Once enabled, the counter starts counting down from the value currently loaded to it. At the first underflow, the timer is loaded from TnCRA; on the second underflow, it is loaded from TnCRB; on the third underflow, it is loaded from TnCRA, and so on. Note that every time the counter is stopped through the selection of “No-Clock” in the counter 1 clock selector (TnCKC), it obtains its first reload value after it has been re-started from TnCRA register.

The timer can be configured to toggle TAn output bit on underflow. This results in the generation of a clock signal on TAn, with the width and duty cycle controlled by the values stored in TnCRA and TnCRB registers. This PWM clock is processor-independent because once the timer is set up, no more interaction is required by software (and therefore the CPU) to generate a continuous PWM signal. Software can select the initial value of the PWM output signal as either high or low. See “Timer I/O Functions” on page 141 for additional details. The timer can be configured to generate separate interrupts on reload from TnCRA and TnCRB. The interrupts can be enabled or disabled under software control. The TAnPND or TnBPND flags, respectively, which are set by the hardware on occurrence of a timer reload, indicate which interrupt occurred. See Section 4.7.4 on page 140 for detailed information.

In this mode of operation, timer/counter 2 can be used as a simple system timer, an external event counter or a pulse accumulate counter. Counter TnCNT2 counts down with the clock selected via the counter 2 clock selector, and TnCNT2 can be configured to generate an interrupt on underflow if the interrupt is enabled by TnDIEN bit. See Section 4.7.4 on page 140 for detailed information.

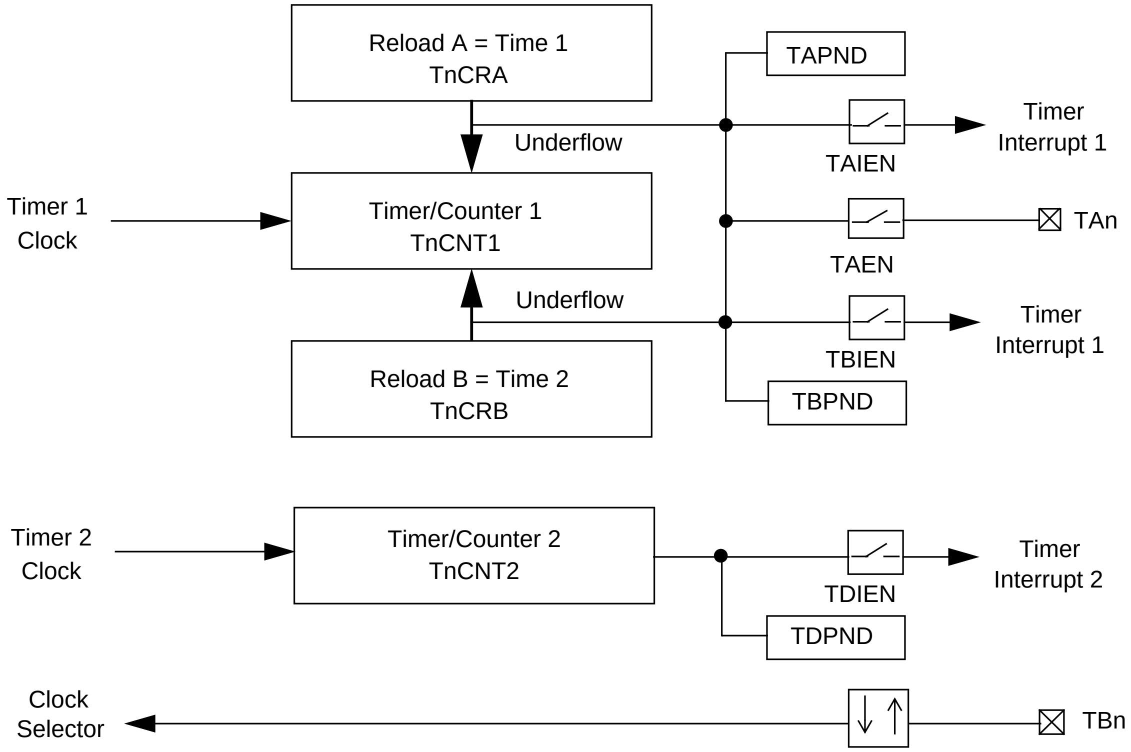


Figure 47. Mode 1, PWM and Counter

Mode 2, Dual Input Capture

Dual Input Capture mode can be used to precisely measure the frequency of an external clock that is slower than the selected clock source frequency or to measure the elapsed time between external events. A transition received on the TAn or TBn pin causes a transfer of timer/counter 1 contents to TnCRA or TnCRB register, respectively. In this mode, timer/counter 2 can be utilized as a system timer that is pre-loaded by software and generates an interrupt on underflow.

Figure 48 shows a block diagram of the timer operating in mode 2. In this mode of operation, the timebase of the capture timer is formed by counter 1, which counts down with the clock selected via the counter 1 clock selector. In Dual Input Capture mode, TAn and TBn pins function as capture inputs. A transition received on TAn pin causes a transfer of the timer contents to TnCRA register. Similarly, a transition received on the TBn pin causes a transfer of the timer contents to TnCRB register. TAn and TBn inputs can be configured to perform a counter preset to $FFFF_{16}$ on reception of a valid capture event.

Embedded Controller Modules (Continued)

In this case, the current value of the counter is transferred to the corresponding capture register; following this, the counter is preset to $FFFF_{16}$. Using this approach enables an external signal's on-time, off-time or period to be directly determined, while reducing CPU overhead.

The pulse width of the input signal on TAn and TBn must be equal to or greater than one system clock cycle (see Section 7.6.10 on page 399 for additional details). The values captured in TnCRA register at different times reflect the elapsed time between transitions on the TAn pin. The same is true for TnCRB register and the TBn pin. Each input pin can be configured to sense either rising or falling edge transitions.

The timer can be configured to generate interrupts on reception of a transition on either TAn or TBn. The interrupts can be enabled or disabled separately for TAn or TBn by TAnIEN and TnBIEN bits. An underflow of TnCNT1 can also generate an interrupt if the interrupt was enabled by TnCIEN bit. All three interrupts have individual pending flags. See Section 4.7.4 on page 140 for detailed information.

Timer/counter 2 can be used as a "simple" system timer in this mode of operation. The TnCNT2 counter counts down with the clock selected via the counter 2 clock selector, and TnCNT2 can be configured to generate an interrupt on underflow if the interrupt was enabled by TnDIEN bit. See Section 4.7.4 on page 140 for detailed information.

Note that TnCNT1 cannot operate in the "Pulse Accumulate" or "External Event Counter" modes of operation since TBn input is used as a capture input. Selecting either "Pulse Accumulate" mode or "External Event Counter" mode for TnCNT1 causes TnCNT1 to stop. However, all available clock source modes may be selected for TnCNT2. Thus it is possible to use TnCNT2 to determine the number of capture events on TBn or the elapsed time between capture events on TBn.

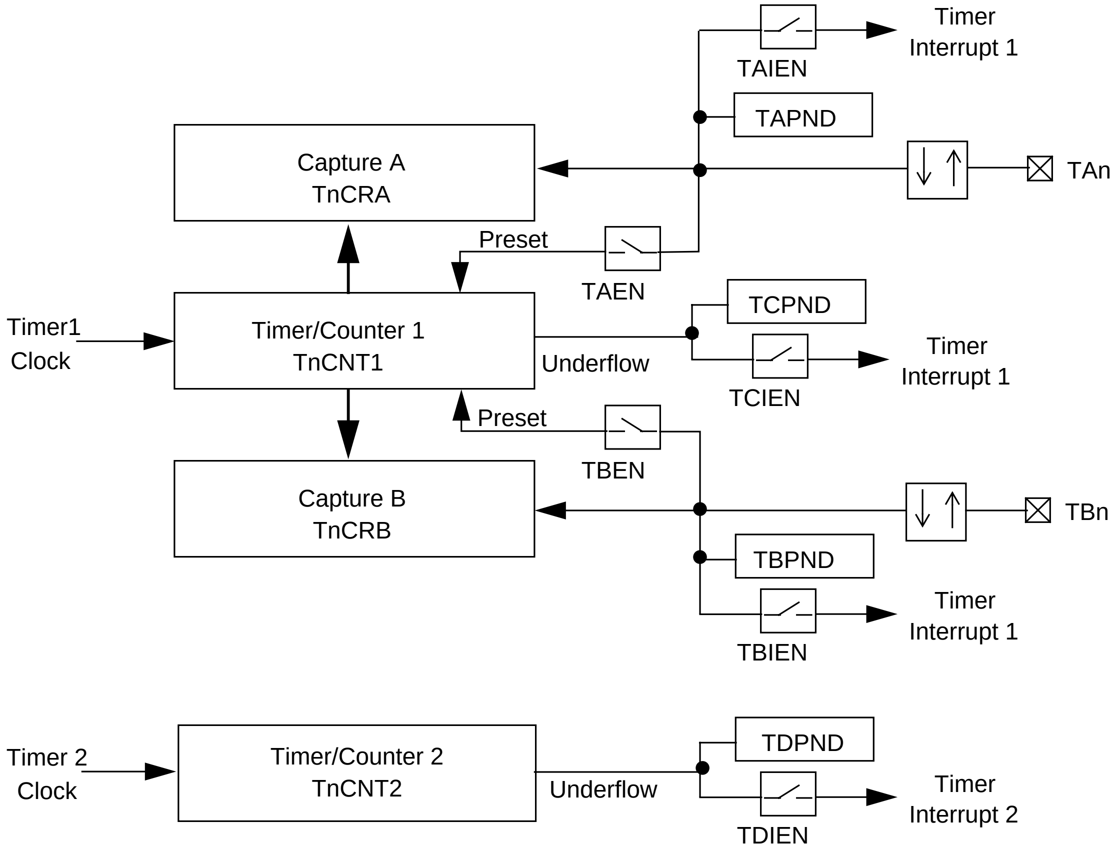


Figure 48. Mode 2, Dual Input Capture

Mode 3, Dual Independent Timer

Dual Independent Timer mode can be used for a wide variety of system tasks such as the generation of periodic system interrupts, based either on the prescaled clock or external events on TBn. The timer can also toggle TAn pin on underflow, allowing the simple generation of a processor-independent 50% duty cycle PWM signal on TAn. In this mode, TnCNT1 counts down and reloads from TnCRA on underflow while TnCNT2 is reloaded from TnCRB on underflow.

In this mode, the timer is configured to operate as a dual independent system timer or dual external event counter. In addition, timer/counter 1 can generate a 50% duty cycle PWM signal on the TAn pin. The TBn pin can be used as an external event input or pulse accumulate input and forms the clock source to either counter 1 or counter 2, as described above. Both counters can also be operated using the prescaled system clock. Figure 49 shows a block diagram of the timer in mode 3.

Embedded Controller Modules (Continued)

Timer/counter 1 (TnCNT1) counts down at the rate of the selected clock (see “Counter Clock Source Select” on page 136 for additional details). On underflow, TnCNT1 is reloaded from TnCRA register and counting proceeds. If enabled, the TAn pin toggles on underflow of TnCNT1. Software can select the initial value of TAn output as either high or low (see “Timer I/O Functions” on page 141 for additional details). In addition, the TnAPND interrupt pending flag is set, and a timer interrupt 1 is generated if TnAIEN bit is set to 1 (see Section 4.7.4 on page 140 for detailed information). Since TAn toggles on every underflow, a 50% duty cycle PWM signal can be generated on TAn without requiring interaction by the core.

Timer/counter 2 (TnCNT2) counts down at the rate of the selected clock (see “Counter Clock Source Select” on page 136 additional details). On every underflow of TnCNT2, the value contained in TnCRB register is loaded into TnCNT2, and counting proceeds downwards from that value. In addition, the TnDPND interrupt pending flag is set, and a timer interrupt 2 is generated if TnDIEN bit is set to 1. See Section 4.7.4 on page 140 for detailed information.

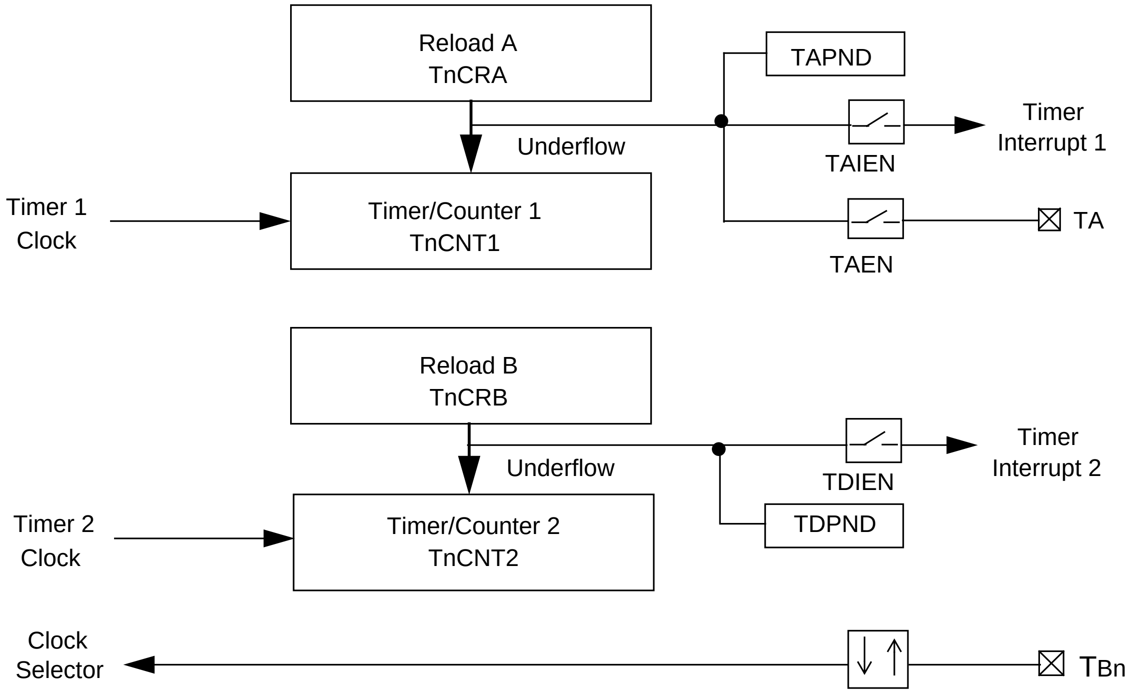


Figure 49. Mode 3, Dual Independent Timer

Mode 4, Input Capture and Timer

It is also possible to operate in a mode that offers a combination of a single timer, with automatic reload, and a single capture timer. In this mode, TnCNT1 operates as a PWM-timer that is reloaded from TnCRA on underflow while TnCNT2 forms the time base of the capture timer. The value on TnCNT2 is transferred to TnCRB when a valid event on TBn is detected. It is possible to toggle TA on every underflow of TnCNT1 and thus generate a 50% duty cycle PWM signal on TAn.

This mode is a combination of modes 3 and 2. It allows timer/counter 2 to operate as a single-input capture timer concurrently with timer/counter 1. (Timer/counter 2 can be used as a system timer as described in mode 3.) Figure 50 shows a block diagram of the timer in mode 4.

TnCNT1 starts counting down once a clock has been enabled. On underflow, TnCNT1 is reloaded from TnCRA register, and counting proceeds downwards from that value. If enabled, the TAn pin toggles on every underflow of TnCNT1. Software can select the initial value of TAn output signal as either high or low (see Section 4.7.5 on page 141 for additional details). In addition, the TnAPND interrupt pending flag is set, and a timer interrupt 1 is generated if TnAIEN bit is set to 1 (see Section 4.7.4 on page 140 for detailed information). Since TAn toggles on every underflow, a 50% duty cycle PWM signal can be generated on TAn without requiring any interaction of software (and therefore the core).

TnCNT2 starts counting down once a clock has been enabled. When a transition is received on TBn, the value contained in TnCNT2 is transferred to TnCRB, and the interrupt pending flag, TnBPND, is set. A timer interrupt 2 is generated if it is enabled. A preset of the counter to $FFFF_{16}$ on detection of a transition on TBn can be enabled. In this case, the current value of TnCNT2 is transferred to TnCRB, followed by a preset of the counter to $FFFF_{16}$. TnCNT2 starts counting downwards from $FFFF_{16}$ until the next transition is received on TBn, which causes the procedure of capture and preset to be repeated. Underflow of TnCNT2 sets the TnDPND interrupt pending flag and can also generate a timer interrupt 2 if the interrupt was enabled (see Section 4.7.4 on page 140 for detailed information.). The input signal on TBn must have a pulse width equal to or greater than one system clock cycle (see Section 7.6.10 on page 399 for additional details). TBn can be configured to sense either rising or negative edge transitions.

Embedded Controller Modules (Continued)

Note that TnCNT2 can not operate in the Pulse Accumulate or External Event Counter modes since TBn input is used as a capture input. Selecting either Pulse Accumulate mode or External Event Counter mode for TnCNT2 causes TnCNT2 to stop.

Note, however, that all available clock source modes may be selected for TnCNT1. Thus it is possible to use TnCNT1 to determine the number of capture events on TBn or the elapsed time between capture events on TBn.

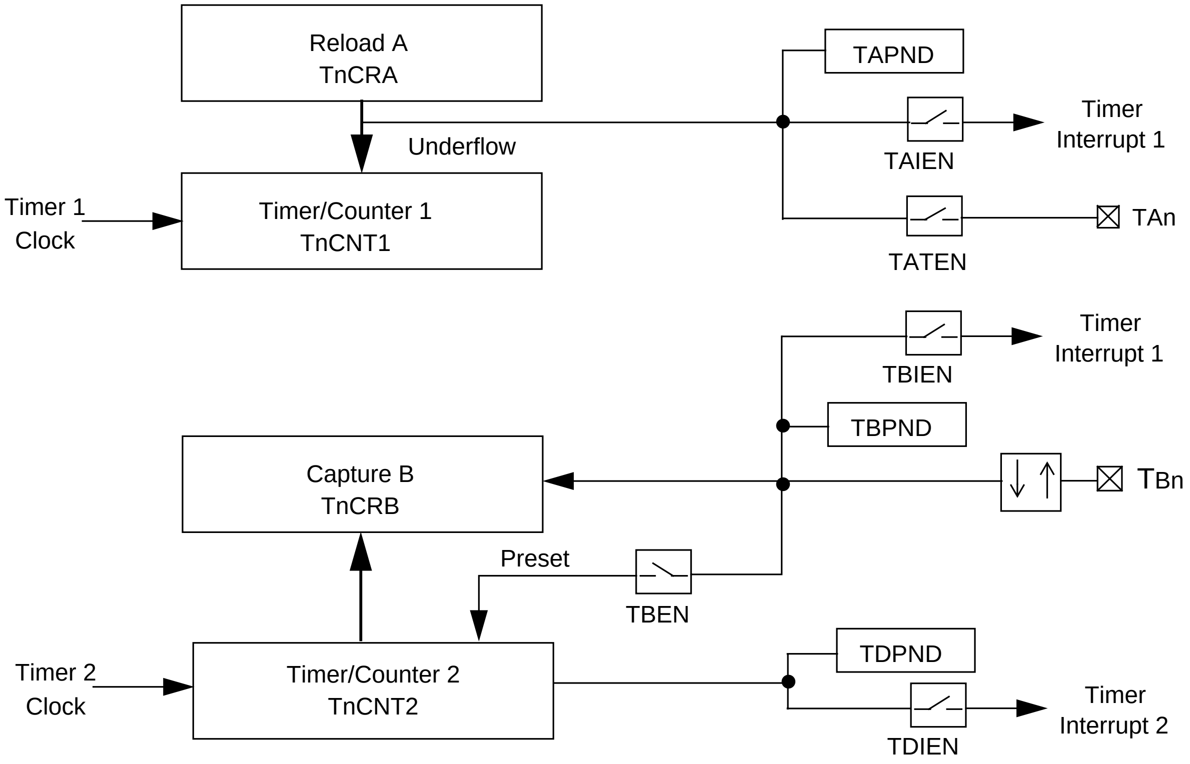


Figure 50. Mode 4, Input Capture and Timer

4.7.4 Timer Interrupts

The MFT16 contains a total of four interrupt sources that are mapped to two different system interrupts. All sources have a pending flag associated with them and can be enabled or disabled under software control. The pending flags are TnXPND, where n is the module and X is a letter from A to D. An interrupt enable flag, TnXIEN, is associated with each interrupt pending flag. Interrupt sources A, B or C can generate a timer interrupt 1; interrupt source D can generate a timer interrupt 2. Note that not all interrupt sources are available in all modes. Table 19 shows which events can trigger an interrupt in each mode of operation:

Table 19. MFT16 Interrupts

System Interrupt	Interrupt Pending Flag	Mode 1	Mode 2	Mode 3	Mode 4
		PWM and Counter	Dual Input Capture	Dual Independent Timer	Input Capture and Timer
Timer Int. 1	TnAPND	TnCNT1 reload from TnCRA	Input capture on TAn transition	TnCNT1 reload from TnCRA	TnCNT1 reload from TnCRA
	TnBPND	TnCNT1 reload from TnCRB	Input Capture on TBn transition	N/A	Input Capture on TBn transition
	TnCPND	N/A	TnCNT1 underflow	N/A	N/A
Tier Int. 2	TnDPND	TnCNT2 underflow	TnCNT2 underflow	TnCNT2 reload from TnCRB	TnCNT2 underflow

Embedded Controller Modules (Continued)

4.7.5 Timer I/O Functions

There are two I/O pins associated with each of the MFT16 modules: TAn and TBn, where n denotes the module on a given device. The functionality of TA and TB depends on the mode of operation and the value of TAEN and TBEN bits. Table 20 shows the function of TA and TB versus the selected mode of operation. Note that if TA functions as a PWM output, the initial and present value of TA is defined by TAOUT. For example, to start with TA high, TAOUT must be set (1) prior to enabling the timer clock.

Table 20. MFT16 I/O Functions

I/O	TnAEN TnBEN	Mode 1	Mode 2	Mode 3	Mode 4
		PWM and Counter	Dual Input Capture	Dual Independent Timer	Input Capture and Timer
TAn	TnAEN=0 TnBEN=X	No Output	Capture TnCNT1 into TnCRA	No Output toggle	No Output toggle
	TnAEN=1 TnBEN=X	Toggle Output on underflow of TnCNT1	Capture TnCNT1 into TnCRA and preset TnCNT1	Toggle Output on underflow of TnCNT1	Toggle Output on underflow of TnCNT1
TBn	TnAEN=X TnBEN=0	Ext. Event or Pulse Accumulate Input	Capture TnCNT1 into TnCRB	Ext. Event or Pulse Accumulate Input	Capture TnCNT2 into TnCRB
	TnAEN=X TnBEN=1	Ext. Event or Pulse Accumulate Input	Capture TnCNT1 into TnCRB and preset TnCNT1	Ext. Event or Pulse Accumulate Input	Capture TnCNT2 into TnCRB and preset TnCNT2

4.7.6 Operation in Development System

The MFT16 supports freezing the counters during breakpoints while operating in development systems. If FREEZE bit is asserted, all timer counter clocks are inhibited, and the current values of timer/counter registers TnCNT1 and TnCNT2 are frozen. Once FREEZE becomes inactive, counting resumes from the previous value. See 4.20.5 on page 273 for more details.

Embedded Controller Modules (Continued)

4.7.7 MFT16 Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

MFT16 Register Map

Mnemonic	Register Name	Type
TnPRSC	Clock Pre-Scaler Register	R/W
TnCKC	Clock Unit Control Register	R/W
TnCNT1	Timer/Counter Register 1	R/W
TnCNT2	Timer/Counter Register 2	R/W
TnCRA	Reload/Capture Register A	R/W
TnCRB	Reload/Capture Register B	R/W
TnCTRL	Timer Mode Control Register	R/W
TnICTL	Timer Interrupt Control Register	R/W
TnICLR	Timer Interrupt Clear Register	WO

Timer/Counter Register 1 (TnCNT1)

The TnCNT1 register is a word-wide read/write register that is not altered by reset. The value on power-on is unknown.

Location: MFT16 1: 00 FD80$_{16}$
MFT16 2: 00 FDA0$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TCNT1**															

Reload/Capture Register A (TnCRA)

The TnCRA register is a word-wide read/write register that is not affected by reset and thus contains random data on power-up.

Location: MFT16 1: 00 FD82$_{16}$
MFT16 2: 00 FDA2$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TCRA**															

Reload/Capture Register B (TnCRB)

The TnCRB register is a word-wide read/write register that is not affected by reset and thus contains random data on power-up.

Location: MFT16 1: 00 FD84$_{16}$
MFT16 2: 00 FDA4$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TCRB**															

Embedded Controller Modules (Continued)

Timer/Counter Register 2 (TnCNT2)

The TnCNT2 register is a word-wide read/write register that is not altered by reset. The power-up value is unknown.

Location: MFT16 1: 00 $FD86_{16}$
MFT16 2: 00 $FDA6_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TCNT2**															

Clock Pre-Scaler Register (TnPRSC)

The TnPRSC register is a byte-wide read/write register. It contains the current value of the clock pre-scaler, CLKPS. The register is cleared on reset. It defines the timer clock pre-scaler ratio.

Location: MFT16 1: 00 $FD88_{16}$
MFT16 2: 00 $FDA8_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**CLKPS**				
Reset	0	0	0	0	0	0	0	0

Bit	Description
4-0	**CLKPS (Clock Pre-Scaler).** The timer clock is generated by dividing the system clock by CLKPS+1. Therefore the maximum timer clock frequency is equal to the system clock (CLKPS=00000) and the minimum timer clock is the system clock divided by 32 (CLKPS=11111).
7-5	**Reserved.**

Clock Unit Control Register (TnCKC)

The TnCKC register is a byte-wide read/write register. It defines the clock source selection for each timer counter. The register is cleared on reset, thus disabling timer/counter 1 and timer/counter 2 clocks.

Location: MFT16 1: 00 $FD8A_{16}$
MFT16 2: 00 $FDAA_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**C2CSEL**			**C1CSEL**		
Reset	0	0	0	0	0	0	0	0

Bit	Description
2-0	**C1CSEL (Counter 1 Clock Select).** Defines the clock mode for timer/counter 1 where: **Bits** **2 1 0 Description** 0 0 0: No Clock (Counter 1 stopped) 0 0 1: Prescaled system clock 0 1 0: External Event on TBn 0 1 1: Pulse Accumulate 1 0 0: Slow-speed Clock Other: Reserved

Embedded Controller Modules (Continued)

Bit	Description
5-3	**C2CSEL (Counter 2 Clock Select).** Defines the clock mode for timer/counter 2. **Bits** **5 4 3 Description** 0 0 0: No Clock (Counter 1 stopped) 0 0 1: Prescaled system clock 0 1 0: External Event on TBn 0 1 1: Pulse Accumulate 1 0 0: Slow-speed Clock Other: Reserved
7-6	**Reserved.**

Timer Mode Control Register (TnCTRL)

The TnCTRL register is a byte-wide read/write register. It defines the mode of operation of timer/counter and TAn and TBn I/O pins. The register is cleared on reset.

Location: MFT16 1: 00 FD8C$_{16}$
MFT16 2: 00 FDAC$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**TAOUT**	**TBEN**	**TAEN**	**TBEDG**	**TAEDG**	**MDSEL**	
Reset	0	0	0	0	0	0	0	0

Bit	Description
1-0	**MDSEL (Mode Select).** Defines the MFT16 mode of operation. **Bits** **1 0 Description** 0 0: Mode 1 0 1: Mode 2 1 0: Mode 3 1 1: Mode 4
2	**TAEDG (TAn Edge Polarity).** 0: A high-to-low transition on TAn causes the action defined by the mode of operation (e.g., input capture). 1: A low-to-high transition on TAn results in the defined action.
3	**TBEDG (TBn Edge Polarity).** 0: A high-to-low transition on TBn causes the action defined by the mode of operation (e.g., input capture or external event count) 1: A low-to-high transition on TBn results in the defined action In Pulse Accumulate mode, when this bit is set to 1, the count is enabled if TBn is high. When cleared (0) and while operating in Pulse Accumulate mode, the counter is enabled if TBn is low.
4	**TAEN (TAn Enable).** Enables TAn to function either as a preset input or as a PWM output, depending on the mode of operation. If this bit is set (1), while operating in Dual Input Capture mode (mode 2), a transition on TAn causes TnCNT1 to be preset to FFFF$_{16}$. In the remaining modes of operation, setting TnAEN enables TAn to function as a PWM output. See Table 20 on page 141 for additional information.
5	**TBEN (TBn Enable).** When set (1), and while operating in either Dual Input Capture mode (mode 2) or Input Capture and Timer mode (mode 4), a transition on TBn causes the corresponding timer/counter to be preset to FFFF16. In mode 2, TnCNT1 is preset to FFFF16; in mode 4, TnCNT2 is preset to FFFF16. The bit has no effect while operating in any mode other than modes 2 or 4. See Table 20 on page 141 for additional information.

Embedded Controller Modules (Continued)

Bit	Description
6	**TAOUT (TAn Output Data).** Contains the value of TAn output when TAn is used as a PWM output. 0: TAn is low 1: TAn is high This bit is set and cleared by hardware and thus reflects the status of TAn. This bit can be read at any time. It may be used to set the initial value of TAn output in PWM mode. Note that if the hardware attempts to toggle this bit at the same time as software is writing to the bit, the software write takes precedence over the hardware update. This bit has no effect when TAn is used as input.
7	**Reserved.**

Timer Interrupt Control Register (TnICTL)

The TnICTL register is a byte-wide read/write register. It contains the interrupt enable bit and associated interrupt pending bits for the four timer interrupt sources. The TnICTL register format is shown below. The register is cleared on reset.

Location: MFT16 1: 00 FD8E$_{16}$

MFT16 2: 00 FDAE$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**TDIEN**	**TCIEN**	**TBIEN**	**TAIEN**	**TDPND**	**TCPND**	**TBPND**	**TAPND**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**TAPND (Timer Interrupt Source A Pending).** When asserted, indicates that an interrupt condition (as shown in Table 19 on page 140) has occurred. This bit can be set by either hardware or software. This bit is cleared by a writing 1 to the respective bit in Timer Interrupt Clear register. A write of 0 to this bit is ignored. The bit is cleared (0) on reset.
1	**Timer Interrupt Source B Pending (TBPND).** Same as TAPND but for a different condition, as shown in Table 19.
2	**Timer Interrupt Source C Pending (TCPND).** Same as TAPND but for a different condition, as shown in Table 19.
3	**Timer Interrupt Source D Pending (TDPND).** Same as TAPND but for a different condition, as shown in Table 19.
4	**TAIEN (Timer Interrupt A Enable)**. 0: No system interrupt occurs, but the associated pending flag TAPND is set 1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 19 **Note:** The bit can be set or cleared by software at any time.
5	**TBIEN (Timer Interrupt B Enable).** 0: No system interrupt occurs, but the associated pending flag TBPND is set 1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 19 **Note:** The bit can be set or cleared by software at any time.
6	**TCIEN (Timer Interrupt C Enable).** 0: No system interrupt occurs, but the associated pending flag TCPND is set 1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 19 **Note:** The bit can be set or cleared by software at any time.
7	**TDIEN (Timer Interrupt D Enable).** 0: No system interrupt occurs, but the associated pending flag TDPND is set 1: Enables a system interrupt based on the occurrence of a condition, as listed in Table 19 **Note:** The bit can be set or cleared by software at any time.

Embedded Controller Modules (Continued)

Timer Interrupt Clear Register (TnICLR)

The TnICLR register is a byte-wide write-only register. It controls the clear of pending flags TAPND, TBPND, TCPND and TDPND, which are located in TnICTRL register.

Location: MFT16 1: 00 FD90$_{16}$
MFT16 2: 00 FDB0$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**				**TDCLR**	**TCCLR**	**TBCLR**	**TACLR**

Bit	Description
0	**TACLR (Timer Pending A Clear).** Writing a 1 to this bit clears the TAPND flag in TnICTL register. 0: Has no effect on TAPND. The previous value of TAPND is maintained 1: Causes the TAPND flag to be cleared (0)
1	**TBCLR (Timer Pending B Clear).** Writing a 1 to this bit clears the TBPND flag in TnICTL register. 0: Has no effect on TBPND. The previous value of TBPND is maintained 1: Causes the TBPND flag to be cleared (0)
2	**TCCLR (Timer Pending C Clear).** Writing a 1 to this bit clears the TCPND flag in TnICTL register. 0: Has no effect on TCPND. The previous value of TCPND is maintained 1: Causes the TCPND flag to be cleared (0)
3	**TDCLR (Timer Pending D Clear).** Writing a 1 to this bit clears the TDPND flag in TnICTL register. 0: Has no effect on TDPND. The previous value of TDPND is maintained 1: Causes the TDPND flag to be cleared (0)
7-4	**Reserved.**

Embedded Controller Modules (Continued)

4.8 PULSE WIDTH MODULATOR (PWM)

The PWM module generates eight 8-bit PWM outputs; each may have a different duty cycle. A common 8-bit clock pre-scaler and an 8-bit down-counter determine the cycle time, the minimal possible pulse width and the duty cycle steps.

4.8.1 Features

- Eight PWM outputs
- One 8-bit fully programmable pre-scaler
- One 8-bit fully programmable down-counter
- 8-bit duty cycle control per output
- Programmable polarity per output
- Low Power mode

4.8.2 Functional Description

The PWM generates eight 8-bit PWM outputs, PWM0 to PWM7. The Duty Cycle registers 0 to 7 (DCRi, i = 0 to 7) control the duty cycle of PWM0 to PWM7 output signals, respectively. The Cycle Time register (CTR) controls the cycle time and duty cycle steps of all outputs.

The PWM use the core domain clock as a reference for its activities. The pre-scaler divider divides the PWM input clock by a pre-programmable ratio of 1:1 through 1:256, as defined by the Clock Pre-Scaler register (PRSC).

The PWM cycle is defined in terms of cycles of the pre-scaler output clock. It has a period of (CTR +1) cycles, out of which the signal will be high for DCRi cycles. INVPi, when set, may be used to inverse this behavior.

The cycle period is defined by the 8-bit down-counter. It counts using the output clock from the pre-scaler divider, starting from the value held in CTR register down to 0. When it reaches 0, the down-counter restarts from the CTR value.

The DCRi register defines the number of clock cycles during which PWMi signal is high during the complete down-counter cycle.

If Inverse PWMi bit (INVPi) in PWM Polarity register (PWMPOL) is active (1), the DCRi register defines the number of clock cycles during which PWMi signal is low in each of the down-counter complete cycles.

Figure 51 is a functional block diagram of the PWM module.

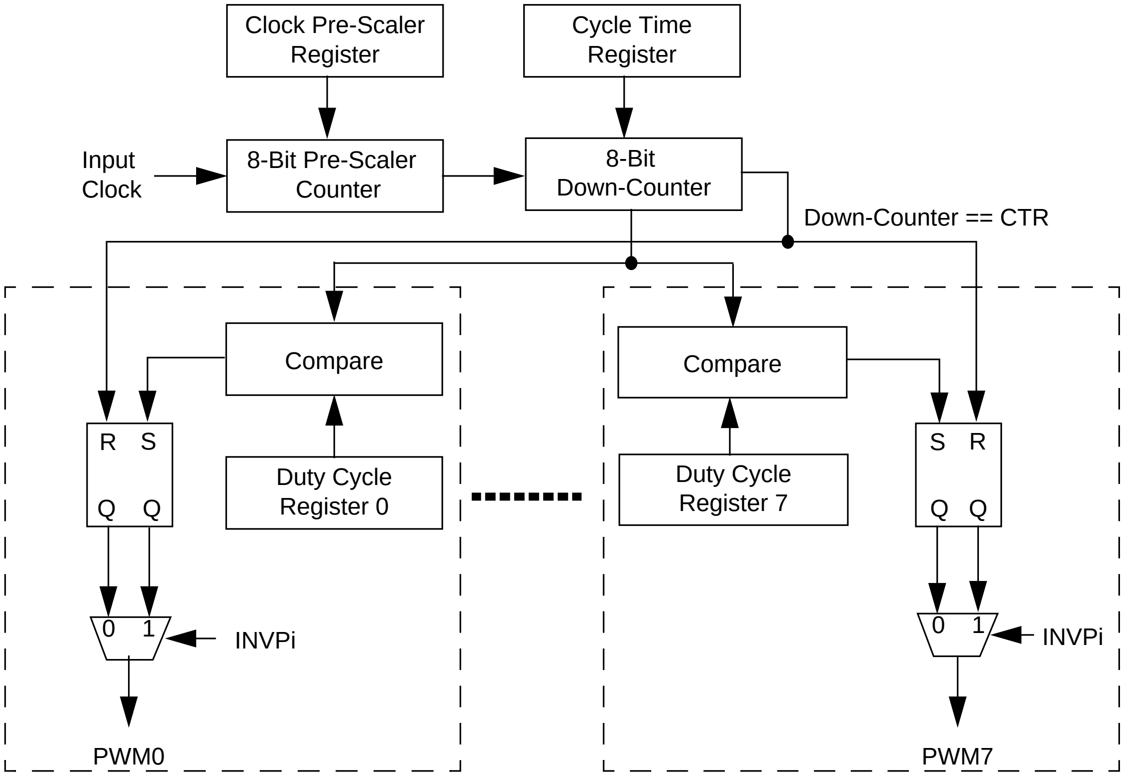


Figure 51. PWM Block Diagram

Embedded Controller Modules (Continued)

4.8.3 Cycle Time and Duty Cycle Calculation

The PWM module supports duty cycles in the range of 0% to 100%.

The PWMi output signal cycle time is:

$(PRSC + 1) \times (CTR + 1) \times T_{CLK}$

where:

- T_{CLK} is the core domain clock cycle time (i.e., the PWM input clock).
- The cycle time may range from $2 \times T_{CLK}$ to $65536 \times T_{CLK}$.

The PWMi output signal duty cycle (in %, when INVPi is 0) is:

$(DCRi + 1) / (CTR + 1) \times 100$.

Special cases:

- If the DCRi value is greater than the CTR value, PWMi signal is always low.
- If DCRi value is equal to the CTR value, PWMi signal is always high.

When Inverse PWMi bit is 1, the value of PWMi output is inverted. i.e., in the period described as 1 is 0 and vice versa.

4.8.4 Power Modes

The PWM is in Low Power mode when Power Mode bit (PWR) in PWM Control Register (PWMCNT) is 0. In this mode, the PWM input clock is disabled (stopped), but the registers are accessible and maintained. The PWMi signal is 0 when INVPi bit is 0; it is 1 when INVPi bit is 1.

The PWM is in normal power mode when PWR bit in PWMCNT register is 1. In this mode, the PWM module is enabled, its registers are accessible and its clock is functional.

The PRSC and CTR registers should be updated during Low Power mode. Otherwise, there may be unpredictable transient behavior.

4.8.5 PWM Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

PWM Register Map

Mnemonic	Register Name	Type
PRSC	Clock Pre-Scaler	R/W
CTR	Cycle Time	R/W
DCRi	Duty Cycle 0 to 7	R/W
PWMPOL	PWM Polarity	R/W
PWMCNT	PWM Control	R/W

Clock Pre-Scaler Register (PRSC)

The PRSC register controls the cycle time and the minimal pulse width. PRSC is cleared (00_{16}) on reset.

Location: 00 $FD00_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	PRSC7-0							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Pre-Scaler Divider Value.** The divider of the PWM input clock is the number defined by PRSC7-0 + 1, e.g., a value of 0016 results in a divide by 1, a value of FF16 results in a divide by 256. The contents of this register may be changed only when the PWM module is in Low Power mode. Otherwise, there may be unpredictable results.

Embedded Controller Modules (Continued)

Cycle Time Register (CTR)

The CTR register controls the cycle time and duty cycle steps, CTR is set (FF$_{16}$) on reset.

Location: 00 FD02$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	CTR7-0							
Reset	1	1	1	1	1	1	1	1

Bit	Description
7-0	**Cycle Time Value.** The 8-bit down-counter divides the pre-scaler output clock by the number defined by CTR(7:0) + 1. For example, a value of 0016 results in a divide by 1, a value of FF16 results in a divide by 256. The contents of this register may be changed only when the PWM module is in Low Power mode. Otherwise, there may be unpredictable results.

Duty Cycle Registers 0 to 7 (DCRi)

The DCRi (i = 0 to 7) registers control the duty cycle of PWMi output signal. DCRi is cleared (00$_{16}$) on reset.

Location: Channel 0 - 00 FD08$_{16}$
Channel 1 - 00 FD0A$_{16}$
Channel 2 - 00 FD0C$_{16}$
Channel 3 - 00 FD0E$_{16}$
Channel 4 - 00 FD10$_{16}$
Channel 5 - 00 FD12$_{16}$
Channel 6 - 00 FD14$_{16}$
Channel 7 - 00 FD16$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	DCRi7-0							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Duty Cycle Value.** DCRi register defines the number of clocks for which PWMi is high (from the full cycle of the PWMi cycle), when Inverse PWMi bit in PWM Polarity register is 0. If the DCRi value > CTR value, PWMi signal is always low. If DCRi value == CTR value, PWMi signal is always high. When Inverse PWMi bit is 1, the value of PWMi is inverse.

Embedded Controller Modules (Continued)

PWM Polarity Register (PWMPOL)

This register controls the polarity of PWM0 to PWM7. The register is cleared (00_{16}) on reset.

Location: 00 $FD04_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	INVP7-0							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Inverse PWM Outputs.** Each bit controls the corresponding polarity of PWM0 to PWM7; e.g., INVP0 controls PWM0. 0: The number in DCRi indicates for how many clocks (of down-counter decrements) PWMi signal is high. 1: The number in DCRi indicates for how many clocks (of down-counter decrements) PWMi signal is low.

PWM Control Register (PWMCNT)

This register controls PWM operation. The register is cleared (00_{16}) on reset.

Location: 00 $FD06_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	Reserved							PWR
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**Power Mode.** This bit controls the operating mode of the PWM, as follows: 0: Low Power mode. PWM input clock is disabled (stopped). Its registers are accessible and maintained. PWMi signal is 0 when INVPi bit is 0. PWMi signal is 1 when INVPi bit is 1. 1: Normal Power mode. PWM module is enabled. Its registers are accessible and its clock is functional.
7-1	**Reserved.**

4.9 UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER-TRANSMITTER (USART)

The USART is a full-duplex synchronous/asynchronous receiver-transmitter that supports a wide range of software programmable baud rates and data formats. It handles automatic parity generation and several error detection schemes. DMA transfers are supported to enable fast processor-independent receive and transmit.

4.9.1 Features

- Full duplex double-buffered receiver-transmitter
- Synchronous operation
- Asynchronous operation
- Programmable baud rate between CLK/2 and CLK/32768 baud
- Numerous framing formats
  - — seven, eight or nine data bits
  - — one or two stop bits
  - — odd, even, mark, space or no parity
- Hardware support of parity-bit generation during transmission and parity-bit check during reception
- Interrupt on transmit buffer empty, receive buffer full and receive error conditions with separate enable
- Software-controlled break transmission and detection
- Software-controlled reset, enable/disable
- Internal diagnostic capability
- Automatic error detection
  - — Parity Error
  - — Framing Error
  - — Data Overrun Error
- 9-bit Attention mode
- DMA support for transmit and receive with separate enable

4.9.2 Functional Overview

The USART is composed of the following functional units:

- Transmitter
- Receiver
- Baud rate generator
- Control and error detection

Figure 52 shows a block diagram of the USART.

Embedded Controller Modules (Continued)

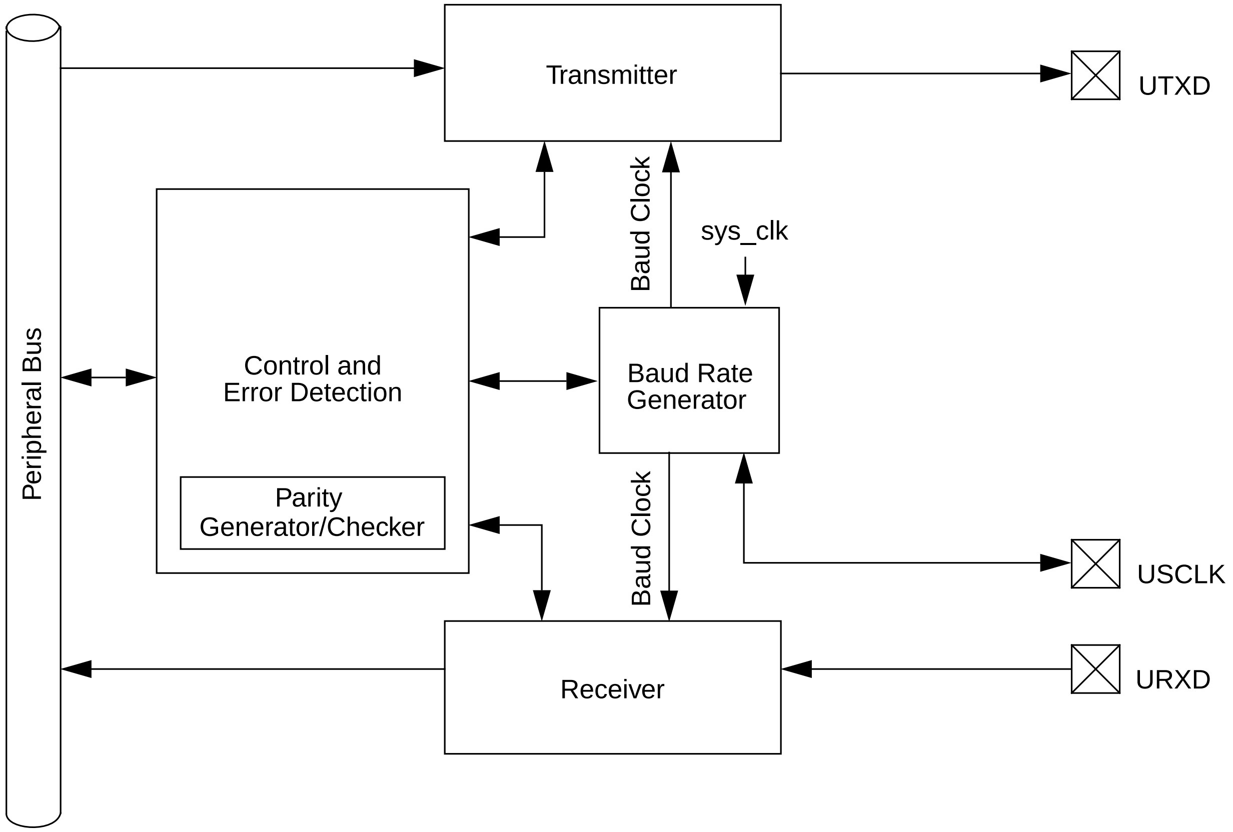


Figure 52. USART Block Diagram

Each functional unit is described briefly in this section.

Transmitter

The Transmitter consists of an 8-bit transmit shift register and an 8-bit transmit buffer. Data is loaded in parallel from the buffer into the shift register and then transmitted serially on the UTXD pin.

Receiver

The Receiver consists of an 8-bit receive shift register and an 8-bit receive buffer. Data is received serially into the shift register from the URXD pin. After eight bits have been received, the contents of the shift register are transferred in parallel to the receive buffer.

Baud Rate Generator

The Baud Rate Generator generates the clock for the Asynchronous and Synchronous modes of operation. It consists of two registers and a two-stage counter. The registers are used to specify a pre-scaler value and baud rate divisor. The first stage of the counter divides the CLK clock in 0.5 increments based on the value of the pre-scaler. The second stage of the counter divides the output of the first stage in integer increments based on the value of the baud rate divisor.

Control and Error Detection

This unit contains the control registers (see Section 4.9.4 on page 158), control logic, error detection circuitry and parity generator/checker. It supports:

- Selection of the data format, mode of operation, clock source and parity type
- Generation and detection of parity
- Reporting of parity errors
- Detection and reporting of data overrun and frame errors
- Interrupts on transmit buffer empty, receive buffer full, receive error and delta clear to send conditions
- Generation and detection of line breaks

4.9.3 Operation

The USART has two basic modes of operation; Synchronous and Asynchronous. In addition, two special Synchronous and Asynchronous modes, attention and diagnostic, are available. This section describes the operating modes of the USART.

Asynchronous Mode

USART Asynchronous mode enables the device to communicate with other devices using two communication signals: transmit (UTXD) and receive (URXD).

In Asynchronous mode, Transmit Shift register (TSFT) and Transmit Buffer register (UTBUF) double buffer data for transmission. To transmit a character, a data byte is loaded into UTBUF register. The data is then transferred to TSFT register. While TSFT register is shifting out the current character (LSB first) on the UTXD pin, UTBUF register is loaded by software with the next byte to be transmitted. When TSFT completes transmitting the last stop bit of the current frame, the contents of UTBUF are transferred to TSFT register and the Transmit Buffer Empty flag (TBE) is set. The TBE flag is automatically reset by the USART when the software loads a new character into UTBUF register. During transmission, the XMIP bit is set high by the USART. This bit is reset only after the USART has sent the last stop bit of the current character and UTBUF register is empty. UTBUF register is a read/write register. TSFT register is not user accessible.

In Asynchronous mode, the input frequency to the USART is 16 times the baud rate, i.e., there are 16 clock cycles per bit time. In Asynchronous mode, the baud rate generator is always used as the UART clock.

Receive Shift register (RSFT) and Receive Buffer register (URBUF) double buffer the data being received. The USART receiver continually monitors the signal on the URXD pin for a low level to detect the beginning of a start bit. On sensing this low level, the USART waits for seven input clock cycles and samples again three times. If all three samples still indicate a valid low, the receiver considers this to be a valid start bit, and the remaining bits in the character frame are each sampled three times around the mid-bit position. For any bit following the start bit, the logic value is found by majority voting, i.e., the two samples with the same value define the value of the data bit. Figure 53 illustrates the process of start-bit detection and bit sampling. Serial data input on the URXD pin is shifted into RSFT register. On receiving the complete character, the contents of RSFT register are copied into URBUF register and Receive Buffer Full flag (RBF) is set. The RBF flag is automatically reset when software reads the character from URBUF register. The RSFT register is not user accessible.

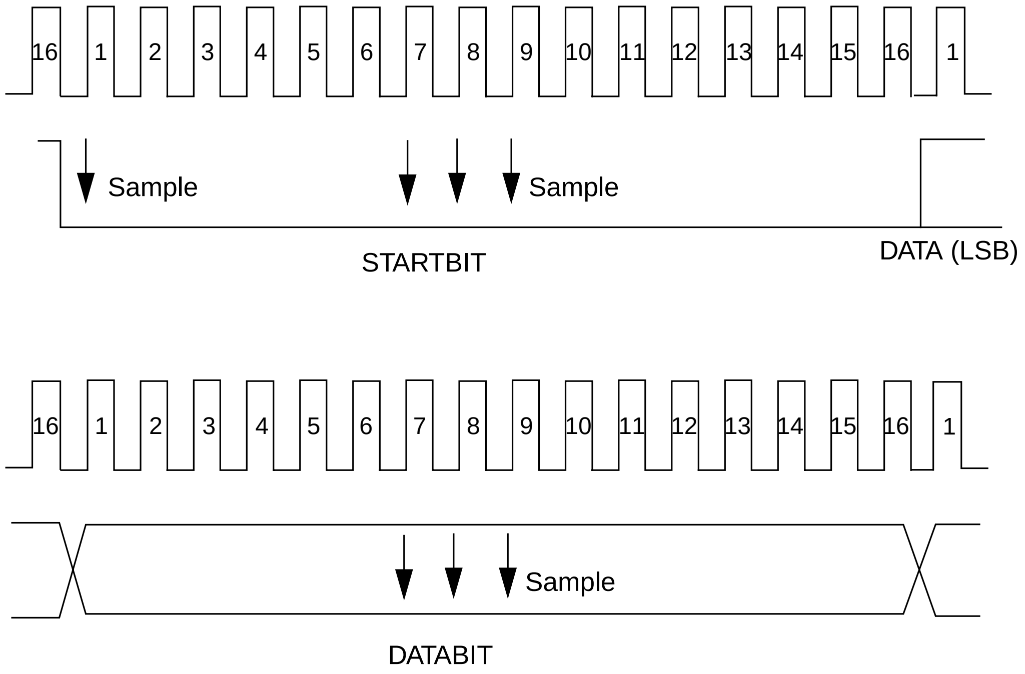


Figure 53. USART Asynchronous Communication

Synchronous Mode

The USART Synchronous mode enables the device to communicate with other devices using three communication signals: transmit (UTXD), receive (URXD) and clock (USCLK). In this mode, data bits are transferred synchronously using the USART clock signal. As shown in Figure 54, the data is transmitted on the rising edge and received on the falling edge of the clock. Data is transmitted and received with the LSB first.

Embedded Controller Modules (Continued)

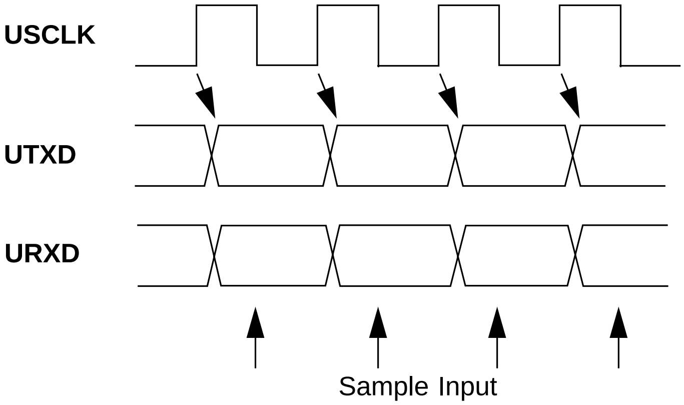


Figure 54. USART Synchronous Communication

In Synchronous mode, Transmit Shift register (TSFT) and Transmit Buffer register (UTBUF) double buffer data for transmission. To transmit a character, a data byte is loaded into TBUF register. The data is then transferred to TSFT register. The TSFT shifts out one bit of the current character, LSB first, on each rising edge of the clock. While the TSFT is shifting out the current character on the UTXD pin, UTBUF register can be loaded by the software with the next byte to be transmitted. When TSFT completes transmitting the last stop bit within the current frame, the contents of UTBUF are transferred to TSFT register, and the Transmit Buffer Empty flag (TBE) is set. The TBE flag is automatically reset by the USART when the software loads a new character into UTBUF register. During transmission, XMIP bit is set high by the USART. This bit is reset only after the USART has sent the last frame bit of the current character and UTBUF register is empty.

Receive Shift register (RSFT) and Receive Buffer register (URBUF) double buffer the data being received. Serial data input on the URXD pin is shifted into RSFT register at the first falling edge of the clock. Each subsequent falling edge of the clock causes an additional bit to be shifted into RSFT register. The USART assumes a complete character has been received after the correct number of rising edges on USCLK (based on the selected frame format) has been detected. On receiving a complete character, the contents of RSFT register are copied into URBUF register, and Receive Buffer Full flag (RBF) is set. The RBF flag is automatically reset when software reads the character from URBUF register.

The transmitter and receiver may be clocked from either an external source available on the USCLK pin or the internal baud rate generator. If the internal baud rate generator is used, the baud clock is output on the USCLK pin.

Attention Mode

Attention mode is available for networking this device with other processors. This mode requires the 9-bit data format with no parity. The number of start/stop bits are user selectable. In this mode, two types of 9-bit characters are sent on the network: address characters consisting of eight address bits and a “1” (one) in the ninth bit position and data characters consisting of eight data bits and a “0” (zero) in the ninth bit position.

While in Attention mode, the USART receiver monitors the communication flow but ignores all characters until an address character is received. On the receipt of an address character, the contents of Receive Shift register are copied to the receive buffer. The RBF flag is set and an interrupt (if enabled) is generated. The ATN bit is automatically reset to zero, and the USART begins receiving all subsequent characters. The software must examine the contents of URBUF register and respond by accepting the subsequent characters (by leaving the ATN bit reset) or waiting for the next address character (by setting the ATN bit again).

The operation of the USART transmitter is not affected by the selection of this mode. The value of the ninth bit to be transmitted is programed by setting the STPXB9 bit appropriately. The value of the ninth bit received is read from the RB9 bit.

Diagnostic Mode

This mode is available for diagnostic tests of the USART. In this mode, the UTXD and URXD pins are internally connected, and data that is shifted out of Transmit Shift register is immediately transferred to Receive Shift register. This mode supports only the 9-bit data format with no parity. The number of start and stop bits is user selectable.

Frame Format Selection

The format shown in Figure 55 consists of a start bit, seven data bits (excluding parity) and one or two stop bits. If parity bit generation is enabled by setting the PEN bit, a parity bit is generated and transmitted following the seven data bits.

Embedded Controller Modules (Continued)

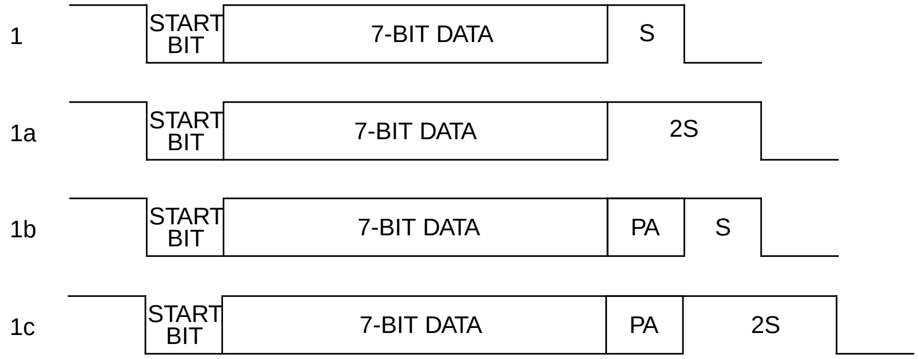


Figure 55. Seven Data Bit Frame Options

The format shown in Figure 56 consists of one start bit, eight data bits (excluding parity) and one or two stop bits. If parity bit generation is enabled by setting the PEN bit, a parity bit is generated and transmitted following the eight data bits.

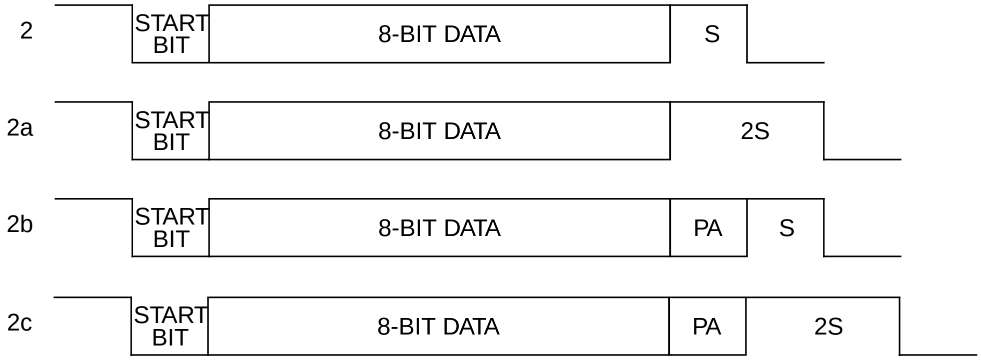


Figure 56. Eight Data Bit Frame Options

The format shown in Figure 57 consists of one start bit, nine data bits and one or two stop bits. This format also supports the USART attention feature. When operating in this format, all eight bits of UTBUF and URBUF are used for data. The ninth data bit is transmitted and received using two bits in the control registers, called STPXB9 and RB9. Parity is not generated or verified in this mode.

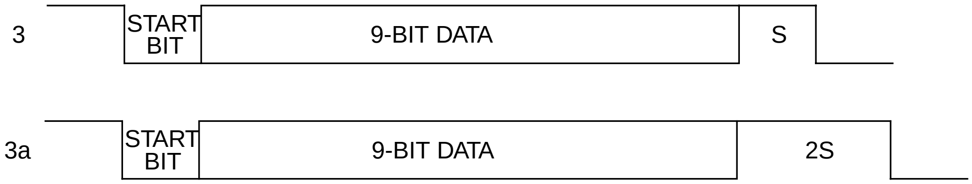


Figure 57. Nine Data Bit Frame Options

Baud Rate Generator

The Baud Rate Generator provides the basic baud clock from the system clock. The system clock is passed through a two-stage divider chain consisting of a 5-bit baud rate pre-scaler (PSC) and an 11-bit baud rate divisor (DIV).

The correspondences between the 5-bit pre-scaler select (PSC) and pre-scaler factors are shown in Table 21.

Embedded Controller Modules (Continued)

Table 21. Pre-Scaler Factors

Pre-Scaler Select	Pre-Scaler Factor	Pre-Scaler Select	Pre-Scaler Factor
00000	NO CLOCK	10000	8.5
00001	1	10001	9
00010	1.5	10010	9.5
00011	2	10011	10
00100	2.5	10100	10.5
00101	3	10101	11
00110	3.5	10110	11.5
00111	4	10111	12
01000	4.5	11000	12.5
01001	5	11001	13
01010	5.5	11010	13.5
01011	6	11011	14
01100	6.5	11100	14.5
01101	7	11101	15
01110	7.5	11110	15.5
01111	8	11111	16

A pre-scaler factor of zero corresponds to NO CLOCK. The NO CLOCK condition is the USART Power-Down mode. In this mode, the USART clock is turned off in order to reduce power consumption. The user should set the pre-scaler factor to zero (NO CLOCK) before selecting a new baud rate. Altering the baud rate while the USART is in operation could cause incorrect data to be received or transmitted. UPSR register must contain a value other than zero when an external clock is used at USCLK.

In Asynchronous mode, the baud rate is calculated by:

$$BR = \texttt{SYS\_CLK/(16 x N x P)}$$

where:

- BR is the baud rate
- SYS_CLK is the system clock
- N is the value of the baud rate divisor + 1
- P is the pre-scaler divide factor selected by the value in the PSR register

The divide by 16 operation is performed because in Asynchronous mode, the input frequency to the USART is 16 times the baud rate. In Synchronous mode, the input clock to the USART is equal to the baud rate.

Interrupts

The USART is capable of generating interrupts on one of the following conditions:

- Receive Buffer Full
- Receive Error
- Transmit Buffer Empty

Embedded Controller Modules (Continued)

Figure 58 shows a diagram of the interrupt sources and associated enable bits.

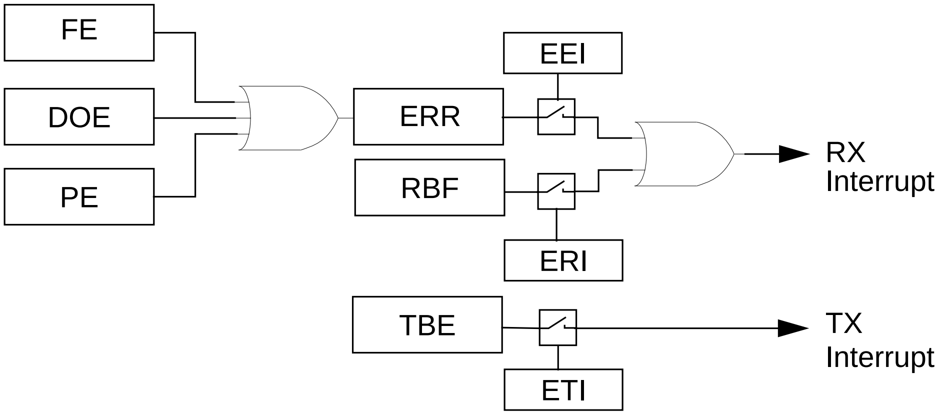


Figure 58. USART Interrupt Sources

The interrupts can be individually enabled or disabled using Enable Transmit Interrupt (ETI), Enable Receive Interrupt (ERI) and Enable Receive Error Interrupt (EEI) bits in UICTRL register.

A transmit interrupt is generated when both the TBE and ETI bits are set. To remove this interrupt, the software must either disable the interrupt by clearing the ETI bit or write to UTBUF register (thus clearing the TBE bit).

A receive interrupt is generated on two conditions:

- If both the RBF and ERI bits are set. To remove this interrupt, the software must either disable the interrupt, by clearing the ERI bit, or read from URBUF register (thus clearing the RBF bit).
- If both the ERR and the EEI bits are set. To remove this interrupt, the software must either disable it by clearing the EEI bit, or read USTAT register, which causes ERR flag to be cleared.

DMA Support

The USART can operate with either one or two DMA channels. Two DMA channels are required for processor-independent full-duplex operation. Both receive and transmit DMA can be enabled individually.

If the transmit DMA is enabled (ETD=1), the USART issues a DMA request every time the TBE flag is set. Enabling the transmit DMA automatically disables the TX interrupt independent of the value of the ETI bit.

Enabling the receive DMA (ERD=1) causes a DMA request to be asserted every time the Receive Buffer Full flag (RBF) is set. Once the receive DMA is enabled the RX interrupt is automatically disabled independent of the value of the ERI bit. However, to detect errors during reception the receive error interrupt should be enabled (EEI=1) while using the DMA.

Break Generation and Detection

A line break is generated when BRK bit is set in MDSL register. The UTXD line remains low until the user resets the BRK bit.

A line break is detected if URXD remains low for a time equivalent to 10 bit times or longer, after a missing stop bit has been detected.

Parity Generation and Detection

Parity is only generated or checked with 7- and 8-bit data formats. It is not generated or checked in Diagnostic Loopback mode, Attention mode or in Normal mode with 9-bit data format. Parity generation and checking is enabled and disabled via PEN bit in UFRS register. PSEL bits in UFRS register are used to select odd, even, mark or space parity.

ISE Mode Operation

The USART module supports breakpoint operation by preserving some of the status bits of the USTAT and UICTRL registers. While the FREEZE bit is asserted, the PE, FE, DOE, BKD and DCTS bits are not cleared on a read of STAT register.

4.9.4 USART Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

USART Register Map

Mnemonic	Register Name	Type
URBUF	Receive Data Buffer Register	RO
UTBUF	Transmit Data Buffer Register	R/W
UPSR	Baud Rate Pre-Scaler Register	R/W
UBAUD	Baud Rate Divisor Register	R/W
UFRS	Frame Select Register	R/W
UMDSL	Mode Select Register	R/W
USTAT	Status Register	RO
UICTRL	Interrupt Control Register	Varies per bit

Receive Data Buffer Register (URBUF)

Location: 00 FD22$_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**URBUF**							

Transmit Data Buffer Register (UTBUF)

Location: 00 FD20$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**UTBUF**							

Baud Rate Pre-Scaler Register (UPSR)

Byte-wide read/write register containing the 5-bit pre-scaler value in bit 7 (MSB) through bit 3 (LSB) and the upper three bits of the baud divisor in bit 2 (MSB) through bit 0 (LSB). The register is cleared (00$_{16}$) on reset.

Location: 00 FD2E$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**UPSC**					**UDIV[10]: UDIV[8]**		
Reset	0	0	0	0	0	0	0	0

Baud Rate Divisor Register (UBAUD)

This byte-wide read/write register contains the lower eight bits of the baud rate divisor. The UPSR register contains the upper three bits. The register is cleared (00$_{16}$) on reset.

Location: 00 FD2C$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**UDIV[7]: UDIV[0]**							
Reset	0	0	0	0	0	0	0	0

Embedded Controller Modules (Continued)

Frame Select Register (UFRS)

This byte-wide read/write register controls the selection of the frame format, including number of data bits, number of stop bits and parity. The register is cleared (00_{16}) on reset.

Location: 00 $FD28_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**PEN**	**PSEL**		**XB9**	**STP**	**CHAR**	
Reset	0	0	0	0	0	0	0	0

Bit	Description
1-0	**CHAR.** Selects the number of data bits per frame. Note that the parity bit is not included in the number of data bits. **Bits** **1 0 Description** 0 0: Frame contains eight data bits 0 1: Frame contains seven data bits 1 0: Frame contains nine data bits 1 1: Loopback mode selected; frame contains nine data bits.
2	**STP.** Programs the number of stop bits to be transmitted. 0: One stop bit transmitted 1: Two stop bits transmitted
3	**XB9.** Contains the value of the 9th data bit for transmission only. The bit has no effect while operating with seven or eight data bits per frame. 0: Transmit 0 as 9th data bit 1: Transmit 1 as 9th data bit
5-4	**PSEL.** Controls the mode of parity bit generation and checking. Note that while operating with nine data bits-per-frame, the parity bit is omitted. In this case, the value of PSEL has no effect. **Bits** **5 4 Description** 0 0: Odd parity 0 1: Even parity 1 0: Mark (1) 1 1: Space (0)
6	**PEN.** Enables or disables the generation of a parity bit generation and parity check. Note that there is no parity bit while operating in the nine data bits-per-frame mode. In this case, this bit has no effect. 0: Parity disabled 1: Parity enabled
7	**Reserved.**

Embedded Controller Modules (Continued)

Mode Select Register (UMDSL)

This byte-wide read/write register controls the selection of the clock source, Synchronous mode, Attention mode and line break generation. It contains the enable bits for the DMA channels. The register is cleared (00_{16}) on reset.

Location: 00 $FD2A_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**ERD**	**ETD**	**CKS**	**BRK**	**ATN**	**MOD**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**MOD.** Selects the Synchronous or Asynchronous mode of operation: 0: Asynchronous mode 1: Synchronous mode
1	**ATN.** Selects the Attention mode of operation. Cleared by hardware after reception of an address frame that is a 9-bit character with a “1” in the ninth bit position. 0: Disable Attention mode 1: Enable Attention mode
2	**BRK.** Setting the bit (1) causes UTXD to go low. UTXD remains low until the bit is cleared (0) by the user.
3	**CKS.** Controls the source of the clock while operating in Synchronous mode (MOD=1). 0: USART operates from the baud rate generator and outputs the baud rate clock on USCLK 1: USART operates from an external clock provided on USCLK While the USART is operated in Asynchronous mode (MOD=0), the bit has no effect.
4	**ETD.** 0: No DMA request is asserted for transmit operations 1: DMA request is asserted when the Transmit Buffer Empty (TBE) flag is set (1)
5	**ERD.** 0: No DMA request is asserted for receive operations 1: DMA request is asserted when the Receive Buffer Full (RBF) flag is set (1)
7-6	**Reserved.**

Status Register (USTAT)

This byte-wide, read-only register contains the receive and transmit status bits. The register is cleared (00_{16}) on reset.

Location: 00 $FD26_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**XMIP**	**RB9**	**BKD**	**ERR**	**DOE**	**FE**	**PE**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**PE.** The bit is set when a parity error is detected within a received character. The bit is cleared by the hardware when USTAT register is read. 0: No parity error detected 1: Parity error detected in a received byte since the last time USTAT was read
1	**FE.** The bit is set when the USART fails to receive a valid stop bit at the end of a frame. Automatically cleared on read of USTAT. 0: No framing error detected 1: Framing error detected on a received byte since the last time USTAT was read
2	**DOE.** The bit is set when a new character is received and transferred to RBUF before the software has read the previous character. Automatically cleared on read of the USTAT. 0: No data overrun error detected 1: Data overrun error detected since the last time USTAT was read
3	**ERR.** The bit is set any time DOE, FE or PE is set. Automatically cleared if DOE, FE and PE are all zero. This bit is read only. Any attempt to write to the bit by software does not alter its present value. 0: No DOE, FE or PE has occurred since the last time USTAT register was read. 1: A DOE, FE or PE error has occurred since the last time USTAT register was read.
4	**BKD.** If set, indicates that a line break condition has occurred. A break condition is detected if RXD remains low for a least ten bit times after a missing stop bit has been detected at the end of a frame. The bit is cleared under the following conditions: – On a read of USTAT register, if the break condition on RXD is no longer present. If RXD is still low when USTAT register is read, the bit is not cleared. – If the read of USTAT register did not cause the bit to be cleared because the break condition on RXD was still in effect, the hardware clears the bit as soon as the break condition no longer exists, i.e., RXD returns to a high level.
5	**RB9.** Contains the ninth data bit of the last frame received when operating with the 9-bit data format. 0: “0” received in ninth bit position 1: “1” received in ninth bit position
6	**XMIP.** Indicates that the USART is transmitting data. It is reset by hardware at the end of the last frame bit. 0: USART is not transmitting 1: USART is transmitting
7	**Reserved.**

Embedded Controller Modules (Continued)

Interrupt Control Register (UICTRL)

This byte-wide register contains the interrupt enable bits and the interrupt status flags. The register is set to 01_{16} on reset.

Location: 00 $FD24_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**EEI**	**ERI**	**ETI**	**Reserved**			**RBF**	**TBE**
Reset	0	0	0	0	0	0	0	1

Bit	Type	Description
0	RO	**TBE.** The bit is set by the hardware when the USART transfers data from UTBUF register to TSFT register for transmission. It is automatically cleared on the next write to UTBUF register. The bit is set on reset. 0: Transmit buffer not empty 1: Transmit buffer empty
1	RO	**RBF.** The bit is set by the hardware when the USART has received a complete data frame and transferred the data from RSFT register to URBUF register. The bit is automatically cleared when RBUF register is read. 0: Receive buffer not full. New data has not been transferred to RBUF since the last time it was read. 1: Receive buffer full. RBUF contains new data since the last time it was read.
4-2		**Reserved.**
5	R/W	**ETI.** An interrupt is generated when the TBE flag is set. Read/write bit. 0: Disable transmitter interrupt 1: Enable transmitter interrupt
6	R/W	**ERI.** An RX interrupt is generated when the RBF flag is set. Read/write bit. 0: Disable receiver interrupt 1: Enable receiver interrupt
7	R/W	**ETI.** An RX interrupt is generated when the ERR flag is set, indicating that a receive error has occurred. 0: Disable receive error interrupt 1: Enable receive error interrupt

4.9.5 Usage Hints

Calculating the Baud Rate in Asynchronous Mode

The equation for calculating the baud rate is:

$$\mathtt{BR = SYS\_CLK/(16xNxP)}$$

where:

- BR is the baud rate
- SYS_CLK is the system clock
- N is the value of the baud rate divisor + 1
- P is the pre-scaler divide factor selected by the value in the PSR register

Assuming a system clock of 5 MHz and a desired baud rate of 9600, the NxP term, according to the equation above, is:

$$\mathtt{NxP = (5x10^{6})/(16x9600) = 32.552}$$

The NxP term is then divided by each pre-scaler factor in Table 21 on page 156 to obtain a value closest to an integer. The factor for this example is 6.5:

$$\mathtt{N = 32.552/6.5 = 5.008\ (N = 5)}$$

Embedded Controller Modules (Continued)

The baud rate register is programed with a baud rate divisor of 4 (N = baud rate divisor +1). This produces a baud clock of:

$$BR = (5x10^6)/(16x5x6.5) = 9615.385$$

$$\% \ error = (9615.385-9600)/9600 = 0.16$$

Note that the percent error is much lower than would be possible without the non-integer pre-scaler factor. Refer to the table below for more examples.

System Clock	Desired Baud Rate	N	P	Actual Baud Rate	Percent Error
4 MHz	9600	2	13	9615.385	0.16
5 MHz	9600	5	6.5	9615.385	0.16
10 MHz	19200	5	6.5	19230.769	0.16
20 MHz	19200	5	13	19230.769	0.16

Calculating the Baud Rate in Synchronous Mode

The equation for calculating the baud rate is:

$$BR = SYS\_CLK/(2xNxP)$$

where:

- BR is the baud rate
- SYS_CLK is the system clock
- N is the value of the baud rate divisor + 1
- P is the pre-scaler divide factor selected by the value in the PSR register

The same procedure is used for determining the values of N and P, as in Asynchronous mode. However, non-integer prescale values are not allowed.

4.10 TIMER AND WATCHDOG (TWD)

The Timer and WATCHDOG module (TWD) generates the clocks and interrupts used for timing periodic functions in the system. It also provides WATCHDOG protection over software execution.

The TWD provides flexibility in system configuration by enabling the configuration of various clock ratios. After setting the TWD configuration, the software can lock it to give a higher level of protection against erroneous software action. Once a section of the TWD is locked, only reset releases it. See Figure 59 for the TWD block diagram.

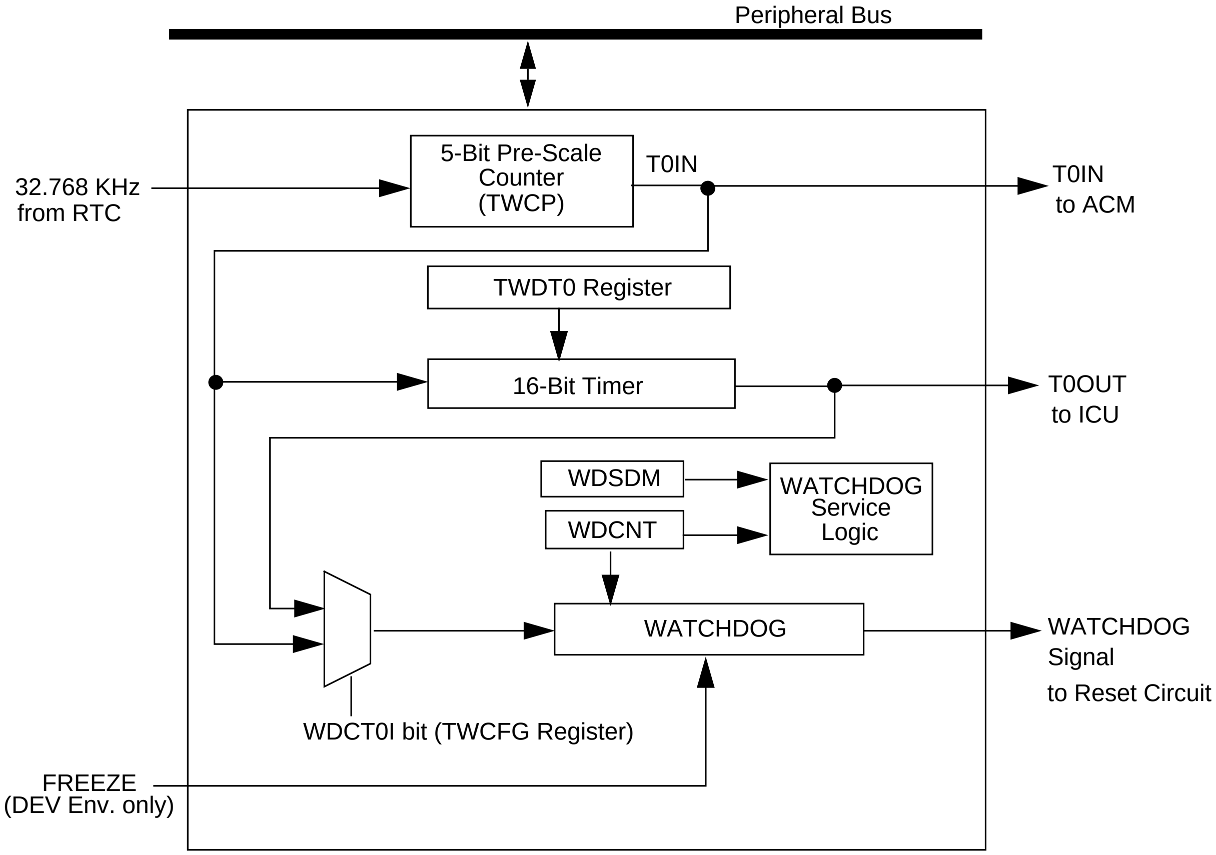


Figure 59. Timer and WATCHDOG Block Diagram

4.10.1 Features

- 32.768 KHz input clock
- Programmable pre-scale counter
- 16-bit programmable periodic interrupt timer
- 8-bit WATCHDOG counter
- WATCHDOG signal generation in response to failure detection, such as:
  - — WATCHDOG service performed too early
  - — WATCHDOG service performed too late
  - — Wrong DATA used in a service by data match

4.10.2 Functional Description

Input Clock

The TWD bases all its counting activities on a 32.768 KHz clock. The WATCHDOG can count using a division of the 32 KHz clock (either T0OUT or T0IN).

Embedded Controller Modules (Continued)

Pre-Scale

A pre-scale counter divides the input clock (32.768 KHz) by a factor of 2^{MDIV}. MDIV in TWCP register is in the range of 0 through 5 (i.e., divide ratio of 1:1 through 1:32). The pre-scaled output is used as an input clock for a 16-bit timer (TWDT0) and is referred to as T0IN.

TWD Timer 0

TWD Timer 0 is a 16-bit, programmable, automatically re-triggered down-counter. It counts on the rising edge of T0IN. It starts from the value loaded to TWDT0 register down to zero and then restarts counting from TWDT0 at the next T0IN cycle.

When the counter reaches 0, T0OUT is set (1) for one T0IN cycle. This makes the Timer 0 cycle:

(TWDT0 + 1) x T0IN-cycle.

T0OUT is input to the ICU and can be used as the time base for activities such as system tick.

When TWDT0 is loaded with a new value, the counter uses it the next time it restarts counting (i.e., after reaching zero). If RST in Timer Control register (T0CSR) is written 1, the timer is restarted on the next rising edge of T0IN.

Notes:

- RST bit in T0CSR register is cleared after completing this load.
- When MDIV in TWCP register is 0, the timer counter may skip one count when loaded with a new value.

WATCHDOG Operation

The WATCHDOG is an 8-bit down counter, operating on the rising edge of its currently selected clock source. On reset, it is disabled (i.e., it does not count and no WATCHDOG signal is generated). A write to the WATCHDOG Count register (WDCNT) or the WATCHDOG Service Data Match (WDSDM) register either starts the counter or, if WATCHDOG is already running, performs a restart (“touch”) operation. Once the WATCHDOG is counting down, only a reset can stop it.

Writing to WDCNT register is enabled while LWDCNT in TWCFG register is 0. A write to WDCNT starts the WATCHDOG, and it begins counting down from the written value. If the service on data match is enabled (WDSDME in TWCFG register is 1), writing to WDSDM register with $5C_{16}$ restarts the WATCHDOG counter from the value stored in WDCNT.

A WATCHDOG signal is triggered if one of the following occurs:

- The counter reaches zero (too late service).
- The WATCHDOG is written to more than once per WATCHDOG clock cycle for the currently selected clock (too early service). Writing to the WATCHDOG more than once per three WATCHDOG clock cycles (for the currently selected clock) may cause the WATCHDOG signal to trigger.
- Data other than $5C_{16}$ is written to WDSDM when WDSDME in TWCFG register is 1.

WATCHDOG Clock Source Selection

Select the clock source as follows:

- WDCT01 bit in TWCFG register is 0: T0OUT
- WDCT01 bit in TWCFG register is 1: T0IN

Changing the WATCHDOG clock source may cause it to gain or lose one clock cycle.

Notes:

- When MDIV in TWCP register is 0, the WATCHDOG counter may skip one count when loaded with a new value.
- After activating WATCHDOG, avoid entering Idle mode in the first four low-frequency clock cycles.

TWD Control and Configuration

The TWD Configuration register (TWCFG) allows you to:

- Set the WATCHDOG clock source: T0IN or T0OUT
- Enable WATCHDOG service on write to WDSDM register
- Define which of TWCFG, TWCPR, TWDT0, T0CSR and WDCNT is locked.

Once LTWCFG, LTWCP, LTWDT0 or LWDCNT, in TWCFG register, is set its respective resources are locked and can be cleared only by reset. Setting any of these registers prevents runaway software from tampering with the respective WATCHDOG function.

Operation in Idle Mode

The TWD is active in Idle mode: the counters continue to function, and interrupts and error signals are issued.

Write operations to TWCP, TWDT0 and WDCNT may be delayed by up to three 32.768 KHz clock cycles. The software should avoid entering Idle mode during this period. WDTLD bit in T0CSR register indicates when it is safe to switch power modes.

4.10.3 TWD Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

TWD Register Map

Mnemonic	Register Name	Type
TWCFG	Timer and WATCHDOG Configuration Register	R/W
TWCP	Timer and WATCHDOG Clock Pre-Scaler Register	R/W
TWDT0	TWD Timer 0 Register	R/W
T0CSR	TWDT0 Control and Status Register	R/W
WDCNT	WATCHDOG Count Register	WO
WDSDM	WATCHDOG Service Data Match Register	WO

Timer and WATCHDOG Configuration Register (TWCFG)

The TWCFG register is a byte-wide, read/write register. It defines the WATCHDOG clock input and service method and enables TWD control register locking. Setting the required configuration and locking the TWCFG stops the software from interfering with the WATCHDOG operation. On reset, non-reserved bits of TWCFG are initialized to 0.

Location: 00 FEE0$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**WDSDME**	**WDCT0I**	**LWDCNT**	**LTWDT0**	**LTWCP**	**LTWCFG**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**LTWCFG.** 0: Enables read/write from/to TWCFG register 1: Any data written to it is ignored and reading from it returns unpredictable values Once LTWCFG is set, it can only be cleared by reset.
1	**LTWCP.** 0: Enables read/write from/to TWCP register 1: Any data written to it is ignored and reading from it returns unpredictable values Once LTWCP is set, it can only be cleared by reset.
2	**LTWDT0.** 0: Enables read/write from/to TWDT0 and T0CSR registers 1: Registers cannot be written to and TWDT0 cannot be read. Any data written to TWDT0 or T0CSR is ignored. Reading from TWDT0 returns unpredictable values. Once LTWDT0 is set, it can only be cleared by reset.
3	**LWDCNT.** 0: Enables write to WDCNT register 1: Any data written to it is ignored and reading from it returns unpredictable values Once LWDCNT is set, it can only be cleared by reset. When WDSDME bit is cleared, touch operations (i.e., writing to WDCNT register) may be performed when LWDCNT bit is either 0 or 1.
4	**WDCT0I.** 0: Selects T0OUT clock as the WATCHDOG clock 1: Selects T0IN as the input clock The hardware clock source selection overrides this clock selection.

Bit	Description
5	**WDSDME.** Selects the WATCHDOG touch mechanism 0: Disables the WATCHDOG service using WDSDM register. In this case, the WATCHDOG should be serviced by writing a value to WDCNT register. When this bit is cleared, write operations to WDSDM are ignored. 1: Selects the use of data match using the WDSDM mechanism.
7-6	**Reserved.**

Timer and WATCHDOG Clock Pre-Scaler Register (TWCP)

The TWCP register is a byte-wide, read/write register. It defines the pre-scale ratio of the input clock and generates the T0IN clock. On reset, the non-reserved bits of TWCP are initialized to 0.

Location: 00 FEE2$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**MDIV**		
Reset	0	0	0	0	0	0	0	0

Bit	Description
2-0	**MDIV.** Defines the pre-scale ratio of the input clock. The pre-scale ratio is 2^{MDIV}. The value of MDIV must be in the range of 0-5, providing a pre-scale ratio of 1 to 32. MDIV allowed values: **Bits** **2 1 0 Clock Ratio** 0 0 0: 1:1 0 0 1: 1:2 0 1 0: 1:4 0 1 1: 1:8 1 0 0: 1:16 1 0 1: 1:32 Other Reserved
7-3	**Reserved.**

TWD Timer 0 Register (TWDT0)

The TWDT0 register is a read/write register. It defines the T0OUT interrupt rate. On reset, this register is initialized to FFFF$_{16}$.

Location: 00 FEE4$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**PRESET**															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Description
15-0	**PRESET.** Defines the counter preset value. Whenever the counter reaches zero, it starts counting down from this value. The T0OUT frequency is the T0IN frequency divided by (PRESET+1). The allowed values of the PRESET field are 0001$_{16}$ through FFFF$_{16}$.

Embedded Controller Modules (Continued)

TWDT0 Control and Status Register (T0CSR)

The T0CSR register is a read/write register. It controls the operation and provides the status of the T0 timer. The non-reserved bits of T0CSR are cleared (0) on reset.

Location: 00 FEE6$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**				**WDLTD**	**Reserved**	**TC**	**RST**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**RST (Reset).** When set (1), forces the timer to restart counting in the next input clock rising edge. The bit is cleared by the input clock rising edge, indicating that the counter resumed its automatic re-triggerable operation. Writing 0 to this bit is ignored.
1	**TC (Terminal Count).** Indicates that the counter has reached zero (terminal count). This bit is cleared each time the register is read. It is a read-only bit and data written to it is ignored.
2	**Reserved.**
3	**WDLTD (WATCHDOG Last Touch Delay).** The bit is set when the WDCNT is written. It is cleared after WATCHDOG is updated. (After WATCHDOG is updated, it is safe to switch to Idle mode.)
7-4	**Reserved.**

WATCHDOG Count Register (WDCNT)

The WDCNT register is a byte-wide, write-only register. It holds the value loaded into the WATCHDOG counter when it is touched and counts down from it. The WATCHDOG is started by the first write to the register. Each successive write restarts the WATCHDOG counter. A write to WDCNT functions as a touch operation when WDSDME bit TWCFG register is cleared, even if WDCNT is locked; in this case, the WATCHDOG counter is restarted using the value loaded in PRESET field before WDCNT was locked (i.e., the new PRESET value is ignored). On reset this register is initialized to 0F$_{16}$.

Location: 00 FEE8$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**PRESET**							
Reset	0	0	0	0	1	1	1	1

Bit	Description
7-0	**PRESET.** Defines the counter preset value.

WATCHDOG Service Data Match Register (WDSDM)

The WDSDM register is a byte-wide, write-only register. When WDSDME in TWCFG register is set, the WATCHDOG counting restarts from the value in WDCNT, when WDSDM is written with 5C$_{16}$. If any other data is written to this register, it triggers a WATCHDOG signal. If RSDATA is written more than once per three WATCHDOG clock cycles, a WATCHDOG signal is also triggered. When the WDSDME bit is cleared, a write to this register is ignored.

Location: 00 FEEA$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**RSDATA**							

Bit	Description
7-0	**RSDATA.**

4.10.4 Usage Hints

The TWD protects WATCHDOG operation from software tampering. To achieve the highest level of protection, proceed as follows:

1. Program the TWDT0 pre-scale and TMWT0 timers to the desired values.
2. Configure the WATCHDOG clock to use T0IN or T0OUT using WDCT0I bit in TWCFG register.
3. Program the WDCTL to the maximum period between WATCHDOG touch operations. Note that from this point, the WATCHDOG starts operating and must be touched periodically to prevent a WATCHDOG error signal.
4. Configure the WATCHDOG to use data match, and lock all the TWD configuration and setting registers by setting bits 0 through 4 and bit 6 of the TWCFG.
5. Touch the WATCHDOG by writing $5C_{16}$ to WDSDM at the appropriate rate (i.e., no more than once every WATCHDOG clock cycle and no less than the period programed to WDCTL).

4.11 ANALOG TO DIGITAL CONVERTER (ADC)

The Analog to Digital Converter (ADC) monitors various voltages and temperatures in the system and reports their values to the core.

The ADC can measure up to 10 external voltage inputs and four internal voltage sources, with 10-bit resolution. The internal voltage sources measure the V_{DD}, V_{CC}, AV_{CC} and V_{BAT} supplies. V_{DD}, V_{CC} and AV_{CC} are divided by 2 to allow both overvoltage and undervoltage detection. The external voltage inputs support thermistor-based temperature measurement.

The ADC can measure up to two, diode-based temperature sources with 1°C resolution. The current generated by the ADC module causes a voltage drop on a remote (external) or local (internal) diode. This voltage is measured by the ADC and converted to temperature units. An Overtemperature Shutdown ($\overline{OTS}$) interrupt becomes active when the temperature of the internal diode exceeds a programmable limit.

The ADC executes cycles of one temperature and three voltage measurements, each assigned to a separate output channel. The temperature channel measures the two temperature sources (remote and local) in consecutive cycles. Each of the three voltage channels measures one selected voltage input during the cycle.

4.11.1 Features

- Voltage measurement
  - Ten external voltage inputs and four internal power supply inputs
  - 10-bit resolution
  - 0 to 2.97V input voltage range
  - High-impedance, ground-referenced inputs
  - Enables thermistor-based temperature measurement
- Diode-based temperature measurement
  - one remote (external) diode input and one local (internal) diode
  - 1°C/bit resolution
  - –40°C to +125°C range (using the remote diode)
  - Open/short detection for the remote diode connection
  - Hardware-monitored, local overtemperature detection
- Internal high-precision reference
- Digital reading output channels
  - Three voltage buffers and one temperature buffer (diode)
  - Input selection for each voltage channel
- Sampling sequence and timing
  - Three voltage and one temperature measurement within 100 ms.
  - Cyclic measurement of the four output channels
  - Separate enable for each channel
  - Programmable conversion-start delay to guarantee input settling time
- Polling- or interrupt-driven interface
- Power consumption
  - Zero current when disabled
  - Low operating current

4.11.2 Functional Description

Inputs. The ADC has 16 inputs (AI0-13, AI15, AI16) divided in three groups:

- External Voltage (AI0-9). These are either general-purpose, positive DC voltage sources or temperature-dependent voltage generated by using a Negative/Positive Temperature Coefficient (NTC/PTC) thermistor in a resistive divider.
- Internal Voltage (AI10-13) These are internally connected to the supply voltages of the device (V_{DD}, V_{CC}, AV_{CC} and V_{BAT}). Voltages higher than the full-scale voltage (V_{FS}) are divided by 2, except for V_{BAT} which is not divided in order to minimize the current drain.
- Diode-Based Temperature Inputs (AI15, AI16). These are temperature-dependent voltage drops, on a remote and a local diode, caused by the flow of a current generated by the ADC. Since the AI8 and AI9 voltage inputs use the same package pins as the remote diode (DP, DN), only one option (voltages or diode) is available at a time.

Embedded Controller Modules (Continued)

Temperature Sensor Interface (TSI). The TSI generates the diode current and converts the voltage drop to a single-ended voltage. In addition, the TSI of the remote diode enables open/short detection for the diode connection cable and sets the corresponding status bits.

Analog Multiplexer. A 16 to 1 analog multiplexer selects one of the inputs for measurement by connecting it to the A/D converter. Switching between inputs at different voltage levels requires delaying the conversion start until the input voltage to the A/D converter has settled to the desired accuracy.

A/D Converter. The Σ–Δ, high-resolution A/D converter receives the selected input and converts it. The result of the conversion is either a 10-bit, unsigned integer digital value (0 to 1023) for voltage inputs or an 8-bit, 2s complement digital value (–128 to +127) with the Least Significant Bit (LSB) having the value of 1°C for temperature inputs.

A high-precision internal reference generator sets the full-scale voltage value (V_{FS}) of the A/D converter.

ADC Cycle. The ADC has four output “channels”: one for temperature and three for voltage measurement. The voltage measurement channels are not related to specific inputs. They hold the input select control data for the next measurement and contain a buffer in which the conversion result is stored. The temperature channel is slightly different, since it lacks the input select control data. It “toggles” between the remote and local diode inputs, unless the remote diode input is disabled.

An ADC cycle includes measurements of all four channels and a calibration operation. The first measurement is of the temperature channel, followed by a calibration measurement and the three voltage channels in ascending order, each for the specific input number contained in its control register. The ADC waits for a programmable delay between the selection of an input and the A/D conversion start that is necessary for A/D input settling to the required accuracy.

At the end of each A/D conversion, the result is stored in the corresponding buffer and a Data Valid (DATVAL) flag is set. At the end of the ADC cycle, a flag is set indicating that all the channels contain new data. If enabled, an interrupt request is sent to the core.

After reading the data at the end of the cycle, the software may choose to set up new values for the input select control data in order to measure different inputs during the following cycle. Another option is to use the old values for one or more voltage channels and repeat the measurement on the previous inputs for a higher sampling rate. Any voltage output channel may be disabled in order to skip its related measurement during the ADC cycle (shorter cycle).

Digital Comparator. A digital comparator checks the temperature measured by the local diode by comparing it with a user-programmable limit. When the temperature of the PC87591x exceeds the Overtemperature Limit, the Overtemperature Shutdown ($\overline{\text{OTS}}$) signal becomes active and an interrupt request (if enabled) is sent to the core.

ADC Control/Status Registers and Data Out Buffers. These may be read/written by the core through the Peripheral bus.

Timing Control. This block reduces the frequency of the system clock to the lower value required by the ADC.

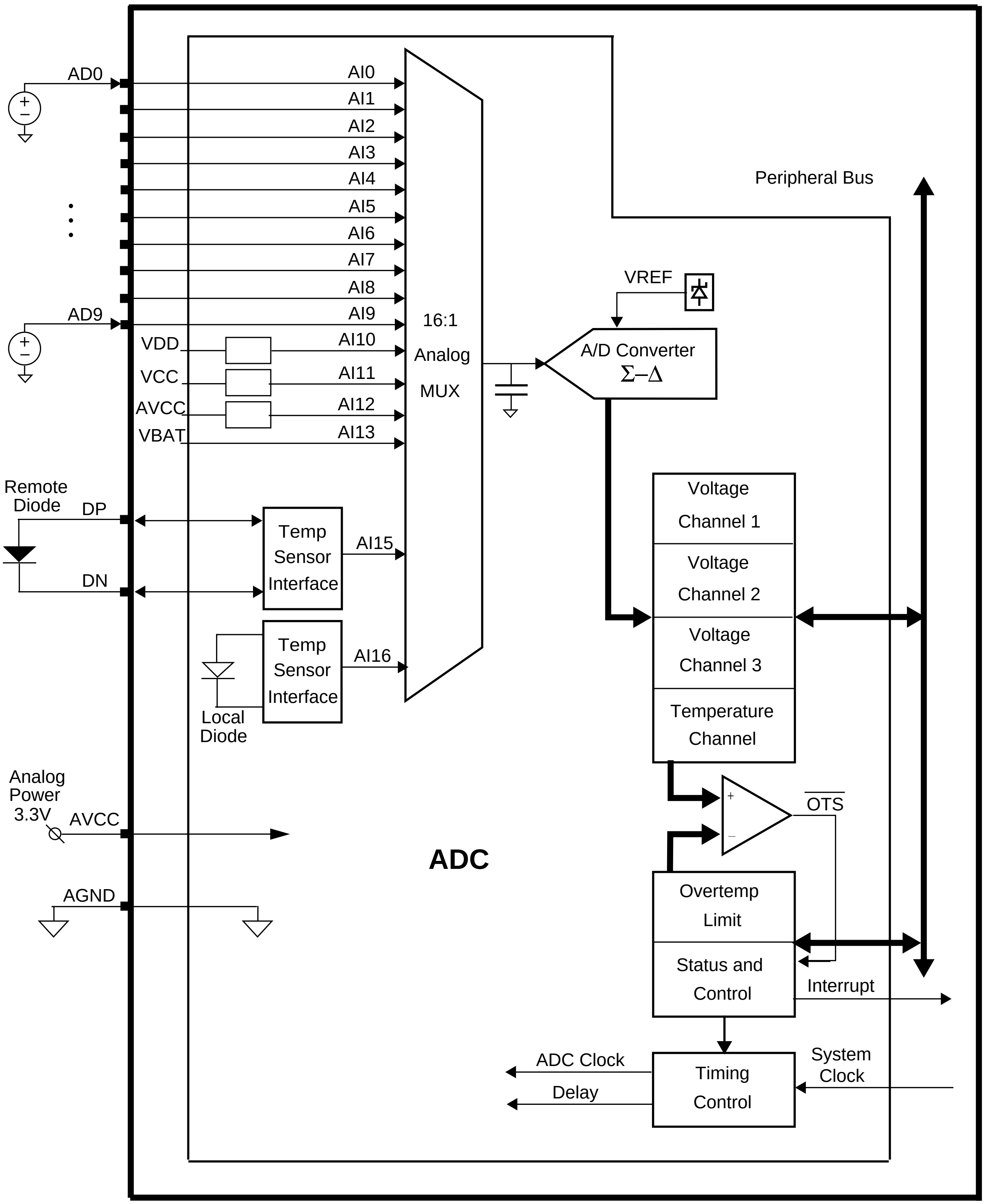


Figure 60. ADC Functional Diagram

4.11.3 Voltage Measurement

The ADC performs a linear conversion of the input voltage signal to a 10-bit, unsigned digital representation. The input signal should be applied relative to the AGND pin and should range from 0V to V_{FS} (Full Scale).

The equations in the folowing table should be used to calculate the input voltage based on the reading from the Voltage Channel Data result (VCHDAT field in VCHiDAT register).

Input Channel	Scale	Calculation[1]
AD0 to AD9	Low	Vi = VCHDAT(9-0) * (1 / 1024) * V_{FSL}
AD0 to AD9	High	Vi = VCHDAT(9-0) * (1 / 1024) * V_{FSH}
AD10 to AD13	Low[2,3]	Vi = VCHDAT(9-0) * (1 / 1024) * V_{FSV}

1. See Section 7.4.1 on page 378 for the dynamic range relevant for each input.
2. High Scale is undefined for these inputs.
3. These inputs are scaled down by 4 at the input and compensated back at the result read phase.

An input signal of zero (ground) is converted to 000_{16}; an input signal equal to (1023/1024) * V_{FS} is converted to $3FF_{16}$.

Changing the input selection for a new measurement requires switching between inputs at different voltage levels. The input interface circuits of the ADC, together with the externally added noise-rejection filters (if applicable), require a settling time to reach the new voltage value with a 10-bit accuracy (less than 1/2 LSB error). Therefore, the ADC waits for a programmable delay time between the selection of the input to be measured and the beginning of the A/D conversion. This Voltage Channel Delay is expressed in ADC clock cycles in ADC Delay Control register (ADLYCTL). The number of ADC clock cycles should be converted to time using the following formula:

$$t_{VD} = \texttt{Number\_of\_ADC\_clocks} * \texttt{(System\_clock\_cycle)} * \texttt{SCLKDIV(5-0)}$$

To calculate the required delay value according to externally added components, see Section 4.11.7 on page 185.

4.11.4 Temperature Measurement

The ADC performs a linear conversion of the temperature, measured by a remote or a local diode, to an 8-bit, 2s complement digital representation. The ADC generates a current that flows through the diode and measures the resulting voltage drop. This differential voltage is converted to an 8-bit value, with the LSB equal to 1°C, according to Table 22.

Table 22. Temperature Format Conversion

Temperature (°C)	Temperature Channel Data	
	Binary	Hex
+127[1]	0111 1111	$7F_{16}$
+125	0111 1101	$7D_{16}$
+100	0110 0100	64_{16}
+40	0010 1000	28_{16}
+25	0001 1001	19_{16}
+1	0000 0001	01_{16}
0	0000 0000	00_{16}
–1	1111 1111	FF_{16}
–25	1110 0111	$E7_{16}$
–40	1101 1000	$D8_{16}$
–128[1]	1000 0000	80_{16}

1. Values higher than +125°C and lower than –40°C are not within the specified operating temperature range.

Embedded Controller Modules (Continued)

To change the diode selection for a new measurement, the input is switched between inputs at different voltage levels. The temperature input interface circuits of the ADC require a settling time to reach the new voltage value with the equivalent of 1/2°C accuracy. Therefore, the ADC waits for a programmable delay time between the selection of the diode to be measured and the beginning of the A/D conversion. This Temperature Channel Delay is expressed in ADC clock cycles in ADC Delay Control register (ADLYCTL). The number of ADC clock cycles should be converted to time using the following formula:

$$\texttt{Delay} = \texttt{Number\_of\_ADC\_clocks} * (\texttt{System\_clock\_cycle}) * \texttt{SCLKDIV(5-0)}$$

To calculate the required delay value according to the externally added components, see Section 4.11.7 on page 185.

4.11.5 ADC Operation

Reset

Section 3.2 on page 65 describes the types of PC87591x resets. The ADC is affected by the core domain reset events, as described below:

All control, configuration and status registers are reset to their default values, as indicated in Section 4.11.6 on page 177. An exception to this is OTSEV (Overtemperature Event) bit in ADCSTS register, which is reset at V_{CC} Power-Up reset only. The Temperature and Voltage (1, 2 and 3) Channel Data Buffer registers are not reset, since their value is undefined until the first measurement occurs (on each of them).

The ADC is disabled, with all interrupt sources masked and all event status bits reset. The remote temperature diode is disabled, and the voltage inputs AD8 and AD9 may be used. The clock division factor, as well as the temperature channel delay and the voltage channel delay, are all set to their maximum value (for the slowest ADC operation speed). The Overtemperature limit is set to the highest value (127°C) for maximum flexibility. Each of the four channels is individually disabled, along with its interrupt source. The Selected Input for all three voltage channels is set to $1F_{16}$ (disabled).

ADC Clock

The ADC clock is generated by dividing the system clock by a factor in the range of 4 to 63, as defined in SCLKDIV field in ACLKCTL register (see Section 4.11.6 on page 177). The system clock's source is the on-chip clock multiplier (see Section 4.18 on page 239). The ADC clock needs to be at a frequency of 0.5 MHz. SCLKDIV must be programed prior to enabling the ADC (i.e., while ADCEN of the ADCCNF register is 0).

Initializing the ADC

The ADC must be initialized before it is enabled. The following steps need to be taken to initialize it before enabling the ADC (i.e., ADCEN bit in ADCCNF register is cleared):

- System Clock Division Factor - SCLKDIV field in ACLKCTL register.
- Temperature Channel Delay - TMPDLY field in ADLYCTL register.
- Voltage Channel Delay - VOLDLY field in ADLYCTL register.
- Select use of Remote Diode or AI8 and AI9 - REMDEN bit in ADCCNF register.
- Copy calibration information from the flash to the index-data fields using the following sequence:
  a. Read a byte from flash information block.
  b. Set index for that parameter in ADCPINX register.
  c. Write data to ADCPD register.
  d. Repeat steps a-c for all parameters, as defined in Section B.1 on page 434.
  e. Select index 00h in ADCPINX and write data of 01_{16} to ADCPD to lock the calibration information.

Enabling and Disabling the ADC

Enabling the ADC. The ADC is enabled by setting ADCEN in ADCCNF register to 1.

After the ADC is enabled, its internal circuits need an activation delay of 100 µs. This activation delay should be added to the ADC cycle until the first batch of measurements (after enabling the ADC) is available. Note that after activation, the first set of results using the large scale mode (CSCALE bit in VCHiCTL register is clear) may be wrong.

The PC87591x temperature is measured as long as the ADC is active. Other measurement operations may be enabled or disabled individually. When measurement conversions are enabled while the ADC is enabled, the measurement operations start on the following conversion cycle.

Disabling the ADC. The ADC is disabled by resetting ADCEN in ADCCNF register when one of these conditions applies:

- V_{CC} Power-Up reset
- Warm reset
- Core enters Idle mode
- The software resets the ADCEN bit.

Embedded Controller Modules (Continued)

In this state, all ADC activities are halted and ADC current consumption from the AV_{CC} is reduced. Note that re-enabling the ADC causes an activation delay.

It is recommended to disable the ADC only after the buffer registers of all four channels have been read.

Interrupt Structure

The ADC Interrupt is a level high interrupt, generated if one (or more) of the events in Table 23 becomes active. The ADC interrupt is connected to the ICU.

Table 23. ADC Interrupt Structure

Event Flag	Register Mnemonic	Mask Bit	Register Mnemonic	Description
OPSHEV	ADCSTS	INTOSEN	ADCCNF	Open/Short Event/Enable
OTSEV	ADCSTS	INTOTEN	ADCCNF	Overtemperature Event/Enable
EOCEV	ADCSTS	INTECEN	ADCCNF	End-of-Cycle Event/Enable
DATVAL	TCHANCTL	INTDVEN	TCHANCTL	Data Valid Event/Enable (Temperature Channel)
DATVAL	VCHN1CTL	INTDVEN	VCHN1CTL	Data Valid Event/Enable (Voltage Channel 1)
DATVAL	VCHN2CTL	INTDVEN	VCHN2CTL	Data Valid Event/Enable (Voltage Channel 2)
DATVAL	VCHN3CTL	INTDVEN	VCHN3CTL	Data Valid Event/Enable (Voltage Channel 3)

When an event flag and its related mask bit are set (enabled), the ADC Interrupt request is asserted. This is indicated by a high level of the ADC Interrupt signal.

The software must reset the event flag (or reset its mask bit) in order to deassert the ADC Interrupt request.

All the interrupt mask bits (interrupt disabled), the data-related event flags (EOCEV and the four DATVAL bits) and the connection-failure event flag (OPSHEV) are cleared by both reset conditions. The overtemperature event flag (OTSEV) is cleared only by V_{CC} Power-Up Reset.

The ADC Interrupt is routed to the ICU as an ADCI signal (see Section 4.3 on page 98).

ADC Operating Principles

Measurement Sequence. The following measurements are executed, in the order listed, during one ADC cycle for all enabled channels:

1. Temperature measurement, from the input indicated by SELIN field in TCHANCTL register. The A/D conversion starts by selecting the input and waiting for the time period (TMPDLY delay) set in ADLYCTL register. The resulting 8-bit digital value is stored in TCHANDAT register, and DATVAL bit in TCHANCTL register is set.
2. Voltage measurement for Voltage Channel 1, from the input selected by SELIN field in VCHN1CTL register. The A/D conversion starts by selecting the input and waiting for the time period (VOLDLY delay) set in ADLYCTL register. The resulting 10-bit digital value is stored in VCHN1DAT register, and DATVAL bit in VCHN1CTL register is set. Note: This measurement is skipped if Voltage Channel 1 is disabled by setting the SELIN field in VCHN1CTL register to $1F_{16}$.
3. Voltage measurement for Voltage Channel 2, as above, using the VCHN2CTL and VCHN2DAT registers.
4. Voltage measurement for Voltage Channel 3, as above, using the VCHN3CTL and VCHN3DAT registers.
5. End of the ADC cycle. EOCEV bit in ADCSTS register is set (in addition to all the relevant DATVAL bits).

The software may read the measurement result for each channel immediately after its DATVAL bit is set. Alternatively, the results may be read at the end of the cycle when EOCEV bit is set. After the data in VCHNiDAT register is read, the software should clear the relevant DATVAL bit to indicate that the data in the buffers has been read.

The ADC cycle duration may be calculated using the formula below (N is the number of enabled voltage channels):

$$T_{ADC\ cycle} = 2 * (t_{TD} + t_{TC}) + (N+1) * (t_{VD} + t_{VC})$$

Where:

t_{TD} - Temperature Conversion Delay Time
t_{TC} - Temperature Conversion Time
t_{VD} - Voltage Conversion Delay Time
t_{VC} - Voltage Conversion time

See Section 7.4 on page 378 for the values of t_{TC}, Section 4.11.4 on page 173 for t_{TD} calculation and Section Section 4.11.3 on page 173 for t_{VC} calculation.

Temperature inputs from remote and local diodes are measured by alternating the remote and local diodes in each successive ADC cycle. If the remote diode measurement is disabled (RDMEASEN is set to 0 in TCHANCTL register or REMDEN in ADCCNF register is set to 0), the local diode temperature is measured every ADC cycle.

Input Selection Field. Each Voltage Channel has its own programmable, input selection field (SELIN in VCHNxCTL register). This field determines which input is measured by the channel during the current ADC cycle. The field also indicates to which input the data in the channel buffer belongs. This field may be modified after the channel buffer has been read and the DATVAL bit has been reset.

If the input selection field is not changed, the same input is measured during the next ADC cycle. This gives a sampling rate of one $T_{ADC\ cycle}$ for the specific input. If this field is changed and a different input is sampled, the sampling rate is lower for each input, but the period for all scanned inputs is shorter.

Operation Sequence. After the ADC is properly initialized and enabled, one of the following example sequences can be used:

- EOCEV-driven ADC operation sequence for all temperature and voltage channels
- DATVAL-driven ADC operation sequence for one temperature or voltage channel

EOCEV-Driven ADC Operation Sequence for All Channels

1. When End-of-Cycle is reached (i.e., after all enabled channel conversions are completed), software can detect the event by waiting for EOCEV bit in ADCSTS register to be set to 1.
2. Read the input number of the temperature channel by reading SELIN in TCHANCTL register.
3. Read the temperature value measured by reading TCHDAT in TCHANDAT register.
4. In preparation for the next measurement, clear DATVAL bit in TCHANCTL register by writing 1 to it.
5. Measure the input number for Voltage Channel 1 by reading SELIN in VCHN1CTL register.
6. Read the number of input measured in Voltage Channel 1 by reading VCHDAT in VCHN1DAT register.
7. In preparation for the next measurement (i.e., to define which input will be measured by Voltage Channel 1 during the next ADC cycle), clear DATVAL bit in VCHN1CTL register by writing 1 to it (it may be the same input or, optionally, a different one).
8. For Voltage Channel 2, repeat steps 5 through 7 for the VCHN2CTL and VCHN2DAT registers.
9. For Voltage Channel 3, repeat steps 5 through 7 for the VCHN3CTL and VCHN3DAT registers.

DATVAL-Driven ADC Operation Sequence for One Channel

1. Measure the end of channel by waiting for DATVAL in TCHANCTL or VCHNxCTL register to be set to 1.
2. Measure the input number by reading SELIN in TCHANCTL or VCHNxCTL register.
3. Read the measured data by reading TCHDAT or VCHDAT in TCHANDAT or VCHNxDAT register, respectively.
4. Optional (for voltage channels only): Change the input to be measured during the next ADC cycle: in VCHNxCTL register, write a new SELIN value.
5. Prepare the temperature or voltage channel to receive new data: in TCHANCTL or VCHNxCTL register, write 1 to DATVAL to clear it.
6. In preparation for the next measurement (i.e., to define which input will be measured by the temperature channel or the voltage channel during the next ADC cycle), clear the DATVAL bit by writing 1 to it (it may be the same input or, optionally, a different one).

Reading Measurement Results

Polling-Driven Operation. Measurement results may be read by polling either EOCEV in ADCSTS register or each of the four DATVAL bits in TCHANCTL register and three VCHNxCTL registers.

Polling EOCEV uses the sequence listed in “EOCEV-Driven ADC Operation Sequence for All Channels”, above. When EOCEV is set, all four channels contain valid data and may be read.

Polling DATVAL uses the sequence listed in “DATVAL-Driven ADC Operation Sequence for One Channel”, above. When a DATVAL bit is set, only its channel contains valid data that may be read. In this case, the EOCEV bit is redundant.

Interrupt-Driven Operation. The ADC may generate an interrupt to the core when any of the valid bits is set (EOCEV in ADCSTS register or DATVAL in either TCHANCTL or any VCHNxCTL register). The interrupt is generated when the interrupt enable bit for the respective status bit is set. The software in the interrupt routine should check the status bits as described above for polling-driven operation to verify which of the DATVAL bits is set.

Embedded Controller Modules (Continued)

An interrupt is expected from EOCEV when using the sequence listed in the EOCEV-driven ADC operation sequence (see above). The EOCEV interrupt indicates that all four channels contain valid data and may be read. Interrupts from the DATVAL bits should be disabled.

An interrupt is expected from one of the DATVAL bits when using the sequence listed in the DATVAL-driven ADC operation sequence (see above). The DATVAL interrupt indicates that only its channel contains valid data that may be read. Interrupts from the EOCEV bit should be disabled.

Failure Detection

Open/Short. At the beginning of the temperature measurement from the remote diode, the Temperature Sensor Interface checks the condition of the interconnection (shorted or disconnected cable) and updates OPSHTYPE and OPSHEV in ADCSTS register (see Section 4.11.6 on page 177). Although the measurement is executed and the DATVAL bit in TCHANCTL register is set, the contents of the data buffer (TCHANDAT) should be ignored.

Overtemperature. At the end of the temperature measurement from the local diode (after DATVAL bit in TCHANCTL register is set), the ADC compares the measured temperature with the limit value set by OTSLIM in TLOCOTL register. Whenever the measured temperature is higher than the limit value, OTSEV in ADCSTS register is set and an active low $\overline{\text{OTS}}$ signal is issued.

Overflow. An overflow occurs when DATVAL bit in TCHANCTL register is set at the end of a measurement, indicating that the result of the previous measurement was not read. If an overflow occurs in at least one channel, the new measurement result overrides the old data in the buffer, and OVFEV in ADCSTS register is set. This indicates that the result of the previous measurement was lost.

4.11.6 ADC Registers

The ADC control/status and data out registers set interfaces with the core through the Peripheral bus. These registers are mapped in the address space of the core, starting at the base address defined in Section B.1 on page 434.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

ADC Register Map

The ADC register set contains seven common, control and status registers and eight channel-specific registers.

Table 24. ADC Register Map

Mnemonic	Register Name	Type
ADCSTS	ADC Status	Varies per bit
ADCCNF	ADC Configuration	R/W
ACLKCTL	ADC Clock Control	R/W
ADLYCTL	ADC Delay Control	R/W
TLOCOTL	Local Diode Overtemperature Limit	R/W
ADCPINX	ADC Parameters Index	R/W
ADCPD	ADC Parameters Data	R/W
TCHANCTL	Temperature Channel Control	Varies per bit
TCHANDAT	Temperature Channel Data Buffer	RO
VCHN1CTL	Voltage Channel 1 Control	Varies per bit
VCHN1DAT	Voltage Channel 1 Data Buffer	RO
VCHN2CTL	Voltage Channel 2 Control	Varies per bit
VCHN2DAT	Voltage Channel 2 Data Buffer	RO
VCHN3CTL	Voltage Channel 3 Control	Varies per bit
VCHN3DAT	Voltage Channel 3 Data Buffer	RO

ADC Status Register (ADCSTS)

This register indicates the global status of the ADC module. ADCSTS is cleared (00_{16}) on V_{CC} Power-Up reset; on other resets, bit 2 is unchanged and other bits are cleared.

Location: 00 $FF20_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**OPSHTYPE**	**OPSHEV**	**OTSEV**	**OVFEV**	**EOCEV**
Warm Reset	0	0	0	0	0	-	0	0
Power-Up Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	RO	**EOCEV (End-of-Cycle Event).** End of ADC cycle; all enabled measurements (up to four) are completed. For each of the enabled channels, the DATVAL bit in the respective TCHANCTL VCHNxCTL register is set, and the data for the channel is stored in the respective Channel Data Buffer register. 0: Cycle in progress (default) 1: End of ADC cycle (the bit remains set until all DATVAL bits in the Channel Control registers are reset)
1	R/W1C	**OVFEV (Data Overflow Event).** Measurement data from the previous cycle was overwritten with data from the current cycle before being read. In the event of a data overflow, the DATVAL bit remains set and new data is placed in Channel Data Buffer register. 0: No overflow (default) 1: Overflow
2	R/W1C	**OTSEV (Overtemperature Signal Event).** Temperature measured by local diode exceeded the limit. 0: Temperature below or equal to the limit (V_{CC} Power-up reset default) 1: Temperature above limit (once set, this bit is cleared by writing 1 to it; writing 0 is ignored)
3	R/W1C	**OPSHEV (Open/Short Event).** Remote diode connection failure (cable shorted or disconnected). Checked before the beginning of the remote diode temperature measurement (if enabled, REMDEN in ADCCNF register and RDMEASEN in TCHANCTL register are set to 1). 0: No fault was detected (default) 1: Connection failure - the cable was shorted or disconnected (once set, this bit is cleared by writing 1 to it; writing 0 is ignored)
4	RO	**OPSHTYPE (Open or Short Type).** Remote diode connection failure type. Indicates the type of error flagged by OPSHEV. 0: DP shorted to DN or to AGND (default) 1: DP disconnected (floating) or shorted to AV_{CC}
7-5		**Reserved.**

Embedded Controller Modules (Continued)

ADC Configuration Register (ADCCNF)

This register controls the operation and global configuration of the ADC module. ADCCNF is cleared (00_{16}) on reset.

Location: 00 $FF22_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**INTOSEN**	**INTOTEN**	**INTECEN**	**REMDEN**	**ADCEN**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**ADCEN (ADC Module Enable).** Controls the operation of the ADC to minimize power consumption and prevent glitch effects during configuration changes; see “Enabling and Disabling the ADC” on page 174. 0: ADC disabled (default) 1: ADC enabled
1	**REMDEN (Remote Temperature Diode Enable).** Controls the utilization of the package pins for either the remote diode (DP, DN) or the AI8 and AI9 voltage inputs (AD8, AD9). 0: AI8, AI9 voltage inputs enabled (default) - forces RDMEASEN in TCHANCTL register to 0; only the local diode is measured by the temperature channel. 1: Remote diode enabled - if RDMEASEN in TCHANCTL register is set, the temperature channel measures both the remote and the local diodes in successive cycles.
2	**INTECEN (Interrupt from End-of-Cycle Event Enable).** Enables generation of an ADC interrupt on an End-of ADC-cycle event (EOCEV in ADCSTS register). 0: Disabled (default) 1: Enabled - ADC Interrupt from EOCEV
3	**INTOTEN (Interrupt from Overtemperature Event Enable).** Enables generation of an ADC interrupt on an Overtemperature event (OTSEV in ADCSTS register). 0: Disabled (default) 1: Enabled - ADC Interrupt from OTSEV
4	**INTOSEN (Interrupt from Open/Short Event Enable).** Enables generation of an ADC interrupt on an Open/Short event (OPSHEV in ADCSTS register). 0: Disabled (default) 1: Enabled - ADC Interrupt from OPSHEV
7-5	**Reserved.**

ADC Clock Control Register (ACLKCTL)

This register controls the system clock to ADC clock division. ACLKCTL is set to $3F_{16}$ on reset.

Location: 00 $FF24_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**SCLKDIV**					
Reset	0	0	1	1	1	1	1	1

Bit	Description
5-0	**SCLKDIV (System Clock Division Factor).** Used to divide the system clock in order to obtain the ADC clock. The system clock frequency is set separately (see Figure 78 on page 239). The resulting ADC clock frequency should be equal to 0.5 MHz. Range: 4 to 63 (default is 63, decimal values); values 0 to 3 are illegal and may result in undetermined ADC behavior.
7-6	**Reserved.**

Embedded Controller Modules (Continued)

ADC Delay Control Register (ADLYCTL)

This register controls the delay between “input switching” and “conversion start” for the voltage and temperature channels. ADLYCTL is set to $A7_{16}$ on reset.

Location: 00 $FF26_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**TMPDLY**				**Reserved**	**VOLDLY**		
Reset	1	0	1	0	0	1	1	1

Bit	Description
2-0	**VOLDLY (Voltage Channel Delay).** Compensates for the settling time of the input interface and externally added filter (if used). The conversion of the VOLDLY value to delay, in terms of ADC clock cycles, is shown below. To calculate the required value and the resulting delay time, see Section 4.11.3 on page 173 and “Calculating the Voltage Channel Delay” on page 186. **VOLDLY Value** **Bits** **2 1 0: Voltage Channel Delay (ADC Clock Cycles)** 0 0 1: 4 0 1 0: 8 0 1 1: 16 1 0 0: 32 1 0 1: 64 1 1 0: 128 1 1 1: 256 (default) Other: Reserved
3	**Reserved**.
7-4	**TMPDLY (Temperature Channel Delay).** Compensates for the settling time of the input interface and externally added filter (if used). The conversion of the TMPDLY value to delay, in terms of ADC clock cycles, is shown below. To calculate the required value and the resulting delay time, see Section 4.11.4 on page 173 and “Calculating the Temperature Channel Delay” on page 188. **TMPDLY Value** **Bits** **7 6 5 4 Temperature Channel Delay (ADC Clock Cycles)** 0 0 0 1: 32 0 0 1 0: 64 0 0 1 1: 128 0 1 0 0: 256 0 1 0 1: 512 0 1 1 0: 1024 0 1 1 1: 2048 1 0 0 0: 4096 1 0 0 1: 8192 1 0 1 0: 16384 (default) Other: Reserved

Local Diode Overtemperature Limit Register (TLOCOTL)

This register holds the limit value used for overtemperature detection. TLOCOTL is set to $7F_{16}$ on reset.

Location: 00 $FF28_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	OTSLIM							
Reset	0	1	1	1	1	1	1	1

Bit	Description
7-0	**OTSLIM (Overtemperature Limit Value).** Threshold value for Overtemperature Event detection. Whenever the temperature measured by the local (internal) diode exceeds this limit, the OTSEV bit is set, an interrupt is sent to the core (if INTOTEN in ADCCNF register is set to 1) and the $\overline{\text{OTS}}$ signal becomes active. To calculate the required value for a specific temperature limit, see Section 4.11.4 on page 173. Range: –128 to +127 (default is +127; these are decimal values); 8-bit, 2s complement value with 1 LSB = 1^{o}C.

ADC Parameters Index Register (ADCPINX)

This register holds an index to the ADC parameters registers. Use it while initializing the ADC (see “Initializing the ADC” on page 174 for more details). ADCPINX is cleared (00_{16}) on reset.

Location: 00 $FF2A_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	Index							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Index.** Defines which parameter register is being accessed by the ADCPD register.

ADC Parameters Data Register (ADCPD)

This register enables access to the ADC parameters registers. Use it while initializing the ADC (see “Initializing the ADC” on page 174 for more details).

Location: 00 $FF2C_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Parameter Data															

Bit	Description
15-0	**Parameter Data.** The register is used to access the ADC parameters register that the Index register (ADCPINX) points to. Writing 01_{16} to index 00_{16} locks the parameter data registers from any further writes until the next reset.

Embedded Controller Modules (Continued)

Temperature Channel Control Register (TCHANCTL)

This register both controls the operation and indicates the status of the Temperature channel. TCHANCTL is set to 10_{16} on reset.

Location: 00 $FF30_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**DATVAL**	**RDMEASEN**	**INTDVEN**	**SELIN**				
Reset	0	0	0	1	0	0	0	0

Bit	Type	Description
4-0	RO	**SELIN (Selected Input).** Indicates which temperature input was selected for measurement. If RDMEASEN is set to 0, only the value 16 dec is read. **Bits** **4 3 2 1 0 Description** 0 1 1 1 1: Remote diode (DP, DN pins) selected 1 0 0 0 0: Local diode (internal) selected (default)
5	R/W	**INTDVEN (Interrupt from Data Valid Enable).** Temperature channel, Data Valid event (End-of-conversion) enabled to the ADC Interrupt. 0: Disabled (default) 1: Enabled - ADC Interrupt from local DATVAL
6	R/W	**RDMEASEN (Remote Diode Measurement Enable).** Enables the temperature measurement using the remote diode (REMDEN in ADCCNF register is set to 1). Using only the local diode enables a higher measurement rate for the internal temperature. The temperature channel cannot be disabled (“skipped”) since the internal temperature measurement is required. 0: Measurement disabled - only the local diode is measured by the temperature channel; forced into this state when REMDEN in ADCCNF register is set to 0 (default). 1: Measurement enabled - the temperature channel measures both the remote and the local diodes in successive cycles; possible only if REMDEN is set to 1.
7	R/W1C	**DATVAL (Data Valid).** The Temperature Channel Data Buffer contains new data. The data may be read immediately or at the end-of-cycle. This flag is cleared when the ADC module is disabled (ADCEN in ADCCNF register is cleared) or by writing 1 to it. 0: No valid data in TCHANDAT register (default) 1: End of conversion - new data is available in TCHANDAT

Temperature Channel Data Buffer (TCHANDAT)

This register (buffer) holds the data measured by the Temperature Channel.

Location: 00 $FF32_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**TCHDAT**							

Bit	Description
7-0	**TCHDAT (Temperature Channel Data).** Temperature of either the remote or local diode, measured by the Temperature Channel. For the temperature word format, see Section 4.11.4 on page 173. The data is valid and may be read only at the end of the conversion, when DATVAL in TCHANCTL register is set to 1. DATVAL must be reset after reading the data. Range: –128 to +127 (values are decimal); 8-bit, 2s complement value with 1 LSB = 1^{o}C (the value at reset is undefined).

Embedded Controller Modules (Continued)

Voltage Channel 1 Control Register (VCHN1CTL)

This register both controls the operation and indicates the status of Voltage Channel 1. VCHN1CTL is set to $1F_{16}$ on reset.

Location: 00 $FF34_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**DATVAL**	**CSCALE**	**INTDVEN**	**SELIN**				
Reset	0	0	0	1	1	1	1	1

Bit	Type	Description
4-0	R/W	**SELIN (Selected Input).** Selects a voltage input to be measured during the next ADC cycle. The new value should be set before the beginning of the channel measurement. When read, this field indicates to which input the contents of the channel data buffer belongs. When written, it selects a new input for the next channel measurement. See Figure 60 on page 172 for details on the channel input signals connection to pins. The channel is disabled by setting this field to 11111_2. When disabled, the channel is "skipped", enabling a higher measurement rate for the remaining channels (shorter ADC cycle). The DATVAL bit is cleared **Bits** **4 3 2 1 0** **Description** 0 0 0 0 0: Channel 0 0 0 0 0 1: Channel 1 .... 0 1 0 0 0: Channel 8 (Available when REMDEN bit in ADCCNF register is cleared) 0 1 0 0 1: Channel 9 (Available when REMDEN bit in ADCCNF register is cleared) 0 1 0 1 0: Channel 10 .... 0 1 1 0 1: Channel 13 Other: Reserved 1 1 1 1 1: Channel Disabled (default)
5	R/W	**INTDVEN (Interrupt from Data Valid Enable).** Enables generation of an ADC interrupt on a Voltage Channel 1, Data Valid event (End-of-conversion). 0: Disabled (default) 1: Enabled - ADC Interrupt from local DATVAL
6	R/W	**CSCALE (Channel Scale).** Controls the input scale of the input for the channel to be converted (as selected by SELIN field). 0: Channel uses the high range scale (default) 1: Channel uses the low range scale
7	R/W1C	**DATVAL (Data Valid).** Voltage Channel 1 Data Buffer contains new data. The data may be read immediately or at the end-of-cycle. This flag is cleared when the channel is disabled, when the ADC module is disabled (ADCEN in ADCCNF register is cleared) or by a write of 1 to it. 0: No new valid data in VCHN1DAT register (default) 1: End of conversion - new data is available in the buffer

Embedded Controller Modules (Continued)

Voltage Channel 1 Data Buffer (VCHN1DAT)

This register (buffer) holds the data measured by Voltage Channel 1.

Location: 00 $FF36_{16}$

Type: RO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**						**VCHDAT**									

Bit	Description
9-0	**VCHDAT (Voltage Channel 1 Data).** Selected input voltage data, measured by Voltage Channel 1. To calculate the voltage, see Section 4.11.3 on page 173. VCHN1DAT holds valid result when DATVAL bit in VCHN1CTL register is set. To prepare for consecutive result, clear DAVAL after reading the data from VCHDAT. Range: 0 to 1023 (0 to V_{FS}); 10-bit, unsigned value with 1 LSB = V_{FS}/1024 (the value at reset is undefined).
15-10	**Reserved.**

Voltage Channel 2 Control Register (VCHN2CTL)

This register controls the operation and indicates the status of Voltage Channel 2. VCHN2CTL is set to $1F_{16}$ on reset.

Location: 00 $FF38_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**DATVAL**	**CSCALE**	**INTDVEN**	**SELIN**				
Reset	0	0	0	1	1	1	1	1

Bit	Type	Description
4-0	R/W	**SELIN (Selected Input).** Same as Voltage Channel 1.
5	R/W	**INTDVEN (Interrupt from Data Valid Enable).** Same as Voltage Channel 1.
6	R/W	**CSCALE (Channel Scale).** Same as Voltage Channel 1.
7	R/W1C	**DATVAL (Data Valid).** Same as Voltage Channel 1.

Voltage Channel 2 Data Buffer (VCHN2DAT)

This register (buffer) holds the data measured by the Voltage Channel 2.

Location: 00 $FF3A_{16}$

Type: RO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**						**VCHDAT**									

Bit	Description
9-0	**VCHDAT (Voltage Channel 2 Data).** Same as Voltage Channel 1.
15-10	**Reserved.**

Embedded Controller Modules (Continued)

Voltage Channel 3 Control Register (VCHN3CTL)

This register controls the operation and indicates the status of Voltage Channel 3. VCHN3CTL is set to $1F_{16}$ on reset.

Location: 00 $FF3C_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**DATVAL**	**CSCALE**	**INTDVEN**	**SELIN**				
Reset	0	0	0	1	1	1	1	1

Bit	Type	Description
4-0	R/W	**SELIN (Selected Input).** Same as Voltage Channel 1.
5	R/W	**INTDVEN (Interrupt from Data Valid Enable).** Same as Voltage Channel 1.
6	R/W	**CSCALE (Channel Scale).** Same as Voltage Channel 1.
7	R/W1C	**DATVAL (Data Valid).** Same as Voltage Channel 1.

Voltage Channel 3 Data Buffer (VCHN3DAT)

This register (buffer) holds the data measured by Voltage Channel 3.

Location: 00 $FF3E_{16}$

Type: RO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**						**VCHDAT**									

Bit	Description
9-0	**VCHDAT (Voltage Channel 3 Data).** Same as Voltage Channel 1.
15-10	**Reserved.**

4.11.7 Usage Hints

Power Supply and Layout Guidelines

For more information, see Section 3.1.3 on page 63.

Power Consumption

ADC power consumption from the analog supply (AV_{CC}) is practically zero if the ADC is disabled by setting ADCEN in ADCCNF register to 0.

When the ADC is enabled, the current consumption depends on the channel type measured (voltage or temperature) and the ADC clock frequency. To minimize current consumption, disable the ADC when not in use. See details in “Enabling and Disabling the ADC” on page 174.

Back-Drive Protection

To maintain the high performance of the analog circuits, AD(0-7), AD8/DP and AD9/DN pins are not back-drive protected. Therefore, the voltage applied to these pins must be within the AGND to AV_{CC} range; otherwise, the device may be damaged. External circuits should not drive currents into these pins when the PC87591x is not powered up.

Measuring Out of Range Voltages

The ADC is capable of measuring positive input voltages from 0V to V_{FS}. Input voltages outside this range should either be divided or level-shifted, as required.

Embedded Controller Modules (Continued)

For positive input voltages higher than V_{FS}, a resistor divider should be used in front of the analog input (see Figure 61a). The divider should be calculated so that its output is lower than the full-scale value (V_{FS}) for the maximum input signal voltage.

For negative input voltages, a resistive level-shifter should be used in front of the analog input (see Figure 61b). The level-shifter should be calculated so that its output is higher than 0V for the minimum input signal voltage.

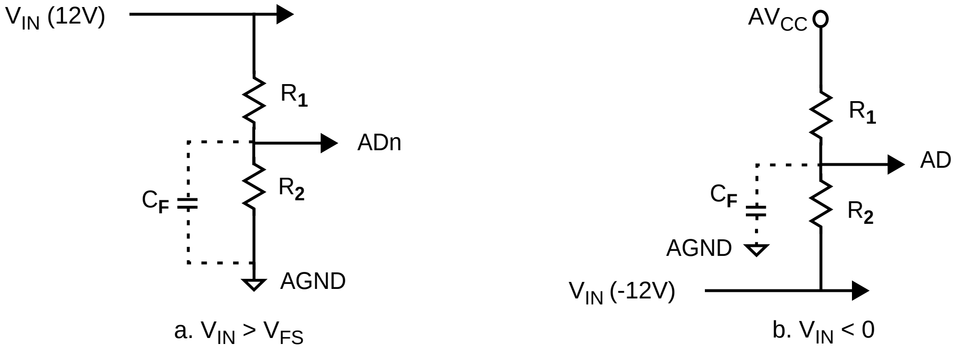


Figure 61. Measurement of Positive and Negative Voltages

Filtering the Noise on Voltage Input Signals

Noise may be coupled to the input signal for various reasons, including close proximity to digital circuits. The slow change rate of the input signals, makes it possible to use an external low pass filter (LPF). This can be implemented by placing a capacitor (C_F) between the divider output and AGND, as shown in Figure 61. The cutoff frequency of this LPF should be at least 22 times the maximum signal frequency required to be measured with a 10-bit accuracy (a smaller capacitor may be used when a lower accuracy is acceptable). The formula below demonstrates the calculation of the components.

$$f_{(-3dB)} = 1 / (2 * \pi * R_{eq} * C_F)$$

where: $R_{eq} = (R_1 * R_2) / (R_1 + R_2)$

Calculating the Voltage Channel Delay

The Voltage measurement delay time (t_{VD}) is the period between input selection and the A/D conversion start. This delay should be long enough to guarantee the voltage settling at the input of the A/D converter to within 1/2 LSB. This includes the input interface circuits of the ADC and the externally added noise-rejection filters (if applicable). Figure 62 shows the equivalent R-C circuit of a filtered input. R_{eq} represents the equivalent resistance of the input divider or level-shifter (see the preceding section, “Filtering the Noise on Voltage Input Signals”). C_F is the filter capacitor. The Input Interface Circuit includes the input the multiplexer.

Since the R-C of the input circuit is short, typically, all that is required is a very short delay, which shortens the overall measurement time. The delay length may be sized during development by increasing the delay time from the minimum value to the point were the ADC readout is not affected by the delay value (i.e., use the minimal delay to which the ADC readout is invariant).

.

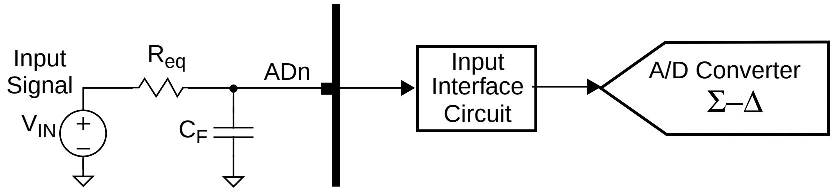


Figure 62. Filtered Voltage Input Equivalent Circuit

Thermistor-Based Temperature Measurement

The ADC is capable of measuring temperature by means of NTC or PTC thermistor devices. These elements change their resistance according to their temperature. The resistance change is converted to voltage by connecting the element in a voltage divider between AV_{CC} and AGND, as shown in Figure 63. The translation of voltage to temperature is determined by the parameters of the NTC/PTC thermistor in use.

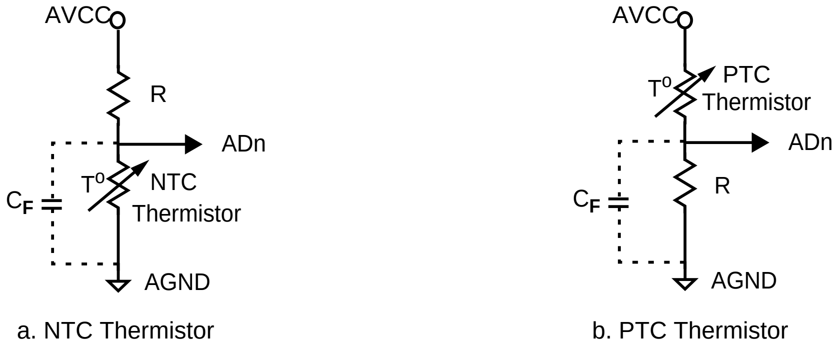


Figure 63. Measuring Temperature Using Thermistors

Remote Diode Selection

Temperature accuracy depends on a good quality transistor used as a diode-connected, small-signal transistor. Accuracy has been experimentally verified for the devices, among them Motorola, listed in Table 25. A temperature sensor can directly measure the die temperature of CPUs with on-board temperature-sensing diodes.

The transistor must be a small-signal type with a relatively high forward voltage; otherwise, the A/D input voltage range might be violated. The forward voltage must be greater than 0.25V at 10 µA for the highest expected temperature. The forward voltage must be less than 0.95V at 100 µA for the lowest expected temperature. Large power transistors do not work. Also, ensure that the base resistance is less than 100Ω. Tight specifications for forward-current gain (+50 to +150, for example) indicate devices with consistent VBE characteristics.

Thermal mass can seriously degrade the temperature sensor's effective accuracy. The use of smaller packages for remote sensors, such as SOT23s, improves the situation.

Table 25. Remote Sensor Transistor Manufacturers

Manufacturer	Model
Central Semiconductor (USA)	CMPT3904
Motorola (USA)	MMBT3904
National Semiconductor (USA)	MMBT3904
Rohm Semiconductor (Japan)	SST3904
Samsung (Korea)	KST3904-TF
Siemens (Germany)	SMBT3904
Zetex (England)	FMMT3904CT-ND

Filtering the Noise on Temperature Input Signals

Noise may be coupled to the input signal for various reasons, including close proximity to digital circuits. The ADC conversion technology has good noise rejection especially for low-frequency signals, such as power supply hum. Rejection of high-frequency noise can be improved using an external filter capacitor and PC board layout.

High-frequency noise is best filtered by placing a capacitor (C_C of 2.2 nF) between pins DP and DN (see Figure 64), in proximity to the PC87591x pins. The connection to the diode should use a twisted wire pair or a PCB that is read out using the guidelines below.

It is most important to prevent any leak to ground, especially from the DP pin and serial resistance on the current loop to the diode.

Embedded Controller Modules (Continued)

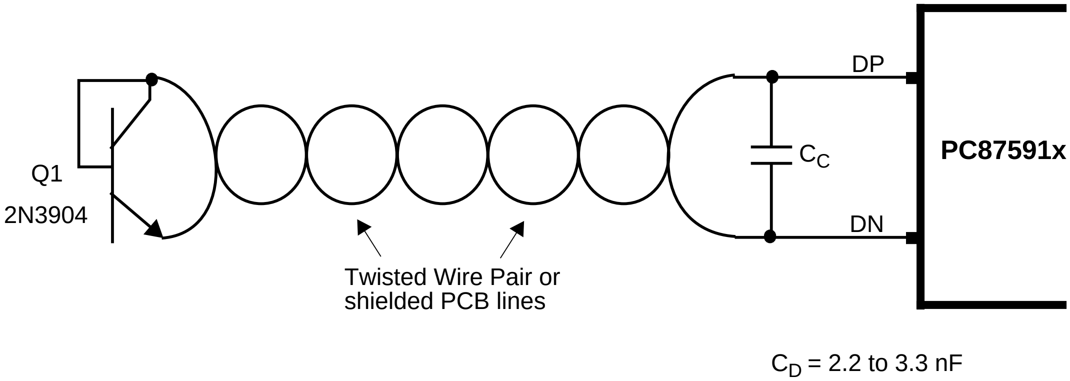


Figure 64. Typical Remote Diode Circuit

Calculating the Temperature Channel Delay

The delay time is the period between diode selection and the A/D conversion start. Due to the switched diode current, this delay should be long enough to guarantee the voltage settling at the input of the A/D converter to within 0.5^{o}C. This includes the input interface circuits of the ADC and the externally added capacitors (C_C).

It is preferable to use the minimal delay possible for the system. This depends on the capacitance on the wires and serial resistance. TMPDLY field of ADLYCTL register defines the delay. Typically, a delay of 128 or 256 clocks (values 3 and 4, respectively) is adequate. To find the best value, start from a high delay and reduce it until the point that the temperature readouts starts to change. This parameter is mostly board layout and component dependent.

PC Board Layout

- Place the PC87591x as close as practical to the remote diode. In a noisy environment, such as a computer motherboard, this distance can be 4 in. to 8 in. (typical) or more, as long as the worst noise sources (such as CRTs, clock generators, memory buses and ISA/PCI buses) are avoided.
- Do not route the DP and DN lines next to high inductance signals. Also, do not route the traces across a fast memory bus, which can easily introduce +30˚C error, even with good filtering. Otherwise, most noise sources are fairly benign.
- Route the DP and DN traces in parallel and in close proximity to each other, away from any high voltage traces such as +12V DC. Beware of leakage currents from PC board contamination; e.g., a 20 MΩ leakage path from DP to ground causes about +1˚C error.
- Connect guard traces to AGND on either side of the DP and DN traces (Figure 65). An additional AGND plane should be placed “beneath” these traces. With guard traces and AGND plane in place, routing near high-voltage traces is no longer a problem.
- Route through as few vias and crossunders as possible to minimize copper/solder thermocouple effects.
- When introducing a thermocouple, make sure that both the DP and DN paths have matching thermocouples.
  In general, PC board-induced thermocouples are not a serious problem. A copper-solder thermocouple exhibits 3 µV/˚C, and it takes about 200 µV of voltage error at DP and DN to cause a 1˚C measurement error. Therefore, most parasitic thermocouple errors can be ignored.
- Use wide traces. Narrow ones are more inductive and tend to pick up radiated noise. The 10 mil widths and spacings recommended in Figure 65 are not absolutely necessary (as they offer only a minor improvement in leakage and noise), but try to use them where practical.

Embedded Controller Modules (Continued)

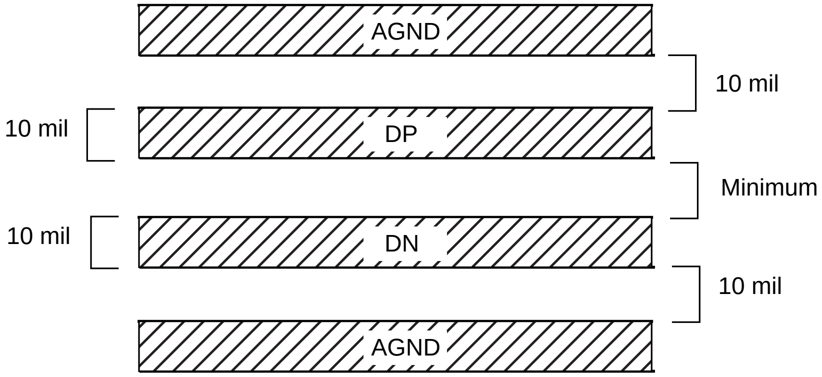


Figure 65. DP and DN, PCB Traces

Twisted Pair and Shielded Cables

For remote sensor distances longer than 8 inches or in particularly noisy environments, a twisted pair cable is recommended. Its practical length is 6 feet to 12 feet (typical) before noise becomes a problem, as tested in a noisy electronics laboratory. For longer distances, the best solution is a shielded twisted pair such as that used for audio microphones. Connect the twisted pair to DP and DN and the shield to AGND close to the PC87591x, and leave the remote end of the shield unterminated.

For very long cable runs, the cable parasitic capacitance adds a delay to the Temperature Channel. You should add the cable capacitance to the value of the C_D (see “Calculating the Temperature Channel Delay” on page 188) and correct the delay setting accordingly.

Cable resistance also affects remote sensor accuracy. A 1Ω series resistance introduces about +1.0˚C to +0.5˚C error.

4.12 DIGITAL TO ANALOG CONVERTER (DAC)

The DAC converts digital input values to analog signals. The DAC support four channels for handling up to four independent conversions in parallel.

4.12.1 Features

- 8-bit resolution
- Independent 4-channel D/A converter
- Fast settling time, 1 µs typical, on 50 pF capacitive load
- Output range from AGND to AV_{CC}
- Independent enable/disable for each channel
- All converters can be automatically disabled in Idle mode
- Low power consumption when enabled; zero power consumption when disabled
- Outputs drive 0V when disabled

4.12.2 Functional Description

The DAC has four independent digital-to-analog converters. Each converter generates an output in the range of 0V to AV_{CC}, with 8-bit resolution. The converters drive the four output pins DA0-3, as shown in Figure 67. An output impedance of 3 KΩ allows a settling time of about 1 µs on a 50 pF load.

When a DAC channel is enabled, its output is defined by the value written to its DACDATn register. DACDAT0-3 control DA0-3, respectively. The maximum output voltage is $(255 \div 256) * AV_{CC}$ and is obtained for a value of FF_{16}. The minimum output, 0V, is obtained for a value of 00_{16}.

The reference voltage of the converters is the AV_{CC} analog power supply voltage. This allows full swing of the outputs from 0V to nearly AV_{CC}.

After reset, all four channels are disabled and the voltage on the DA0-3 outputs is 0V.

In Idle mode, the DAC channels may be enabled or disabled (to reduce power consumption). Two control modes are provided:

- Automatic disable of all channels on entering Idle mode
- Selective disable of channels by software, before entering Idle mode

4.12.3 D/A Conversion

Output Signal

The DAC performs a linear conversion of the input digital value DACDATA7-0 registers to a unipolar analog output signal, relative to the analog ground pin (AGND).

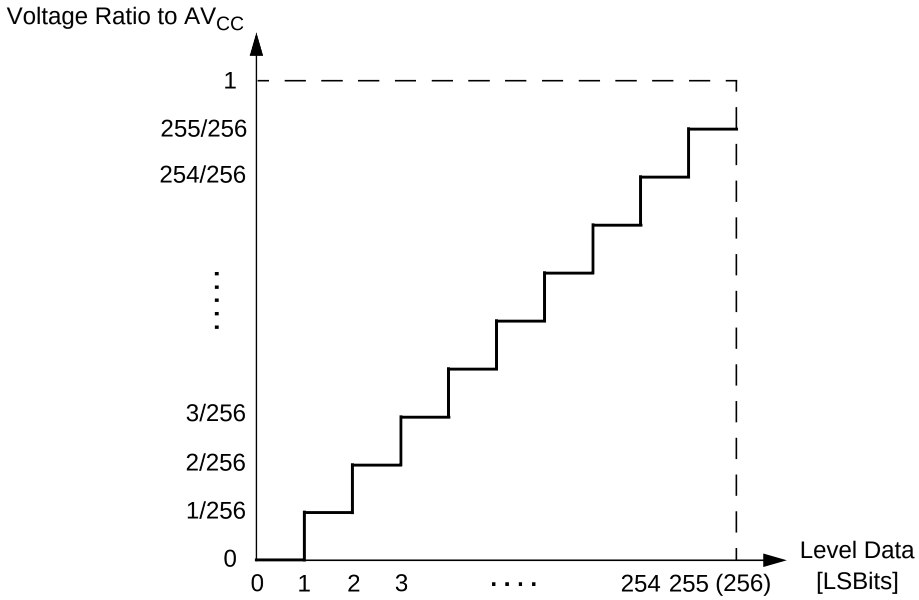


Figure 66. Channel Data to Output Voltage Ratio Conversion

When the value of DACDATA7-0 is 00_{16}, the respective output has an output signal of 0V (AGND). When the value of DACDATn register is FF_{16}, the respective output has an output signal of (255/256) * AV_{CC}. For other values, as shown in Figure 66:

$$V_{OUT} = (\mathrm{DACDATAn}) * (AV_{CC} / 256)$$

Reference Voltage

The analog output voltages are converted relative to a reference voltage. The reference voltage of the converters is the analog power supply. To assure good signal quality at the output, a low-noise analog power supply should be used.

Conversion Time

When a DAC channel is enabled, the conversion is started by writing to DACDATn registers. The output settling time is defined as the time the DAC requires to get to within 1/2 LSB of the final value; see “Output Settling Time” on page 195.

Embedded Controller Modules (Continued)

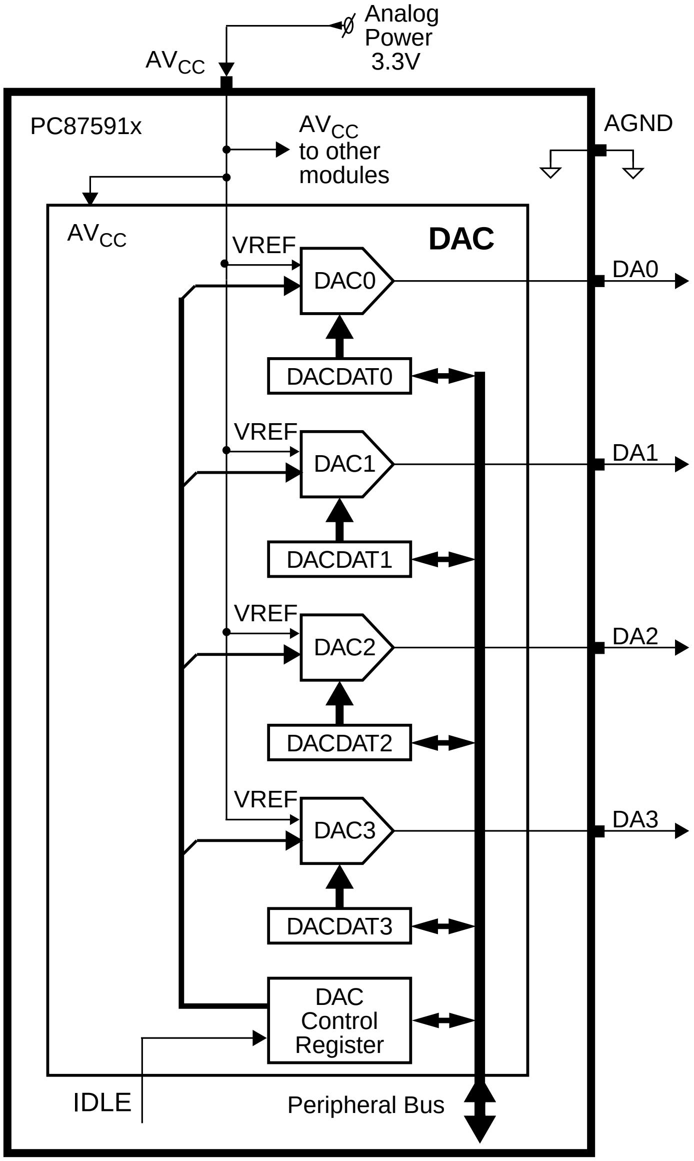


Figure 67. DAC Functional Diagram

4.12.4 Operation

Initializing the DAC

The PC87591x wakes up after power-up with all the D/A channels disabled (DACEN0-3 bits in DACCTRL register are cleared to 0). In this state, all DAC activities are halted, and its current consumption is reduced to zero.

DACDATn registers (n=0 to 3) must be initialized to 00_{16}, or according to the required output level, before setting DACEN0-3 in DACCTRL register to 1.

Enabling and Disabling the DAC

Enabling the DAC. Each channel of the DAC is enabled independently by setting its DACEN bit. After enabling, it settles to the value stored in DACDATn register after the specified settling time.

Disabling the DAC. The DAC channels may be independently disabled in order to reduce current consumption by clearing the corresponding DACENn (n=0 to 3) bit in DACCTRL register. In this case, the output pin drives 0V, even if the respective DACDATn register does not contain 00_{16}.

All DAC channels are automatically disabled when entering Idle mode if ENIDLE bit in DACCTRL register is cleared to 0. This happens regardless of the state of DACENn (n=0 to 3) bit in DACCTRL register. In this case, the DA0-3 outputs drive 0V.

If the ENIDLE bit is set to 1, entering the Idle mode does not affect DAC operation, and DA0-3 outputs drive the voltage level set by DACDATn (n=0 to 3) registers.

4.12.5 DAC Registers

The DAC interfaces with the core using one control and four data registers. These registers are mapped to the core address space, as defined in Appendix A on page 408.

DAC Register Map

Mnemonic	Register Name	Type
DACCTRL	DAC Control	R/W
DACDAT0	DAC Data Channel 0	R/W
DACDAT1	DAC Data Channel 1	R/W
DACDAT2	DAC Data Channel 2	R/W
DACDAT3	DAC Data Channel 3	R/W

DAC Control Register (DACCTRL)

This register controls the operation of the DAC module. DACCTRL is cleared (00_{16}) on reset.

Location: 00 $FF40_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**ENIDLE**	**DACEN3**	**DACEN2**	**DACEN1**	**DACEN0**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**DACEN0 (DAC Channel 0 Enable).** Enables the DAC channel. The DA0 output pin drives a voltage level, according to the value written into the corresponding DACDAT0 register. When cleared, the DA0 output pin drives 0V. 0: Disabled (default) 1: Enabled
1	**DACEN1 (DAC Channel 1 Enable).** Same as DACEN0 bit description, using DA1 output and DACDAT1 register.
2	**DACEN2 (DAC Channel 2 Enable).** Same as DACEN0 bit description, using DA2 output and DACDAT2 register.
3	**DACEN3 (DAC Channel 3 Enable).** Same as DACEN0 bit description, using DA3 output and DACDAT3 register.
4	**ENIDLE (Enable in Idle).** Controls the DAn (n=0 to 3) outputs in Idle mode. 0: Disabled - DAn outputs drive 0V (default) 1: Enabled - DAn outputs according to DACENn bits and DACDATn registers
7-5	**Reserved.**

DAC Data Channel 0-3 Registers (DACDAT0-3)

These registers hold the data to be loaded into Channels 0-3 of the DAC. These registers are not affected by reset or disable of the respective channel.

Location: Channel 0 - 00 $FF42_{16}$
Channel 1 - 00 $FF44_{16}$
Channel 2 - 00 $FF46_{16}$
Channel 3 - 00 $FF48_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**DAC DATAi**							

Bit	Description
7-0	**DAC Data.** 8-bit unsigned binary value used for the D/A operation.

4.12.6 Usage Hints

Power Consumption

When a channel is enabled and no load is connected, the DAC current consumption depends on the value set in DACDAT register. Minimal current is consumed when the data is 00_{16}; maximum current is consumed when the data is 55_{16}.

DAC Output Protection

To maintain the high performance of the analog circuits, the DAn (n=0 to 3) pins are not back-drive protected. Therefore, the voltage applied to these pins must be within the AGND to AV_{CC} range; otherwise the device may be damaged.

External circuits should not drive currents into these pins when the PC87591x is not powered up, because this may cause the internal power-up reset circuit to fail.

Output Voltage Accuracy

Besides the intrinsic accuracy of the D/A channels, the output voltage accuracy directly depends on the accuracy of the AV_{CC} power supply, which serves as reference voltage. In order to improve the accuracy of the output voltage, the actual AV_{CC} value, measured by the ADC module (see Section 4.11 on page 170), should be used when computing the value of DACDATA7-0 in DACDATn registers (see Section 4.12.3 on page 190).

The external load on DA0-3 pins may also affect the final output voltage of the DAC. Since the output resistance of these pins is typically 3 KΩ, use high-impedance loads; if high accuracy or high output currents are required, use external analog drivers.

For the worst case calculation, if the output resistance is 4 KΩ (maximum limit), the external load (R_L) must not be lower than 2 MΩ (see Figure 68). In this case, the error caused by the load is lower than 1/2 LSB.

To work with loads of 3 KΩ (1 mA at 3V) with an error lower than 1/2 LSB, the output resistance of the external driver should be lower than:

$$R_{0EXT} < 3\ \mathrm{K\Omega} / (2 * 256) = 5.8\ \Omega$$

Embedded Controller Modules (Continued)

Output Settling Time

The DAC output settling time depends on the external load characteristics and the required accuracy. Figure 68 shows the equivalent circuit used for evaluating DAC behavior. Each DAC output has a typical output impedance of 3 KΩ. For example, if the total load is a 50 pF capacitor only, the output settles to 1/2 LSB within 1 µs. The total load capacitance is comprises the analog output capacitance (C_{AO}) and the external load capacitance (C_L).

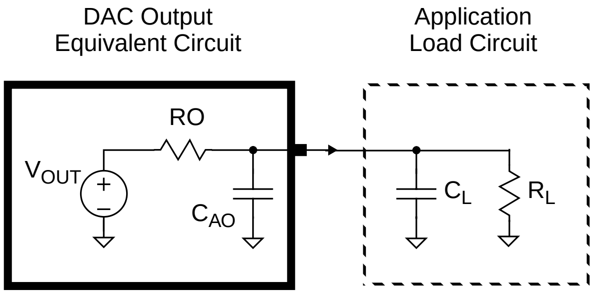


Figure 68. DAC Output Equivalent Circuit

Filtering Noise on Output Signals

Output signals may have unwanted noise caused by nearby digital circuits. When using slow changing signals in a noisy environment, a low-pass filter (LPF) may be added externally. This may also be required in applications where the DAC outputs control sensitive circuits like audio amplifiers. This can be implemented as a simple RC circuit. The cutoff frequency of this LPF should be above the required signal frequency.

Embedded Controller Modules (Continued)

4.13 ACCESS.BUS (ACB) INTERFACE

The PC87591x includes two ACCESS.bus interface modules. The registers of each module are prefixed with ACBn, where ‘n’ is module number 0 or 1. The signal names are suffixed with ‘n’.

Each ACCESS.bus (ACB) interface is a two-wire serial interface that is compatible with the ACCESS.bus physical layer. It is also compatible with Intel’s SMBus and Philips’ I^{2}C bus. The module can be configured as either a bus master or slave and can maintain bidirectional communication with multiple master and slave devices. As a slave device, the ACB module may issue a request to become the bus master.

The ACB interface provides full support for a two-wire ACCESS.bus synchronous serial interface. It permits easy interfacing to a wide range of low-cost memories and I/O devices, including EEPROMs, SRAMs, timers, A/D converters, D/A converters, clock chips and peripheral drivers.

4.13.1 Features

- ACCESS.bus, SMBus and I^{2}C compliant
- ACCESS.bus master
- ACCESS.bus slave
  - One or two user-defined addresses
  - Global (broadcast) address
  - ARP address
- Supports polling- or interrupt-controlled operation
- Generates a wake-up signal on detection of a Start Condition in Power-Down mode
- Optional internal pull-up on SDAn and SCLn pins

4.13.2 Functional Description

The ACCESS.bus protocol uses a two-wire interface for bidirectional communication between the ICs connected to the bus. The two interface lines are the Serial Data Line (SDL) and the Serial Clock Line (SCLn). These lines should be connected to a positive supply via a pull-up resistor and remain high even when the bus is idle.

The ACCESS.bus protocol supports multiple master and slave transmitters and receivers. Each IC has a unique address and can operate as a transmitter or a receiver. Some peripherals are receivers only.

During data transactions, the master device initiates the transaction, generates the clock signal and terminates the transaction. For example, when the ACB initiates a data transaction with an attached ACCESS.bus-compliant peripheral, the ACB becomes the master. When the peripheral responds and transmits data to the ACB, their master/slave (data transaction initiator and clock generator) relationship is unchanged even though their transmitter/receiver functions are reversed.

Data Transactions

One data bit is transferred during each clock pulse. Data is sampled during the high state of the serial clock (SCLn). Consequently, throughout the clock’s high period, the data should remain stable (see Figure 69). Any change on the SDAn line while SCLn is in high state during a transaction causes the current transaction to abort. New data should be sent during the low SCLn state. This protocol permits a single data line to transfer both command/control information and data, using the synchronous serial clock.

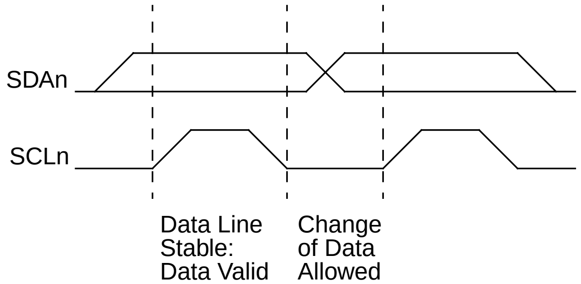


Figure 69. Bit Transfer

Each data transaction is composed of a Start Condition, a number of byte transfers (set by the software) and a Stop Condition to terminate the transaction. Each byte is transferred with the most significant bit first. An Acknowledge signal must follow each byte (8 bits). The following sections provide further details of this process.

Embedded Controller Modules (Continued)

At each clock cycle, the slave can stall the master while it handles the previous data or prepares new data. The slave does this, for each bit transferred or on a byte boundary, by holding SCLn low to extend the clock low period. Typically, slaves extend the first clock cycle of a transfer if a byte read has not yet been stored or if the next byte to be transmitted is not yet ready. Some microcontrollers with limited hardware support for the ACCESS.bus extend the access after each bit, thus allowing the software time to handle this bit.

The ACCESS.bus master generates Start and Stop Conditions (control codes). After a Start Condition is generated, the bus is considered busy. It retains this status for a given amount of time after a Stop Condition is generated. A high-to-low transition of the data line (SDAn) while the clock (SCLn) is high indicates a Start Condition. A low-to-high transition of the SDAn line while the SCLn is high indicates a Stop Condition (Figure 70).

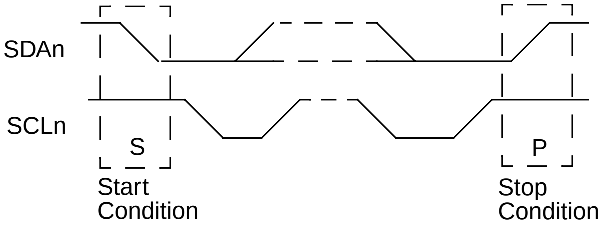


Figure 70. Start and Stop Conditions

In addition to the first Start Condition, a Repeated Start Condition can be generated in the middle of a transaction. This allows either another device to be accessed or a change in the direction of the data transfer.

Acknowledge Cycle

The Acknowledge cycle consists of two signals:

- Acknowledge Clock pulse is sent by the master with each byte transferred
- Acknowledge signal is sent by the receiving device (Figure 71)

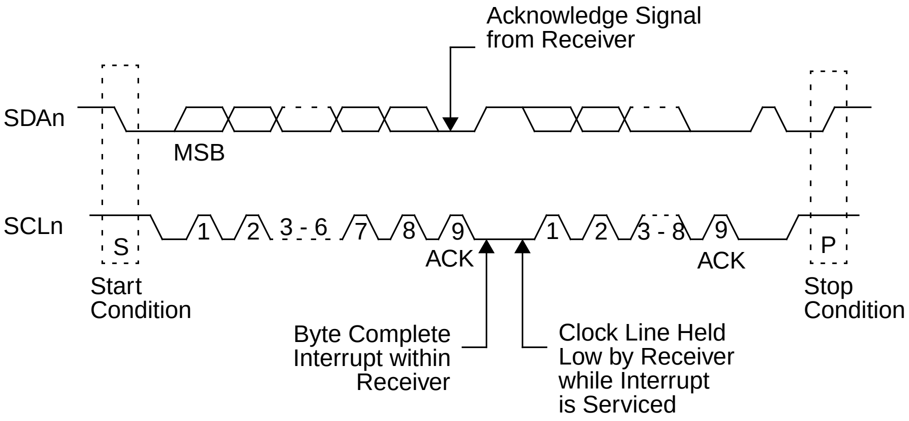


Figure 71. ACCESS.bus Data Transaction

The master generates the Acknowledge clock pulse on the ninth clock pulse of the byte transfer. The transmitter releases the SDAn line (permitting it to go high) to allow the receiver to send the Acknowledge signal.The receiver pulls down the SDAn line during the Acknowledge clock pulse, thus signalling that it has correctly received the last data byte and is ready to receive the next byte. Figure 72 illustrates the Acknowledge cycle.

Embedded Controller Modules (Continued)

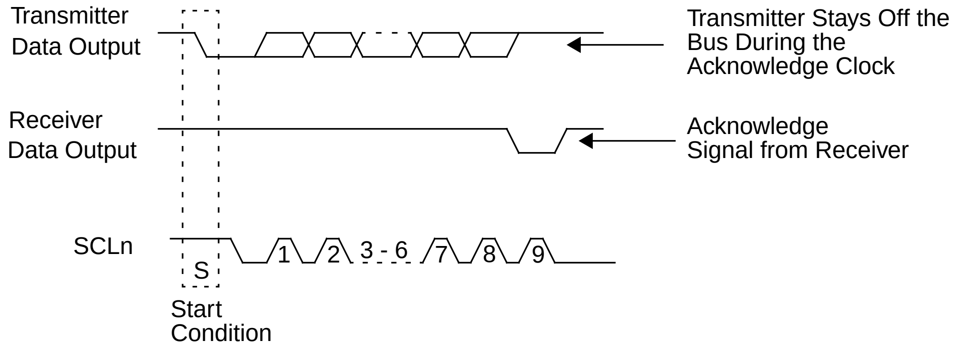

Transmitter
Data Output
Transmitter Stays Off the Bus During the Acknowledge Clock
Receiver
Data Output
Acknowledge Signal from Receiver
SCLn
S
1
2
3 - 6
7
8
9
Start
Condition

Figure 72. ACCESS.bus Acknowledge Cycle

“Acknowledge After Every Byte” Rule

The master generates an Acknowledge clock pulse after each byte transfer. The receiver sends an Acknowledge signal after every byte is received.

There are two exceptions to the “acknowledge after every byte” rule:

- When the master is the receiver, it must indicate to the transmitter an end of data by not acknowledging (negative acknowledge) the last byte clocked out of the slave. This negative acknowledge still includes the Acknowledge clock pulse (generated by the master), but the SDAn line is not pulled down.
- When the receiver is full or otherwise occupied, or if a problem occurs, it sends a negative acknowledge to indicate that it cannot accept additional data bytes.

Addressing Transfer Formats

Each device on the bus has a unique address. Before any data is transmitted, the master transmits the address of the slave being addressed. The slave device should send an Acknowledge signal on the SDAn line once it recognizes its address.

The address consists of the first seven bits after a Start Condition. The eighth bit contains the direction of the data transfer (R/$\overline{W}$). A low-to-high transition during a SCLn high period indicates the Stop Condition and ends the transaction of SDAn (Figure 73).

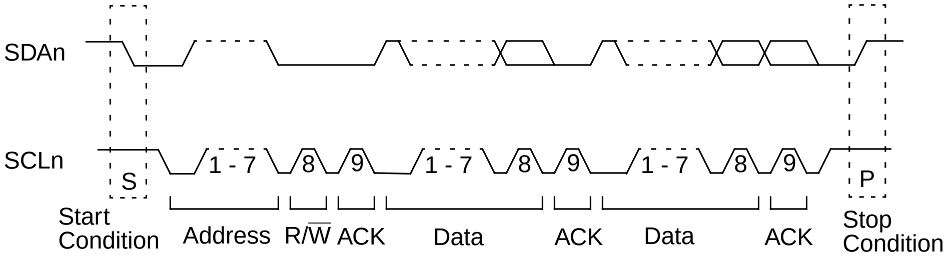


Figure 73. A Complete ACCESS.bus Data Transaction

When the address is sent, each device in the system compares this address with its own. If there is a match, the device considers itself addressed and sends an Acknowledge signal. Depending on the state of the R/$\overline{W}$ bit (1=read, 0=write), the device acts as a transmitter or a receiver.

The I^{2}C bus protocol allows a general call address to be sent to all slaves connected to the bus. The first byte sent specifies the general call address (00_{16}); the second byte specifies the general call meaning (for example, “Write slave address by software only”). Slaves that require data acknowledge the call and become slave receivers; other slaves ignore the call.

Arbitration on the Bus

Multiple master devices on the bus require arbitration between their conflicting bus access demands. Control of the bus is initially determined according to address bits and clock cycle. If more than one master tries to address the same slave, data comparisons determine the outcome of this arbitration. In Master mode, the device immediately aborts a transaction if the value sampled on the SDAn line differs from the value driven by the device. (An exception to this rule is SDAn while receiving data; in this case, the lines may be driven low by the slave without causing an abort.)

The SCLn signal is monitored for clock synchronization to allow the slave to stall the bus. The actual clock period is the longest one set by the master or the slave stall period. The clock high period is determined by the master with the shortest clock high period.

When an abort occurs during address transmission, a master that identifies the conflict should give up the bus and switch to Slave mode. It should then continue to sample SDAn to see if it is being addressed by the winning master on the bus.

4.13.3 Master Mode

Requesting Bus Mastership

An ACCESS.bus transaction starts with a master device requesting bus mastership. It asserts a Start Condition, followed by the address of the device it wants to access. If this transaction is successfully completed, the software may assume that the device has become the bus master.

For the device to become the bus master, the software should perform the following steps:

1. Configure INTEN in ACBnCTL1 register to the desired operation mode (Polling or Interrupt) and set START in the same register. This causes the ACB to issue a Start Condition on the ACCESS.bus as soon as the ACCESS.bus is free (some conditions, such as when BB in ACBnCST register is set to 0, can delay start). It then stalls the bus by holding SCLn low.
2. If a bus conflict is detected (i.e., some other device pulls down the SCLn signal before the PC87591x does), BER in ACBnST register is set.
3. If there is no bus conflict, MASTER and SDAST in ACBnST register are set.
4. If INTEN in ACBnCTL1 register is set and either BER or SDAST in ACBnST register is set, an interrupt is sent to the core.

Sending the Address Byte

Once the PC87591x is the active master of the ACCESS.bus (MASTER in ACBnST register is set), it can send the address on the bus. The address sent should **not** be any of the following:

- The PC87591x's own address, as defined by ADDR in ACBnADDR register, if SAEN in ACBnADDR is set.
- The PC87591x's own address, as defined by ADDR in ACBnADDR2, if SAEN in ACBnADDR2 is set.
- The global call address, if GCMTCH in ACBnST register is set.
- The ARP address, if ARPMATCH in ACBnST register is set.

To send the address byte, use the following sequence:

1. For a receive transaction where the software requires only one byte of data, the software should set ACK in ACBnCTL1 register. If only an address needs to be sent (e.g., for quick read/write protocols) or if the device requires stall for some other reason, set STASTRE in ACBnCTL1 register to 1.
2. Write the address byte (7-bit target device address) and the direction bit to ACBnSDA register. This causes the module to generate a transaction. At the end of this transaction, the acknowledge bit received is copied to NEGACK in ACBnST register. During the transaction, the SDAn and SCLn lines are continuously checked for conflict with other devices. If a conflict is detected, the transaction is aborted, BER in ACBnST register is set and MASTER in ACBnST register is cleared.
3. If STASTRE in ACBnCTL1 register is set and the transaction was successfully completed (i.e., both BER and NEGACK in ACBnST register are cleared), STASTR in ACBnST register is set. In this case, the ACB stalls any further ACCESS.bus operations (i.e., holds SCLn low). If INTEN in ACBnCTL1 register is set, it also sends an interrupt to the core.
4. If the requested direction is transmit and the start transaction was completed successfully (i.e., neither NEGACK nor BER in ACBnST register is set and no other master has accessed the device), SDAST in ACBnST register is set to indicate that the module awaits attention.
5. If the requested direction is receive, the start transaction was completed successfully and STASTRE in ACBnCTL1 register is cleared, the module starts receiving the first byte automatically.
6. Check that both BER and NEGACK in ACBnST register are cleared. If INTEN in ACBnCTL1 register is set, an interrupt is generated when either BER or NEGACK is set.

Embedded Controller Modules (Continued)

Master Transmit

After becoming the bus master, the device can start transmitting data on the ACCESS.bus.

To transmit a byte, the software should:

1. Check that BER and NEGACK bits in ACBnST register are cleared and SDAST bit is set. In addition, if STASTRE bit in ACBnCTL1 register is set, make sure that STASTR bit in ACBnST register is cleared.
2. Write the data byte to be transmitted to ACBnSDA register.

When the slave responds with a negative acknowledge, NEGACK in ACBnST register is set and SDAST in ACBnST register remains cleared. In this case, if INTEN bit in ACBnCTL1 register is set, an interrupt is sent to the core.

Master Receive

After becoming the bus master, the device can start receiving data on the ACCESS.bus.

To receive a byte, the software should:

1. Check that SDAST bit in ACBnST register is set and BER bit is cleared. In addition, if STASTRE bit in ACBnCTL1 register is set, make sure that STASTR in ACBnST register is cleared.
2. If the next byte is the last byte that should be read, set ACK bit in ACBnCTL1 register to 1. This causes a negative acknowledge to be sent.
3. Read the data byte from ACBnSDA register.

Master Stop

To end a transaction, set STOP in ACBnCTL1 register before clearing the current stall flag (i.e., SDAST, NEGACK or STASTR in ACBnST register). This causes the module to send a Stop Condition immediately and to clear STOP in ACBnCTL1 register. A Stop Condition may be issued only when the PC87591x is the active bus master (MASTER in ACBnST register is set to 1).

Master Bus Stall

The ACB module can stall the ACCESS.bus between transfers while waiting for the core’s response. The ACCESS.bus is stalled by holding the SCLn signal low after the acknowledge cycle. Note that this is interpreted as the start of the following bus operation. The user must make sure that the next operation is prepared before the flag that causes the bus stall is cleared.

The flags that can cause a bus stall in Master mode are:

- Negative acknowledge after sending a byte (NEGACK in ACBnST register is set to 1).
- SDAST in ACBnST register is set to 1.
- STASTRE in ACBnCTL1 register is set to 1 after a successful start (STASTR in ACBnST is set to 1).

Repeated Start

A repeated start is performed when the PC87591x is already the bus master (MASTER in ACBnST register is set). In this case, the ACCESS.bus is stalled and the ACB module awaits core handling due to a negative acknowledge (NEGACK in ACBnST register is set to 1), an empty buffer (SDAST in ACBnST is set to 1) and/or a stall after start (STASTR in ACBnST is set to 1).

For a repeated start:

1. Set START in ACBnCTL1 register to 1.
2. In Master Receive mode, read the last data item from ACBnSDA.
3. Follow the address send sequence, as described in “Sending the Address Byte” on page 199.
4. If the ACB is awaiting handling because STASTR in ACBnST is set to 1, clear it only after writing the requested address and direction to ACBnSDA.

Master Error Detection

The ACB detects an illegal Start or Stop Condition (i.e., a Start or Stop Condition within the data transfer or the acknowledge cycle) and a conflict on the data lines of the ACCESS.bus. If an illegal condition is detected, BER is set and Master mode is exited (MASTER in ACBnST register is cleared).

Bus Idle Error Recovery

When a request to become the active bus master or a restart operation fails, BER in ACBnST register is set to indicate the error. In some cases, both the PC87591x and the other device may identify the failure and leave the bus idle. In this case, the start sequence may not finish and the ACCESS.bus may remain deadlocked.

To recover from deadlock, use the following sequence:

1. Clear BER in ACBnST register and BB in ACBnCST register.
2. Wait for a time-out period to check that there is no other active master on the bus (i.e., BB in ACBnCST remains cleared).
3. Disable and then re-enable the ACB to put it in non-addressed Slave mode. (This completely resets the module.)

At this point, some of the slaves may not identify the bus error. To recover, the ACB module becomes the bus master. It asserts a Start Condition, sends an address byte and then asserts a Stop Condition that synchronizes all the slaves.

4.13.4 Slave Mode

A slave device waits in Idle mode for a master to initiate a bus transaction. Whenever the ACB module is enabled is not acting as a master (i.e., MASTER in ACBnST register is cleared), it acts as a slave device.

Once a Start Condition on the bus is detected, the PC87591x checks whether the address sent by the current master matches any of the following possibilities:

- The ADDR value in ACBnADDR register, if SAEN in this register is set to 1
- The ADDR value in ACBnADDR2 register, if SAEN in this register is set to 1
- The global call address (00_{16}), if GCMEM in ACBnCTL1 register is set to 1
- The global ARP address ($110\ 0001_2$), if ARPMEN in ACBnCTL3 register is set to 1.

The address match is checked even when MASTER in ACBnST register is set. If a bus conflict (on SDAn or SCLn) is detected, BER is set, MASTER is cleared and the PC87591x continues to search the received message for a match.

If an address ARP or global match is detected:

1. The PC87591x asserts its SDAn pin during the acknowledge cycle.
2. MATCH in ACBnCST register, MATCHAF in ACBnST register (or GCMATCH if it is a global call address match, or ARPMATCH if it is an ARP address) and NMATCH in ACBnST register are set. If XMIT in ACBnST register is set (i.e., Slave Transmit mode), SDAST in the same register is also set to indicate that the buffer is empty.
3. If INTEN in ACBnCTL1 register is set, an interrupt is generated if both INTEN and NMINTE in ACBnCTL1 register are set.
4. The software then reads XMIT in ACBnST register to identify the direction requested by the master device; it then clears NMATCH in the same register so that future byte transfers are identified as data bytes.

Slave Receive and Transmit

Slave Receive and Transmit are performed after a match is detected and the data transfer direction is identified. After a byte transfer, the ACB module extends the acknowledge clock until the software reads or writes ACBnSDA register. The receive and transmit sequences are identical to those used in the master routine.

Slave Bus Stall

When operating as a slave, the PC87591x stalls the ACCESS.bus by extending the first clock cycle of a transaction in the following cases:

- SDAST in ACBnST register is set.
- NMATCH in ACBnST register and NMINTE in ACBnCTL1 register are set.

Slave Error Detection

The ACB detects illegal Start and Stop Conditions (occurring within the data transfer or the acknowledge cycle) on the ACCESS.bus. When an illegal Start or Stop Condition is detected, BER is set and MATCH and GMATCH are cleared, setting the module as an unaddressed slave.

4.13.5 Power-Down

When the PC87591x is in Idle mode, the ACB module is not active but retains its status. If the ACB is enabled (ENABLE in ACBnCTL2 register is set) on detection of a Start Condition, a wake-up signal is issued to the MIWU. This signal may be used to switch the PC87591x to Active mode.

Following the Start Condition that woke up the PC87591x, the ACB module can not check the address byte for a match. The ACB responds with a negative acknowledge. The device should resend both the Start Condition and the address after the PC87591x has had time to wake up.

Before entering Idle mode, make sure that BUSY in ACBnCST register is inactive. This guarantees that the PC87591x will not stop responding after it acknowledges an address that was sent.

4.13.6 SDA and SCL Pin Configuration

The SDAn and SCLn are open collector signals that the user can choose to enable or disable. SDAn and SCLn also have internal pull-up resistors that the user may enable. For more information about configuring these pins, see Table 8 on page 54 and Section 4.5.2 on page 114.

4.13.7 ACB Clock Frequency Configuration

The ACB module enables the user to set the ACCESS.bus clock frequency. The SCLn clock period is set by SCLFRQ in ACBnCTL2 and ACBnCTL3 registers. This clock low period may be extended by stall periods initiated by the ACB module or by another ACCESS.bus device. In case of a conflict with another bus master, a shorter clock high period may be forced by the other bus master until the conflict is resolved.

4.13.8 ACB Registers

For a summary of the abbreviations used for Register Type, see "Register Abbreviations and Access" on page 34.

ACB Register Map

Mnemonic	Register Name	Type
ACBnSDA	ACB Serial Data	R/W
ACBnST	ACB Status	Varies per bit
ACBnCST	ACB Control Status	Varies per bit
ACBnCTL1	ACB Control 1	R/W
ACBnADDR	ACB Own Address	R/W
ACBnADDR2	ACB Own Address 2	R/W
ACBnCTL2	ACB Control 2	R/W
ACBnCTL3	ACB Control 3	R/W

ACB Serial Data Register (ACBnSDA)

The ACBnSDA register is a shift register used to transmit and receive data. The most significant bit is transmitted (received) first and the least significant bit is transmitted (received) last. Reading or writing to the ACBnSDA register is allowed only when SDAST in ACBnST register is set or for repeated starts after setting the START bit. An attempt to access the register in other cases may produce unpredictable results.

Locations: Channel - 1 00 FF60$_{16}$
Channel - 2 00 FFE0$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Data**							

Bit	Description
7-0	**Data.**

Embedded Controller Modules (Continued)

ACB Status Register (ACBnST)

The ACBnST register maintains current ACB status. Some of its bits may be cleared by software, as described below. On reset, and when the module is disabled, ACBnST is cleared (00_{16}).

Location: Channel - 1 00 $FF62_{16}$
Channel - 2 00 $FFE2_{16}$

Type: varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**SLVSTP**	**SDAST**	**BER**	**NEGACK**	**STASTR**	**NMATCH**	**MASTER**	**XMIT**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	RO	**XMIT (Transmit Mode).** 0: ACB not in master/slave Transmit mode 1: ACB in master/slave Transmit mode
1	RO	**MASTER (Master Mode).** 0: Arbitration loss (BER is set) or Stop Condition occurred 1: ACB in Master mode (successful request for bus mastership)
2	R/W1C	**NMATCH (New Match).** The bit is set when the address byte following a Start Condition or a repeated start causes an address match, ARP address match or a global call match. NMATCH is cleared by writing 1 to it. Writing 0 to NMATCH is ignored. If INTEN in ACBnCTL1 register is set, an interrupt is sent when this bit is set.
3	R/W1C	**STASTR (Stall After Start).** The bit is set by the successful completion of sending an address (i.e., a Start Condition sent without a bus error or negative acknowledge), if STASTRE in ACBnCTL1 register is set. This bit is ignored in Slave mode. When STASTR is set, it stalls the ACCESS.bus (by pulling down the SCL line) and suspends any further action on the bus (e.g., receiving the first byte in Master Receive mode). In addition, if INTEN in ACBnCTL1 register is set, it also causes the ACB module to send an interrupt to the core. Writing 1 to STASTR clears it. It is also cleared when the module is disabled and is always cleared when STASTRE is cleared. Writing 0 to STASTR has no effect.
4	R/W1C	**NEGACK (Negative Acknowledge).** The bit is set by hardware when a transmission is not acknowledged on the ninth clock (in this case, SDAST is not set). Writing 1 to NEGACK clears it. It is also cleared when the module is disabled. Writing 0 to NEGACK is ignored.
5	R/W1C	**BER (Bus Error).** The bit is set by the hardware when a Start or Stop Condition is detected during data transfer (i.e., Start or Stop Condition during the transfer of bits 2 through 8 and acknowledge cycle) or when an arbitration problem is detected. Writing 1 to BER clears it. It is also cleared when the module is disabled. Writing 0 to BER is ignored.
6	RO	**SDAST (SDA Status).** When set, this bit indicates that the SDA data register is waiting for data (Transmit mode - master or slave) or holds data that should be read (Receive mode - master or slave). This bit is cleared when reading from ACBnSDA register during a receive or when written to during a transmit. When START in the ACBnCTL1 is set, reading ACBnSDA register does not clear SDAST. This enables the ACB to send a repeated start in Master Receive mode.
7	R/W1C	**SLVSTP (Slave Stop).** When set, SLVSTP indicates that a Stop Condition was detected after a slave transfer (i.e., after a slave transfer in which MATCH or GCMATCH was set). Writing 1 to SLVSTP clears it. It is also cleared when the module is disabled. Writing 0 to SLVSTP is ignored.

Embedded Controller Modules (Continued)

ACB Control Status Register (ACBnCST)

The ACBnCST register maintains current ACB status and controls several ACB module functions, as described below. On reset and when the module is disabled, the non-reserved bits of ACBnCST are cleared (00_{16}).

Location: Channel 1 - 00 $FF64_{16}$
Channel 2 - 00 $FFE4_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**ARPMATCH**	**MATCHAF**	**TGSCL**	**TSDA**	**GCMATCH**	**MATCH**	**BB**	**BUSY**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	RO	**BUSY.** When set (1), indicates that the ACB module is in one of the following states: – Generating a Start Condition – In Master mode (MASTER in ACBnST register is set) – In Slave mode (MATCH or GMATCH in ACBnCST register is set) – In the period between detecting a Start Condition and completing the reception of the address byte; after this, the ACB either becomes not busy or enters Slave mode. The BUSY bit is cleared by the completion of any of the above states or by disabling the module. It should always be written 0.
1	R/W1C	**BB (Bus Busy).** When set (1), indicates the bus is busy. It is set either when the bus is active (i.e., a low level on either SDA or SCL) or by a Start Condition. It is cleared when the module is disabled, on detection of a Stop Condition or by writing 1 to this bit. See Section 4.13.9 on page 207 for a description of the use of this bit.
2	RO	**MATCH (Address Match).** In Slave mode, MATCH is set (1) when SAEN in ACBnADDR register is set and the first seven bits of the address byte (the first byte transferred after a Start Condition) match the 7-bit address in ACBnADDR register. It is cleared by Start Condition, a Repeated Start or a Stop Condition (including illegal Start or Stop Condition).
3	RO	**GCMATCH (Global Call Match).** In Slave mode, GCMTCH is set (1) when GCMEN in ACBnCTL1 register is set and the address byte (the first byte transferred after a Start Condition) is 00_{16}. It is cleared by a Start Condition, a Repeated Start or a Stop Condition (including illegal Start or Stop Condition).
4	RO	**TSDA (Test SDA Line).** Reads the current value of the SDA line. This bit can be used while recovering from an error condition in which the SDA line is constantly pulled low by a slave that went out of synch. Data written to this bit is ignored.
5	R/W	**TGSCL (Toggle SCL Line).** Enables toggling the SCL line during the process of error recovery. When the SDA line is low, writing 1 to this bit toggles the SCL line for one cycle. Writing 1 to TGSCL while SDA is high is ignored. The bit is cleared when the clock toggle is completed.
6	RO	**MATCHAF (Match Address Field).** When the MATCH bit is set, MATCHAF indicates with which of the two possible slave addresses (ADDR cleared in ACBnADDR register or set in ACBnADDR2 register) the match has occurred. If both addresses match, the bit is cleared.
7	RO	**ARPMATCH (ARP address Match).** In Slave mode, ARPMTCH is set (1) when ARPMEN in ACBnCTL3 register is set and the address byte (the first byte transferred after a Start Condition) is $110\ 0001_2$. It is cleared by Start Condition, a Repeated Start or a Stop Condition (including illegal Start or Stop Condition).

Embedded Controller Modules (Continued)

ACB Control Register 1 (ACBnCTL1)

The ACBnCTL1 register is a byte-wide, read/write register that configures and controls the ACB module. On reset, the ACBnCTL1 is cleared (00_{16}).

Location: Channel 1 - 00 $FF66_{16}$
Channel 2 - 00 $FFE6_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**STASTRE**	**NMINTE**	**GCMEN**	**ACK**	**Reserved**	**INTEN**	**STOP**	**START**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**START.** Should be set when a Start Condition must be generated on the ACCESS.bus. – If the PC87591x is not the active bus master (MASTER in ACBnST register is set to 0), setting START generates a Start Condition as soon as the ACCESS.bus is free (BB in ACBnCST register is set to 0). An address transmission sequence should then be performed. – If the PC87591x is the active master of the bus (MASTER in ACBnST register is set to 1), when START is set, a write to ACBnSDA register generates a Start Condition. ACBnSDA data is then transmitted as the slave's address and the requested transfer direction. In case of a Repeated Start Condition, the set bit may be used to switch the direction of the data flow between the master and the slave or to choose another slave device without using a Stop Condition in between. The START bit is cleared either when the Start Condition is sent or on detection of a Bus Error (BER in ACBnST register is set to 1). This bit should be set only when in Master mode or when requesting Master mode.
1	**STOP.** In Master mode, setting this bit generates a Stop Condition, which completes or aborts the current message transfer. This bit clears itself after STOP is issued.
2	**INTEN (Interrupt Enable).** When INTEN is cleared (0), the ACB interrupt is disabled. When INTEN is set, interrupts are enabled. An interrupt is generated (the interrupt signals to the ICU are high) on one of the following events: – An address match is detected (NMATCH in ACBnST register is set to1 and NMINTE in ACBnCTL1 register is set to 1). – A Bus Error occurs (BER in ACBnST register is set to 1). – A negative acknowledge is received after sending a byte (NEGACK in ACBnST register is set to 1). – Acknowledgment of each transaction (same as the hardware set of SDAST in ACBnST). – In Master mode, if STASTRE in ACBnCTL1 register is set to 1 after a successful start (STASTR in ACBnST register is set to 1). – Detection of a Stop Condition while in Slave mode (SLVSTP in ACBnST register is set to 1).
3	**Reserved.**
4	**ACK (Acknowledge).** When acting as a receiver, this bit holds the value of the next acknowledge cycle. It should be set when a negative acknowledge must be issued on the next byte. This bit is cleared (0) after the first acknowledge cycle. This bit is ignored when in Transmit mode. It cannot be reset by software.
5	**GCMEN (Global Call Match Enable).** When set, enables the matching of an incoming address byte to the general call address (Start Condition followed by address byte of 00_{16}) while the ACB is in Slave mode. When cleared, the ACB does not respond to a global call.
6	**NMINTE (New Match Interrupt Enable).** When set, enables the interrupt on a new match (i.e., when NMATCH in ACBnST register is set). The interrupt is issued only if INTEN in ACBnCTL1 register is set.
7	**STASTRE (Stall After Start Enable).** When set (1), enables the Stall After Start mechanism. In such a case, the ACB stalls the bus after the address byte. When STASTRE is cleared, STASTR in ACBnST is always cleared.

Embedded Controller Modules (Continued)

ACB Own Address Register (ACBnADDR and ACBnADDR2)

The ACBnADDR and ACBnADDR2 registers hold the module's ACCESS.bus addresses. The reset value of these registers are undefined.

ACBnADDR:
Location: Channel 1 - 00 $FF68_{16}$
Channel 2 - 00 $FFE8_{16}$

ACBnADDR2:
Location: Channel 1- 00 $FE6C_{16}$
Channel 2- 00 $FFEC_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**SAEN**	**ADDR**						

Bit	Description
6-0	**ADDR (Address).** Holds the 7-bit ACCESS.bus address of the PC87591x. When in Slave mode, the first seven bits received after a Start Condition are compared to this field (the first bit received is compared to bit 6, the next bit to bit 5 and so on until the last bit, which is compared to bit 0). If the address field matches the received data and SAEN in ACBnADDR register is set to 1, a match is declared.
7	**SAEN (Slave Address Enable).** When set (1), indicates that the ADDR field holds a valid address and enables the match of ADDR to an incoming address byte. When cleared, the ACB does not check for an address match.

ACB Control Register 2 (ACBnCTL2)

The ACBnCTL2 register enables/disables the module and determines the ACB clock rate. On reset and while the module is disabled (ENABLE in ACBnCTL2 register is set to 0), ACBnCTL2 is cleared (00_{16}).

Location: Channel 1 - 00 $FF6A_{16}$
Channel 2 - 00 $FFEA_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**SCLFRQ6-0**							**ENABLE**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**ENABLE.** When set, the ACB module is enabled. When the Enable bit is cleared, the ACB module is disabled, ACBnCTL1, ACBnST and ACBnCST are cleared and the clocks are halted.
7-1	**SCLFRQ6-0 (SCL Frequency bits 6 through 0).** This field, together with SCLFRQ8-7 in ACBCTL3 register, defines the SCL's period (low time and high time) when the PC87591x serves as a bus master. The clock low time and high time are defined as follows: t_{SCL} =4*SCLFRQ*t_{CLK} $t_{SCLl} = t_{SCLh}$ where t_{CLK} is the PC87591x clock cycle when in Active mode (see Section 7.6.3 on page 385). SCLFRQ may be programed to values in the range of 00 0001000_2 (8_{10}) through 11 1111111_2 (511_{10}). Values outside this range gives unpredictable results.

ACB Control Register 3 (ACBnCTL3)

The ACBnCTL3 register expands the clock pre-scaler field and enables the match to ARP addresses. ACBnCTL2 is cleared on reset (00_{16}).

Location: Channel 1 -00 $FF6E_{16}$
Channel 2- 00 $FFEE_{16}$ (Update for Multiple Instances)

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**ARPMEN**	**SCLFRQ8-7**	
Reset	0	0	0	0	0	0	0	0

Bit	Description
1-0	**SCLFRQ8-7 (SCL Frequency bits 8 and 7).** Extends SCLFRQ field and is concatenated with bits 0-6, which are part of ACBnCTL2 register. Detailed use of SCLFRQ is provided in the SCLFRQ6-0 description in ACBnCTL2.
2	**ARPMEN (ARP Match Enable).** When set, enables the matching of an incoming address byte to the SMBus ARP address ($110\ 0001_2$) while the ACB is in Slave mode. When cleared, the ACB does not respond to an ARP address.
7-3	**Reserved.**

4.13.9 Usage Hints

1. When the ACB is disabled, BB in ACBnCST register is cleared. After the ACB is enabled (by setting ENABLE in ACBnCTL2 register), the bus may be in the middle of a transaction with another device in systems with more than one master. This status is not reflected by BB.

   To prevent bus errors, the ACB must synchronize with the bus activity status before issuing a request to become the bus master for the first time. The software should check that there is no activity on the bus by checking the BB bit after the time-out period allowed by the bus.
2. When waking up from power-down before checking MATCH in ACBnCST register, check BUSY in the same register to make sure that the address transaction is completed.
3. The BB bit can help solve a deadlock in which two or more devices detect a usage conflict on the bus and both cease being bus masters at the same time. In this situation, the BB bits of both devices are active (because each “detects” another master currently performing a transaction, while in fact there is no transaction). This potentially causes the bus to stay locked until a device on the bus sends a Stop Condition (through STOP in ACBnCTL1 register).

   The BB bit allows the software to monitor bus usage so that it can detect whether the bus remains unused over a certain period of time while BB is set. It also avoids sending a STOP signal in the middle of the transaction of another device on the bus.
4. In some cases, the bus may get stuck with the SCL and/or SDA lines active, such as when an erroneous Start or Stop Condition occurs in the middle of a slave receive session.

   If the SCL line is stuck active, the module that holds the bus must release it.

   If the SDA line is stuck active, the sequence below releases the bus (Note**:** In normal cases, SCL may be toggled only by the bus master; this sequence is a recovery scheme which is an exception and should be only used if there is no other master on the bus):

   a. Disable and re-enable the module to set it for the Slave mode not addressed.
   b. Set START in ACBnCTL1 register to attempt to issue a Start Condition.
   c. Check if the SDA line is active (low) by reading TSDA in ACBnCST register. If it is active, issue a single SCL cycle by writing 1 to TGSCL in the same register. If it is not active, skip to step e.
   d. Check if MASTER in ACBnST register is set, which indicates that the Start Condition was sent. If it is not set, repeat step c and this step until the SDA is released.
   e. Clear BB. This enables START to be executed. Continue according to “Bus Idle Error Recovery” on page 200.

4.14 ANALOG COMPARATORS MONITOR (ACM)

The Analog Comparators Monitor (ACM) checks the voltage level of eight analog inputs and reports their values to the core.

The ACM can either measure the voltage of each input with 6-bit resolution or compare the level of all eight inputs with a programmable threshold.

An operation burst includes either the voltage measurement of all inputs or one comparison check of all inputs to a fixed threshold. The bursts may be triggered by a low-frequency clock for periodic operation.

4.14.1 Features

- Voltage measurement
  - Eight analog inputs
  - 6-bit resolution
  - 0V to V_{CC} Full Scale (FS) input voltage range
  - High-impedance inputs
- Comparison check
  - 6-bit threshold resolution
  - Simultaneous on all inputs
- Ratiometric measurement using V_{CC}
- Digital reading output
  - Eight buffers for voltage value
  - One buffer for comparison check
- 170 µs voltage measurement of all the inputs
- Polling- or interrupt-driven interface
- Power consumption
  - Zero current when disabled
  - Low average current in Idle mode

4.14.2 Functional Description

Inputs. The ACM has eight analog inputs with a voltage ranging from 0V to V_{CC} (FS), as shown in Figure 74.

The voltage level at each input is compared with a reference level, generated by a 6-bit D/A converter.

Operation. The ACM can be operated in one of three modes:

- Voltage Level Burst mode - The reference level generated by the D/A converter is ramped from 0V to FS by incrementing the value loaded into the converter. Whenever the ramp crosses the level of an input, its comparator toggles state, and the D/A loaded value is latched into the corresponding Voltage Level Buffer. At the end of the ramp, execution stops and a set of eight voltage level values (one per input) is available.
- Threshold Comparison Burst mode - A programmable value is loaded into the D/A converter, generating a constant threshold level. A bit is set for each input if the input’s voltage level is higher than the threshold. At the end of the single comparison, the Comparison Result register holds the status of the eight inputs.
- Low Power Threshold Comparison mode - The inputs are periodically compared with a fixed threshold. The comparison is triggered by a low rate signal from the TWD module (T0IN). In addition, an interrupt is generated if at least one input is below (or above) the threshold.

Timing. When the PC87591x is in Active or Active Executing Wait mode, ACM module timing is based on the core domain clock (CLK). When the PC87591x is in Idle mode, timing is based on the low frequency clock (LFCLK) to minimize power consumption.

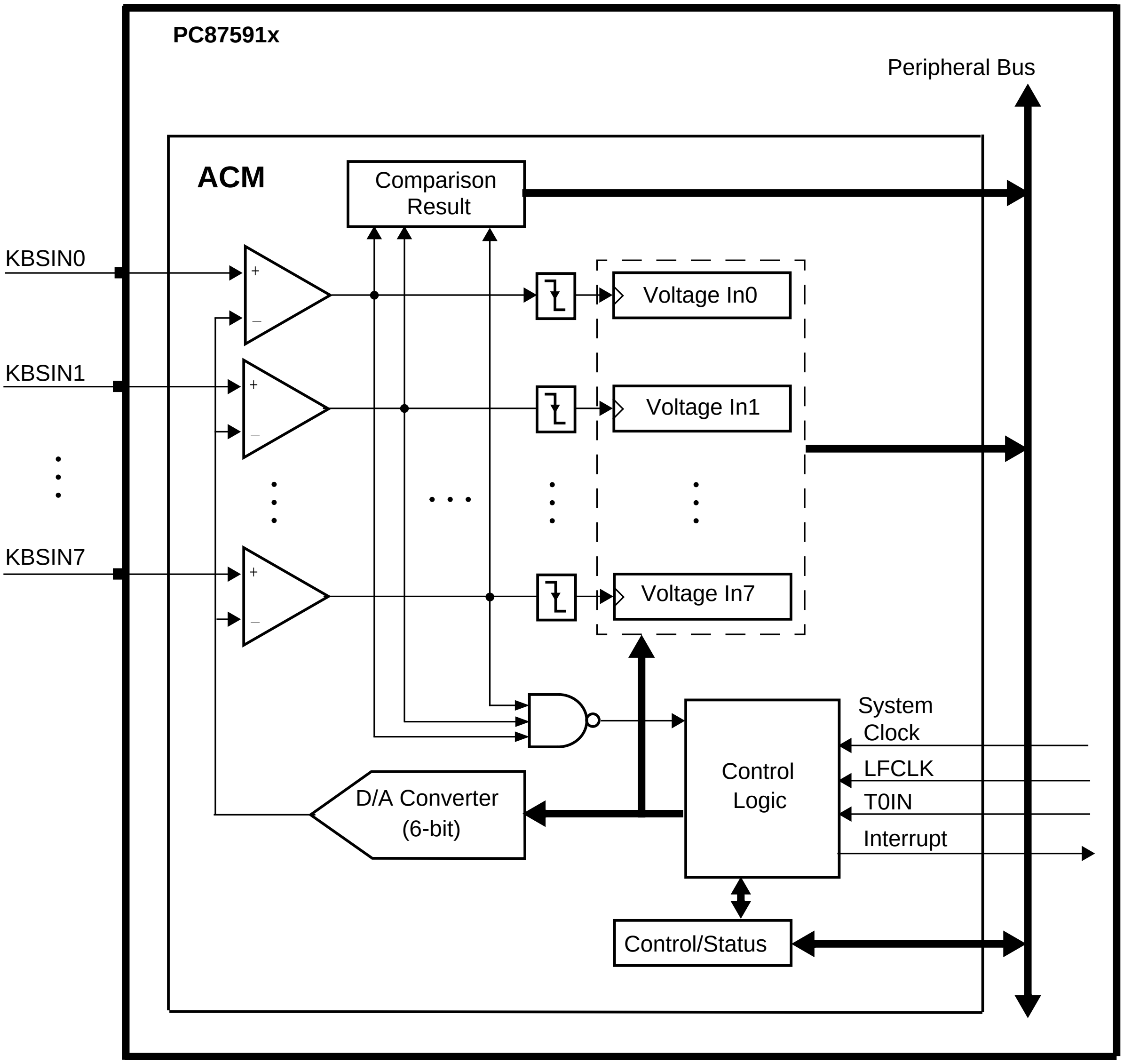


Figure 74. ACM Functional Diagram

4.14.3 Voltage Level

V_{CC} supply is the reference and full scale (FS) voltage for the D/A converter. This enables ratiometric voltage measurement (parts of FS), were V_{CC} is used to power an external voltage divider connected to the KBSINn inputs.

The 6-bit Voltage Level Data field (VOLTLVL) in VOLDATn (n = 0 to 7) registers and the Comparison Threshold Data field (THRSHD5-THRSHD0) in THRDAT register are converted to input voltage, according to input voltage ratio (to FS), as shown in Figure 75.

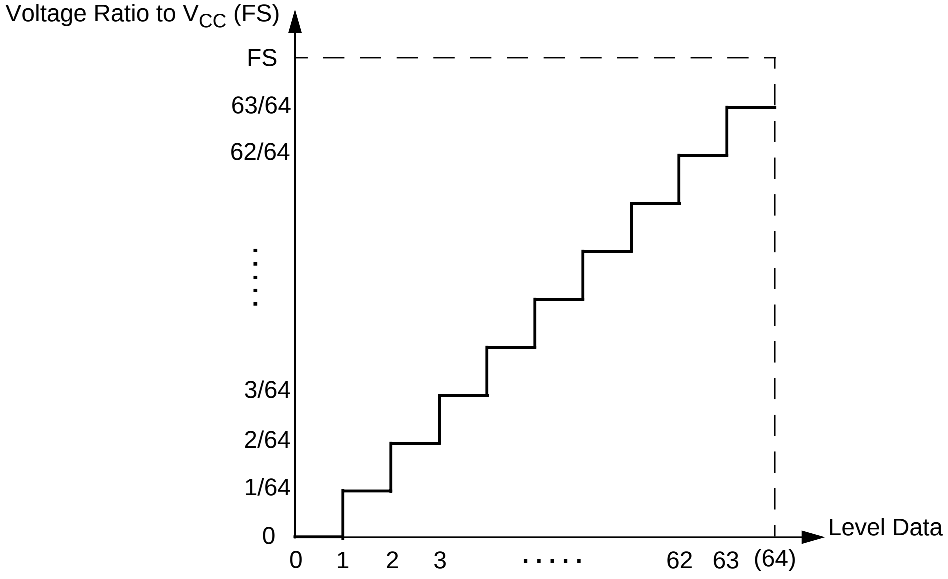


Figure 75. Level Data to Input Voltage Ratio Conversion

4.14.4 ACM Operation

Reset

The ACM is initialized by core domain reset (Section 3.2 on page 65 describes the PC87591x reset events).

Upon reset, the ACM is disabled, with all interrupt sources masked and the over/under threshold select set to “below”. All Event status bits are reset. The trigger division factor, as well as the data sampling delay, are set to their maximum value (for the slowest ACM operation speed). The comparison threshold is set to zero.

All control, configuration and status registers are reset to their default values, as indicated in Section 4.14.5 on page 212. Voltage Level Data Buffer registers and the Comparison Result register’s values are invalid until the first measurement occurs (on each of them).

Sampling Delay

This delay separates between the new value being loaded into the D/A converter and the data being sampled into the Voltage Level Data Buffer registers and the Comparison Result register. Its value should be longer than the total time required by the D/A output to settle (within 1/2 LSB of 6 bits) and the comparators to toggle their state.

The value set by SMPDLY field in ACMTIM register is expressed in terms of core CLK clock cycles. The maximum frequency at which each delay may be used is specified in the register’s description.

Initializing the ACM

The ACM must be initialized before it is enabled and used. The following operations should be done:

- Select the ACM Mode Control by setting ACMMOD field in ACMCNF register.
- Enable interrupts are required using the following bits of ACMCNF register:
  - Interrupt from End-of-Measurement Event Enable by setting INTEMEN bit.
  - Interrupt from Over/Under Threshold Event Enable by setting INTOUEN bit.
- Select Over or Under Threshold mode, using OVUNSEL bit in ACMCNF register.
- Set the data sampling delay using SMPDLY field in ACMTIM register.
- Select the low power trigger rate by setting T0DIV field in ACMTIM register.
- Select the Comparison Threshold Data for wake-up by setting THRSHD field in THRDAT register.

After initializations are done, the ACM may be enabled by writing 1 to START bit in ACMCTS register.

Note: Setting any of the above bits/fields during ACM operation may cause unpredictable results.

Embedded Controller Modules (Continued)

Interrupt Structure

The ACM Interrupt is generated to the core if an event from Table 26 becomes active. Since the events are enabled in different ACM operation modes, only one event may occur at a time.

Table 26. ACM Interrupt Structure

Event Flag	Register Mnemonic	Mask Bit	Register Mnemonic	Description
EOMEV	ACMCTS	INTEMEN	ACMCNF	End-of-Measurement event and associated interrupt enable
OVUNTHEV	ACMCTS	INTOUEN	ACMCNF	Over/Under Threshold event and associated interrupt enable

When an event flag and its related mask bit are set (enabled), the ACM Interrupt request is asserted. This is indicated by a high level of the ACM Interrupt signal.

The software must reset the event flag (or its mask bit) to de-assert the ACM Interrupt request.

All the event flags (EOMEV, EOCEV and OVUNTHEV) are cleared by writing 1 to START in ACMCTS register.

The ACM Interrupt is routed both to the ICU and to the MIWU as an ACMI signal (see Section 4.3 on page 98 and Section 4.4 on page 105).

ACM Operating Sequences

The ACM has three operating modes:

- Voltage Level Burst mode
- Threshold Comparison Burst mode
- Low-Power Threshold Comparison mode

After the ACM is properly initialized, one of these modes should be used. The operation sequences are as follows:

Voltage Level Burst Mode (ACMMOD = 01_2)

1. Start a new burst of eight voltage measurements by setting START bit in ACMCTS register to 1.
2. When End-of-Voltage level measurement burst is reached, software can detect the event by waiting for EOMEV in ACMCTS register to be set to 1.
3. Read measured voltage level - Input 0, for channel 0, by reading VOLTLVL in VOLDAT0 register.
4. Read measured voltage level - Input 1, for channel 1, by reading VOLTLVL in VOLDAT1 register.
5. Read measured voltage level - Input 2, for channel 2, by reading VOLTLVL in VOLDAT2 register.
6. Read measured voltage level - Input 3, for channel 3, by reading VOLTLVL in VOLDAT3 register.
7. Read measured voltage level - Input 4, for channel 4, by reading VOLTLVL in VOLDAT4 register.
8. Read measured voltage level - Input 5, for channel 5, by reading VOLTLVL in VOLDAT5 register.
9. Read measured voltage level - Input 6, for channel 6, by reading VOLTLVL in VOLDAT6 register.
10. Read measured voltage level - Input 7, for channel 7, by reading VOLTLVL in VOLDAT7 register.
11. Clear the event flag (release the ACM interrupt if enabled) by writing 1 to EOMEV in ACMCTS register.

Threshold Comparison Burst Mode -(ACMMOD = 10_2)

1. Set the desired threshold value in bits THRSHD(5-0) in THRDAT register.
2. Start a new burst of threshold comparison by setting START in ACMCTS register to 1.
3. When End-of-Threshold comparison burst is reached, software can detect the event by waiting for EOCEV in ACMCTS register to be set to 1.
4. Read comparison result for inputs 0 to 7 by reading bits CMPIN(0-7) in CMPRES register.
5. Clear the event flag by writing 1 to EOCEV in ACMCTS register.

Low-Power Threshold Comparison Mode (ACMMOD = 11_2)

1. Set the desired threshold value in bits THRSHD(5-0) in THRDAT register.
2. Start a new periodic, low-power threshold comparison by setting START in ACMCTS register to 1.
3. When there is an input under (or over) threshold event, software can detect the event by waiting for OVUNTHEV in ACMCTS register to be set to 1.
4. Read comparison result for inputs 0 to 7 by reading bits CMPIN(0-7) in CMPRES register.
5. Clear the event flag (release the ACM interrupt if enabled) by writing 1 to OVUNTHEV in ACMCTS register.

Polling Driven Operation. Valid measurement results may be retrieved by polling the event flags in ACMCTS register, EOMEV, EOCEV or OVUNTHEV, according to the selected operation mode. After reading the relevant data, clear the polled event flag by writing 1.

Interrupt Driven Operation. When receiving an active ACM interrupt request, the software should check the two event flags, EOMEV or OVUNTHEV, to identify the source of the interrupt. After reading the relevant data, clear (by setting to 1) the event flag that caused the interrupt. Clearing the flag releases the ACM interrupt request.

4.14.5 ACM Registers

The ACM control/status and data out register set interfaces with the core through the Peripheral bus.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

ACM Register Map

The ACM register set contains four control and status registers and nine data out registers.

Table 27. ACM Register Map

Mnemonic	Register Name	Type
ACMCTS	ACM Control and Status	Varies per bit
ACMCNF	ACM Configuration	R/W
ACMTIM	ACM Timing Control	R/W
THRDAT	Comparison Threshold Data	R/W
CMPRES	Comparison Result	RO
	Reserved	
VOLDAT0	Voltage Level Data - Input 0	RO
VOLDAT1	Voltage Level Data - Input 1	RO
VOLDAT2	Voltage Level Data - Input 2	RO
VOLDAT3	Voltage Level Data - Input 3	RO
VOLDAT4	Voltage Level Data - Input 4	RO
VOLDAT5	Voltage Level Data - Input 5	RO
VOLDAT6	Voltage Level Data - Input 6	RO
VOLDAT7	Voltage Level Data - Input 7	RO

ACM Control and Status Register (ACMCTS)

This register controls the operation and indicates the status of the ACM module. ACMCTS is cleared (00_{16}) on reset.

Location: 00 $FD40_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**				**OVUNTHEV**	**EOCEV**	**EOMEV**	**START**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	W	**START (ACM Operation Start).** Start trigger for the ACM operation in the mode selected by ACMMOD field in ACMCNF register. If the new ACMMOD value differs from the value currently in use, all event flags (EOMEV, EOCEV, OVUNTHEV) are cleared and operation in the new mode begins. Writing a 1 to the START bit without modifying the ACMMODE value is ignored by the module. 0: No effect - (default) 1: Operation start trigger - automatically returns to 0 after start
1	R/W1C	**EOMEV (End-of-Measurement Event).** End of voltage level measurement burst (for all eight inputs). The bit is set only when the ACM is in Voltage Level Burst mode (ACMMOD = 01_2 in ACMCNF register). 0: Measurement in progress, or **not** in Voltage Level Burst mode (default) 1: End of voltage level measurement burst
2	R/W1C	**EOCEV (End-of-Comparison Event).** End of threshold comparison in Threshold Comparison Burst mode (ACMMOD = 10_2 in ACMCNF register). 0: Comparison in progress, or **not** in Threshold Comparison Burst mode (default) 1: End of threshold comparison
3	R/W1C	**OVUNTHEV (Over/Under Threshold Event).** At least one input is above (or below) the voltage threshold value. The bit is set only when the ACM is in Low Power Threshold Comparison mode (ACMMOD = 11_2 in ACMCNF register). 0: No input above (or below) the threshold, or **not** in Low Power Threshold Comparison mode (default) 1: One or more inputs above (or below) the threshold
7-4		**Reserved.**

Embedded Controller Modules (Continued)

ACM Configuration Register (ACMCNF)

This register controls the configuration of the ACM module. ACMCNF is cleared (00_{16}) on reset.

Location: 00 $FD42_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**OVUNSEL**	**INTOUEN**	**INTEMEN**	**Reserved**	**ACMMOD**	
Reset	0	0	0	0	0	0	0	0

Bit	Description
1-0	**ACMMOD (ACM Mode Control).** Configures the operation mode of the ACM module. See “ACM Operating Sequences” on page 211. This value becomes the active mode after writing a 1 to START bit in ACMCTS register. **Bits** **1 0 Description** 0 0: ACM disabled (default) 0 1: Voltage Level Burst mode - single measurement of the voltage level for all the eight inputs 1 0: Threshold Comparison Burst mode - single comparison of all the eight inputs to the set threshold 1 1: Low-Power Threshold Comparison mode - periodic comparison of the inputs to a constant threshold (recommended for Idle PC87591x operation mode). Its operation is stopped by the selection of a different mode (ACM disabled = 002 is the recommended value).
2	**Reserved.**
3	**INTEMEN (Interrupt from End-of-Measurement Event Enable).** Enables generation of an ACM interrupt on an End of Voltage Measurement Burst event (EOMEV in ACMCTS register). 0: Disabled (default) 1: Enabled - ACM Interrupt from EOMEV
4	**INTOUEN (Interrupt from Over/Under Threshold Event Enable).** Enables generation of an ACM interrupt on at least one input above (or below) the threshold event (OVUNTHEV ACMCTS register). 0: Disabled (default) 1: Enabled - ACM Interrupt from OVUNTHEV
5	**OVUNSEL (Over or Under Threshold Select).** “Any Over” or “Any Under” logic selection for OVUNTHEV bit in ACMCTS register. 0: At least one input below threshold sets OVUNTHEV=1 (default) 1: At least one input above threshold sets OVUNTHEV=1
7-6	**Reserved.**

ACM Timing Control Register (ACMTIM)

This register controls the sampling delay for the voltage level and compare out data; it also controls the division of the trigger signal. ACMTIM is set to 37_{16} on reset.

Location: 00 $FD44_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**T0DIV**		**Reserved**	**SMPDLY**		
Reset	0	0	1	1	0	1	1	1

Bit	Description
2-0	**SMPDLY (Data Sampling Delay).** Compensates for the settling time of the D/A converter and the input comparators. To calculate the required delay value, see “Sampling Delay” on page 210. **Bits 2 1 0 — Settling Period[1] — Compare Period[1] — Max Core Freq[2]** 0 0 0: 1 10 5.5MHz 0 0 1: 1 15 8.5 MHz 0 1 0: 2 20 11.5 MHz 0 1 1: 2 25 14.2 MHz 1 0 0: 2 30 16.5 MHz 1 0 1: 2 35 20.0 MHz 1 1 0: 3 45 20.0 MHz 1 1 1: 3 55 20.0 MHz (default) 1. The Settling and Compare periods are in clock cycles. 2. Conversion step time is equal to Settling Period + Compare Period
3	**Reserved.**
5-4	**T0DIV (Trigger Signal Division Factor).** Controls the division of the T0IN trigger signal in Low Power Threshold Comparison mode. A higher division factor gives a lower power consumption in Idle operation mode (see Section 4.17 on page 235). T0IN frequency is separately set (see Section 4.10 on page 164). **Bits 5 4 — T0In Division Factor** 0 0: 16 0 1: 32 1 0: 64 1 1: 128 (default)
7-6	**Reserved.**

Embedded Controller Modules (Continued)

Comparison Threshold Data Register (THRDAT)

This register holds the data used by the D/A converter in the threshold comparison modes. ACMCNF is cleared (00_{16}) on reset.

Location: 00 $FD46_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**THRSHD5-THRSHD0**					
Reset	0	0	0	0	0	0	0	0

Bit	Description
5-0	**THRSHD5-THRSHD0 (Comparison Threshold Data).** Data to be used by the D/A converter in Threshold Comparison Burst and Low Power Threshold Comparison modes. All eight inputs are compared with the resulting voltage value. Range: 0 to 63 (0 to 63/64 * V_{CC}); 6-bit, unsigned value with 1 LSB = V_{CC}/64 (the value at reset is 0).
7-6	**Reserved.**

Comparison Result Register (CMPRES)

This register contains the result of the input comparison with the threshold.

Location: 00 $FD48_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**CMPIN7**	**CMPIN6**	**CMPIN5**	**CMPIN4**	**CMPIN3**	**CMPIN2**	**CMPIN1**	**CMPIN0**

Bit	Description
0	**CMPIN0 (Comparison Result for Input 0).** Result of KBSIN0 comparison with the voltage threshold value. Relevant only in Threshold Comparison Burst mode (ACMMOD = 10_2 in ACMCNF register) and Low Power Threshold Comparison mode (ACMMOD = 11_2). The value at reset is undefined. 0: Input below threshold 1: Input above threshold
1	**CMPIN1 (Comparison Result for Input 1).** Same as CMPIN0 for KBSIN1.
2	**CMPIN2 (Comparison Result for Input 2).** Same as CMPIN0 for KBSIN2.
3	**CMPIN3 (Comparison Result for Input 3).** Same as CMPIN0 for KBSIN3.
4	**CMPIN4 (Comparison Result for Input 4).** Same as CMPIN0 for KBSIN4.
5	**CMPIN5 (Comparison Result for Input 5).** Same as CMPIN0 for KBSIN5.
6	**CMPIN6 (Comparison Result for Input 6).** Same as CMPIN0 for KBSIN6.
7	**CMPIN7 (Comparison Result for Input 7).** Same as CMPIN0 for KBSIN7.

Voltage Level Data Buffer - Input 0 through 7 (VOLDAT0-7)

This register holds the voltage level measurement result for inputs 0 through 7.

Location: 00 $FD50_{16}$ - Input 0
00 $FD52_{16}$ - Input 1
00 $FD54_{16}$ - Input 2
00 $FD56_{16}$ - Input 3
00 $FD58_{16}$ - Input 4
00 $FD5A_{16}$ - Input 5
00 $FD5C_{16}$ - Input 6
00 $FD5E_{16}$ - Input 7

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**VOLTLVL**					

Bit	Description
5-0	**VOLTLVL (Voltage Level Data).** Voltage level of the respective input i (KBSINi for i 0 through 7), while operating in Voltage Level Burst mode (ACMMOD = 01_2 in ACMCNF register). The readout is an unsigned value with 1 LSB = $V_{CC}/64$; the input voltage is calculated as: V_{IN} = VOLTVL * (V_{CC}/64).
7-6	**Reserved.**

4.14.6 Usage Hints

Voltage Level Burst

The total duration of the Voltage Level Burst measurement is a function of the system clock frequency (see Section 4.18 on page 239) and the data sampling delay, selected by SMPDLY field in ACMTIM register. The values in Table 28 should only be used to evaluate the maximum duration of the measurement. Actual duration may be shorter, depending on the input level value. The software should use either EOMEV flag polling or the ACM interrupt, as described in Section 4.14.4 on page 210.

Table 28. Voltage Level Burst Duration

System Clock Frequency (MHz)	SMPDLY Setting	Maximum Burst Duration (μs)
4	000	176
10	010	140
20	101	118

4.15 ON-CHIP RAM

The PC87591x provides up to 4096-byte on-chip RAM array (2048-byte in PC87591E and PC87591L devices).

A 16-bit wide data bus links the core and the system RAM array. The system RAM can be byte or word accessed. Each system RAM read or write operation is one cycle long and does not include any wait states. See Section 1.5.1 on page 33 for system RAM memory map.

4.16 ON-CHIP FLASH

The PC87591x has an on-chip flash memory device. This flash includes a Main Block, which serves as an on-chip Base Memory, and an Information Block, which is used for device-specific parameters and user-defined data.

The Main Block (Base Memory) is used to store program and constant data of the core and/or host. In IRE and OBD environments, the on-chip flash is used as (on-chip) Base Memory. In DEV environment, for easy and efficient code development, the on-chip Base Memory is replaced by off-chip SRAM, which is referred to as off-chip Base Memory. See Section 1.4 on page 29 for more details about the memory emulation concept.

Note that in the PC87591L, the on-chip base memory is set to function as a ROM, with the associated limitations on write and erase operations.

4.16.1 Features

- Two flash memory blocks:
  - — Main Block (Base Memory)
  - — Information Block
- Designated areas for:
  - — Core boot code
  - — Host boot code
  - — Factory Parameters
  - — Protection information
- Supports flash operations:
  - — Read (byte/word)
  - — Write (byte/word)
  - — Erase (page)
- Several access paths for flash operations:
  - — Core
  - — Host
  - — JTAG
  - — Parallel Interface
- Flash contents protection:
  - — Different levels for each access path
  - — Different levels for each memory block
- Flash Timing Control for:
  - — Command-based automatic control
  - — Busy and error indicators
- Low Power operation

Embedded Controller Modules (Continued)

4.16.2 Functional Description

The flash is byte accessible for reads and writes; it is erased in pages. Its organization is optimized for the PC87591x target application and security needs. Table 29 summarizes the various flash size parameters.

Table 29. PC87591x Flash Organization

Main Block (Base Memory)				
Size[1]	**Organization[2]**	**Page Size**	**Access Width**	**Access Time Control**
64 Kbytes[3]	32Kx16 (Section 0)	512 bytes	byte/word	SZCFG1
64 Kbytes[4]	32Kx16 (Section 1)	512 bytes	byte/word	SZCFG1

Information Block		
Size	**Page Size**	**Access Width**
64x16 (Section 0)	128 bytes	byte/word
64x16 (Section 1)	128 bytes	byte/word

1. See “Accessing Base Memory” on page 35 for the mapping of the memory blocks in the core address space.
2. The memory width should be used for correct configuration of the BIU to enable cycle-by-cycle compatibility between on-chip and off-chip Base Memory.
3. Block Size is 4 Kbytes only in PC87591L devices.
4. This Main Block is available in PC87591S devices only.

Flash Protection

The flash is partitioned into separate areas to protect the contents. Section 4.16.6 on page 226 provides details of the Main Block and Information Block protection mechanisms, respectively. The main protected areas are:

- Main Block
  - — Core Boot Block - an area dedicated for the core boot code and protected from corruption by the CR or host.
  - — Host Boot Block - an area dedicated for host boot code and protected from corruption by the host.
  - — Other parts - optionally protected by software
- Information Block
  - — Factory Parameters - hold various parameters and calibration information defined during PC87591x production; protected from all alterations.
  - — Protection Word - information about the flash data protection that may restrict accessing, writing and/or erasing all or parts of the flash.
  - — Other parts - software-controlled read and write protection.

Embedded Controller Modules (Continued)

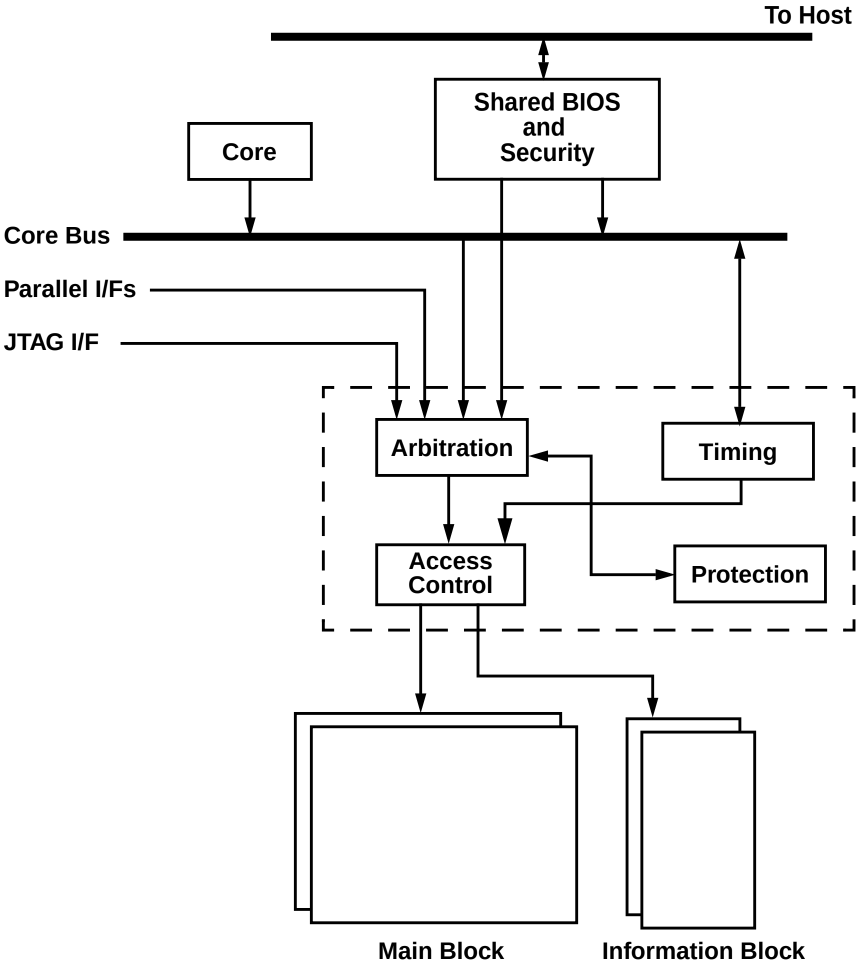


Figure 76. On-Chip Flash Block Diagram

Access Paths

The flash memory can be read, written and erased through several access paths with different levels of protection for the flash contents. Each of these paths can be operational in only some of the environments (i.e., IRE, OBD, DEV or PROG). The flash access paths are:

- Host interface
- CR16B core
- JTAG flash access
- Parallel interface

The host and core use core bus read write transactions to access the flash. The JTAG and Parallel Interfaces have dedicated access paths. An arbitration scheme selects the active interface that accesses the flash at any given time. The host and core accesses may interleave. JTAG flash access is enabled in OBD and ENV environments when the appropriate enable bits are set (see “Flash Interface Functional Description” on page 251 for the access enabling scheme). The Parallel interface is enabled in PROG environment only.

Table 30 summarizes the flash information accessibility, the restrictions that apply in the various environments and the protection bit settings for each of the memory areas. Details of read, write and erase access are provided in Section 4.16.4 on page 224. The following notation is used:

NA	-	Not accessible
RO	-	Read only
RW	-	Read, Write
RWE	-	Read, Write and Erase
RWe	-	Read, Write and a limited Erase (i.e., entire flash erase only).

Embedded Controller Modules (Continued)

Table 30. PC87591x Flash Read/Write/Erase Permission Matrix

Memory Area	Interface	Host[1]	Core	JTAG[2]	Parallel I/F[2]
	Environment	**IRE, OBD & DEV**		**OBD & DEV**	**PROG**
Main Block (Base Memory)	Core Boot Block	NA[3]	RO[3]	RWE	RWE
	Host Boot Block	RO[3]	RWE[3]	RWE	RWE
	Other	RWE[3]	RWE[3]	RWE	RWE
Information Block	Factory Parameters	NA	RO	RO	RWE
	Protection Word	NA	RO	RWe	RWe

1. Additional read and/or write protection may apply according to the setting of the software controlled protection mechanism.
2. Read and write operations are enabled only if the Flash Access Enable bit (FLAEN in FL_CT_ST register) is 1 (see page 262). When the Flash Access Enable bit is cleared, only a special erase may be performed.
3. In DEV Environment, the Base Memory is implemented using an off-chip SRAM, thus any access by the core and/or host is routed to it. Host access limitations apply in DEV environment. The core is allowed full read and write access to the off-chip Base Memory.

Flash Page Organization and Erase Options

A flash erase operation changes all bits in the erased memory area to 1s. The minimum flash portion that can be erased is a page. A byte address can be written once before the next erase.

The erase process includes verification. During an erase, a busy indicator is turned on. On completion, the busy indicator is turned off, an error indicator is turned on if an error is detected and the on-chip flash returns to standby.

A flash program operation changes bits that are 1 to 0. A flash program operation may performed on bytes or words.

The PC87591x flash is organized in rows, grouped in pages and partitioned into sections as follows:

Table 31. PC87591x Section and Page Organization

Block	Section Name	Address Range	Page Size
Main Block	Section 0[1]	$00\ 0000_{16}$ - $00\ \mathrm{FFFF}_{16}$	512 bytes
	Section 1[2]	$01\ 0000_{16}$ - $01\ \mathrm{FFFF}_{16}$	512 bytes
Information Block	Section 0	000_{16} - $07\mathrm{F}_{16}$	128 bytes
	Section 1	080_{16} - $0\mathrm{FF}_{16}$	128 bytes

1. In the PC87591L, the main block is in addresses $00\ 0000_{16}$ - $000\mathrm{FFF}_{16}$.
2. Available only in the PC87591S.

The PC87591x provides three erase operations:

- Page Erase - erase either one page of the flash Main Block or Information Block.
- Section Erase - erase an entire section.
- Special Erase - erase the entire flash (except factory parameters).

Notes:

- The high voltage required for flash operation is generated on-chip from V_{CC}.
- Flash endurance: 10,000 write-erase cycles (minimum).
  The PC87591L flash is guaranteed for five write-erase cycles only.
- Data retention: more than 100 years at room temperature.

Flash Read, Write and Erase Time

The flash read access time affects the performance of the core. It is therefore recommended to configure the interface to achieve the maximum performance allowed for the operation frequency. Table 32 lists the access time for each section of the flash.

Table 32. Access Time for Flash Sections

Block Name	Section Number	Max Frequency[1,2]	Write Time	Erase Time
Main Block	Sections 0 and 1	F_{FMAX} @ fast read	Approx. 20 µs, as defined in Table 33	Approx. 10 ms, as defined in Table 33
Information Block	Sections 0 and 1	F_{FMAX} @ 3 clock cycles		Approx. 10 ms, as defined in Table 33

[1] See Section 7.5 on page 381 for the values of these parameters.

[2] The Zone Configuration register SZCFG1 should be used to set the access speed, based on the current operation speed and minimum required access time.

Notes:

- Write cycle timing is not cycle-by-cycle emulated for off-chip Base Memory (SRAM).
- For fast read operation, the respective BIU zone should be configured for fast read.
- For n wait states, the respective BIU zone should be configured for normal read with n wait states. This provides an n+2 cycles read throughput.
- Sections 0 and 1 are also referred to as the Fast zone.

The Flash Timing Control, included as part of the PC87591x on-chip flash, controls the access time for read, write and erase operations of the flash memory. This simplifies the handling and use of the flash by the user. To enable operation with various clock rates the Flash Timing Control includes a set of registers that enable the core to set the parameters. For more details on the Flash Timing Control, see Section 4.16.3 on page 222.

4.16.3 Flash Interface States and Timing Control

The Flash Timing Control automates six states: Standby, Read, Program, Page Erase, Section Erase and Special Erase. The widths of pulses used in flash operation are specified in registers and should be set by software based on the operation frequency of the core clock.

- Standby State

  The flash does not perform a read, program or erase operation. This is the default state when no operation is performed; current consumption of the flash is therefore minimal.

- Read State

  The flash performs a read operation from either the Main Block or the Information Block. Host or core reads through the core bus are extended to the required length based on the BIU SZCGF1settings. JTAG interface access uses a busy flag to indicate that the read operation is in progress. When the read is completed, the flash returns to Standby state.

- Program State

  The flash programs a byte or word in either the Main Block or Information Block. This state is entered by a core or host write (with the Program control bit set) or by a JTAG program command. While in this state, the busy indicator is set. When the program operation is completed, if the verify is enabled, the data is verified and the error bits are updated accordingly. During Program state, reset is ignored. During programing of the flash, additional writes to the flash are ignored and reads are delayed until the flash page programing is completed.

- Page Erase State

  The Page Erase state is entered by a core or host write or by a JTAG command when the Page Erase bit is set. An address within the page specifies which page is to be erased. While in this state, the busy indicator is set. When the erase operation is completed and the verification is enabled, the data is verified and the error bits are updated accordingly. During Page Erase state, reset is ignored, additional writes to the flash are ignored and reads are delayed until the flash page erase is completed.

- Section Erase State

  The Section Erase state is entered by a core write or by a JTAG command when the Section Erase bit is set. An address within the section specifies which section is to be erased. While in this state, the busy indicator is set. When the erase operation is completed, the data is verified and the error bits are updated accordingly. During Section Erase state, reset is ignored, additional writes to the flash are ignored and reads are delayed until the flash section erase is completed.

Embedded Controller Modules (Continued)

A write to an address within a Main Block section results in an erase of the Main Block section. A write to an address within an Information Block section results in an erase of the Information Block section and its corresponding Main Block section (e.g., Main Block section 0 and Information Block section 0).

- Special Erase State

  The Special Erase state is entered by a command from JTAG. Special erase enables erasing the entire flash, except for the factory parameters. It is the only operation on the flash enabled for the JTAG when the flash protection field is set to Protect. While in this state, the busy indicator is set. When the erase operation is completed, the data is verified and the error bits are updated accordingly. During Special Erase state, reset is ignored. The protected data information is always the last to be erased.

Both program and erase can be initiated by the host, core or JTAG interface. Note that no instruction fetch or data read/write can be performed from/to the on-chip flash while it is being programed or erased.

While the JTAG Flash interface is enabled, the Debugger interface generates a reset, which causes the core, the Host interface circuit and the core bus to be held in reset.

Flash Operation Timing Control

The Information Block read, the program and erase timings are controlled by internal timers. (The read timing from the Main Block is controlled by the BIU configuration). The timers default on Power-Up reset to operate at a core frequency of 4 MHz. Software may change this for operation with other core clock frequencies.

Section 4.16.7 on page 228 lists the timing control registers. Table 33 provides the values that should be programed into the control registers for each core domain clock frequency. It is the responsibility of the firmware to make sure that the timing control registers are set according to the current core domain clock frequency before any flash erase or flash program is allowed.

Table 33. Control Registers Values per Core Domain Clock

Core Domain Clock	FLPSLR	FLSTART	FLTRAN	FLPROG	FLPERASE	FLSERASE	FLEND	FLSEND	FLRCV
[MHz]	[hex]	[hex]	[hex]	[hex]	[hex]	[hex]	[hex]	[hex]	[hex]
4	0	14	28	A	9	9	14	32	4
5	0	19	32	C	C	C	19	3E	5
6	0	1E	3C	F	E	E	1E	4B	6
7	0	23	46	11	11	11	23	57	7
8	0	28	50	14	13	13	28	64	8
9	0	2D	5A	16	16	16	2D	70	9
10	1	19	32	C	C	C	19	3E	5
12	1	1E	3C	F	E	E	1E	4B	6
20	3	19	32	C	C	C	19	3E	5

Reset

While the on-chip flash is in Read state, driving reset aborts the current operation. Flash program and Erase operations continue to completion and the reset is extended accordingly. If reset is applied during Program or Erase state, the contents of memory are not guaranteed (i.e., the program or erase may fail).

After reset, the on-chip flash defaults to Standby state.

When JTAG access to flash is disabled, the Flash interface is reset by all reset events. When the JTAG access to flash is enabled, the PC87591x is reset, but the Flash interface allows flash access after all of the following have occurred:

- The core clocks are stable in their default setting.
- The protection information has been read.
- Default values have been loaded into the Flash Interface registers.

On exiting from “JTAG Access to Flash” mode, the Flash Interface module is reset again (i.e., protection information is read and default values loaded into the Flash Interface registers)

Pipelined Programing Cycles

Pipelined programing uses a word-wide double-buffer (for Main Block accesses) or a byte-wide double-buffer (for Information Block accesses). Accesses should be word-wide or byte-wide, correspondingly; otherwise, results are undefined.

When the double-buffer is empty, the pipelined programing is initiated on a valid byte/word write to the on-chip flash. The address of the transaction is the starting address. The Flash Timing Control automatically increments the address for the next write to point to the next byte/word.

Pipelined programing is completed when the double-buffer is empty or on reaching the highest address within the row. On completion, the busy indicator is turned off, any data in the double-buffer is discarded, an error indicator is turned on if an error is detected, and the on-chip flash returns to standby.

Results are undefined when Main Block and Information Block accesses are interleaved while the double-buffer is not empty.

Low Power Read Operation Mode

The PC87591x supports low power read operation for the on-chip flash. When this is enabled (see “Flash Memory Control Register (FLCR)” on page 230), single core bus cycle read accesses consume less power. Low power operation may be enabled only when the core clock is below the following limit:

- Below F_{MAX} for Sections 0 and 1 controlled by LOWPWRF bit

4.16.4 Flash Access and Control

The flash may be accessed for read, program and erase of data in either the Main Block or Information Block. This section describes the activities required to enable any of the operations from each of the access paths (Core, Host, JTAG).

Before an access path to the flash may be used, the PC87591x must be in a state that enables that path:

- Core and Host Interface - Active in IRE, OBD or DEV environment (Information Block only for DEV environment). For host write operations, the host must gain control of the flash (HLOCK bit in SMCCST register must be set). Any access (read or write) protection may override the requested access.
- JTAG Interface - OBD or DEV environments with the JTAG Flash interface enabled.

Section 4.19 on page 248 provides details of the JTAG flash access. Section 4.16.5 on page 226 provides details of the PROG environment bus. For all interfaces, read, program and erase protection flags may prevent the operations described from being completed.

Note that during Erase and Program of the flash, an access to the flash is handled as follows:

- Core read is delayed to the end of the read.
- Core write to the Main Block or Information Block is ignored.
- Core write to IBAI register is ignored.
- Host reads or writes are ignored by the Flash interface and restarted by the host once the flash program/erase is completed.

If core activity is required during flash program or erase, make sure it executes from RAM or external memory before the flash erase/programing begins.

Main Block Read

When the flash is not busy, read operations return the data in the accessed on-chip flash address. Either byte or word may be read.

The SZCFG1 register defines the number of core bus cycles used to access Base Memory (see Section 4.1.11 on page 86 and “Static Zone Configuration Register (SZCFGn)” on page 85 for further details). The number of cycles that each access requires, for each one of the zones, is defined in “Flash Read, Write and Erase Time” on page 222.

If the BIU configuration is identical in all environments, the access time to the on-chip flash Main Block and off-chip Base Memory, for read operations, is the same. This allows cycle-by-cycle compatibility in IRE, OBD and DEV operating environments.

Information Block Read

Information Block data is read using an indirect access scheme. This information may be read by the core, through the JTAG Interface or the Parallel Interface.

The core accesses the Information Block using indirect byte/word read access. To read from the Information Block:

- The byte or word address must be stored in IBAI register (note that Information Block addresses are mapped from 000 as shown in Table 31 on page 221).; word addresses must be even.
- Data byte/words must be read using a byte/word read operation from BD register.

The JTAG performs a read operation with an Information Block address attribute set. Only word read is allowed.

Embedded Controller Modules (Continued)

Main Block Page Erase

Pages in the Main Block may be erased one at a time. The size of the pages are defined in Table 30 on page 221 and "Flash Read, Write and Erase Time" on page 222. To enable erasing by the core or host, the page must be enabled for write.

1. Verify that the flash is not busy.
2. Prevent access to the flash while erase is in progress, if required.
3. Set the page erase bit.
4. Write to an address in the desired page.
5. Wait until the busy bit is cleared and check the error flags to see if the erase was successful.
6. To erase additional pages, repeat steps 3 through 5.
7. Clear the page erase bit to prevent accidental erase operations.

To erase a page using the JTAG interface, see Section 4.19.4 on page 251.

Information Block Page Erase

Pages in the Information Block may be erased one at a time. The size of the pages are defined in Table 30 on page 221 and Table 31 on page 221. Note that the factory parameters and the protection information may not be updated by the core.

To erase a page using the JTAG interface, see Section 4.19.4 on page 251. Note that the Factory Parameters page may not be erased by the JTAG interface.

Main Block Section Erase

Sections in the Main Block may be erased with a single erase operation. The size of the sections are defined in Table 30 on page 221 and Table 31 on page 221. To enable an erase by the core, the entire section must be enabled for write. Note that Host-initiated section erases are not allowed.

1. Verify that the flash is not busy.
2. Prevent access to the flash while erase is in progress, if required.
3. Set the section erase bit in the control register (SMER bit in FLCR register).
4. Write to an address in the desired section.
5. Wait until the busy bit is cleared and check the error flags to see if the erase was successful.
6. To erase additional sections, repeat steps 3 through 5.
7. Clear the section erase bit (SMER bit in FLCR register) to prevent accidental erase operation.

To erase a page using the JTAG interface, see Section 4.19.4 on page 251.

Main and Information Block Section Erase

Sections in the Main Block and Information Block may be erased simultaneously using a single erase operation. The size of the sections are defined in Table 30 on page 221 and Table 31 on page 221. To erase a section using the JTAG interface, see Section 4.19.4 on page 251.

Special Erase

This operation is enabled only for the JTAG interface. If the Flash Contents Protection bit is set, this operation should be done to enable any other access to the flash (or to enable the Debugger interface functions).

To perform a special erase using the JTAG interface, see Section 4.19.4 on page 251.

Program the Main Block

The program operation enables changing bits with a value of 1 to 0. Each byte address of the Main Block may be programed once before being erased. If required, perform an erase operation prior to the program operation. To program a byte in the Main Block:

1. Verify that the flash is not busy.
2. Prevent access to the flash while the program operation is in progress, if required.
3. Set the program bit in the control register (PE bit in FLCR register).
4. Write to the byte or word that needs to be programed.
5. Wait until the busy bit is cleared and check the error flags to see if the program operation was successful.
6. To program additional bytes/words, repeat steps 3 through 5.

To program a byte or word using the JTAG interface, see Section 4.19.4 on page 251.

Program a Page in the Information Block

The program operation enables changing bits with a value of 1 to 0. Each byte address of the Information Block may be programed once before being erased. If required, perform an erase operation prior to the program operation.

To program a word using the JTAG interface, see Section 4.19.4 on page 251.

4.16.5 Parallel Flash Interface

The Parallel Flash interface used in PROG environment allows the PC87591x to be tested and programed quickly with Automated Test Equipment (ATE) and programers in the OEM's manufacturing flow.

For details about this interface, contact a National Semiconductor sales representative.

4.16.6 Flash Protection

The Main Block can be accessed by the core, host, JTAG interface and Parallel interface. The Information Block may be accessed by the core, JTAG interface and Parallel interface. A protection mechanism that combines hardware and software controlled elements is provided. The protection level depends on the access path used and the contents of the 'protection word'.

Protection Word

The protection word is stored in address $0FE_{16}$ in the flash Information Block. It is constructed as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Res**	**RST2EN**		**Reserved**			**Flash Contents Protect**			**Force MBTA Zero**	**RTC Lock Default**	**Host Boot Block**	**Core Boot Block**			

Bit	Description
3-0	**Core Boot Block.** Defines the size of the Core Boot Block The boot block starts at address 00 0000 and ends at the address specified by this field. **Bits** **3 2 1 0 Description** 1 1 1 1: Flash is erased. When in IRE and OBD environments, the core is kept in reset (default). 1 1 1 0: 4K 1 1 0 1: 8K 1 1 0 0: 16K 1 0 1 1: 24K 1 0 1 0: 32K 1 0 0 1: 40K 1 0 0 0: 48K 0 1 1 1: 56K 0 1 1 0: 80K 0 1 0 1: 96K 0 1 0 0: 112K 0 0 1 1: 128K Other: Reserved
4	**Host Boot Block.** Defines the use and the size of a Host Boot Block area. The Host Boot Block is always at the end of the on-chip flash memory. Protection is provided by the Shared Memory function.[1] 0: A 64 Kbyte Host Boot Block is specified 1: No Host Boot Block is available (default)

Embedded Controller Modules (Continued)

Bit	Description
5	**RTC Lock Default.** Defines the reset value of the Lock RTC Host Access (LKRTCHA) bit in the Lock SuperI/O Host Access register (LKSIOHA). This enables preventing host access to the RTC at any time when the RTC is used by security applications running on the core. 0: RTC Lock is disabled on reset (LKRTCHA=0) 1: RTC Lock is enabled on reset (LKRTCHA=1) (default)
6	**Force MBTA Zero.** Enables the setting of the host boot block location to either end of on-chip flash or to the expansion memory[1]. See “Shared Memory Core Top Address Register (SMCTA)” on page 308. 0: MBSD field in SMCTA register is reset to the implemented on-chip flash size 1: MBSD field in SMCTA register is reset to 0000_{16} (default)
9-7	**Flash Contents Protect.** Enables protecting the contents of the on-chip flash from probing through the JTAG interface or Parallel interface. **Bits** **9 8 7 Description** 1 1 1: Flash access is enabled for both read and write Other: Flash access is protected. In any environment other than IRE, the device remains reset. In OBD and DEV environments, only special erase through the JTAG interface is enabled.
12-10	**Reserved.**
14-13	**RST2EN ($\overline{\text{RESET2}}$ Enable).** Controls the use of the $\overline{\text{RESET2}}$ alternate function. The following options are supported in the PC87591x: **Bits** **15 14 Description** 0 0: Reserved 0 1: $\overline{\text{RESET2}}$ is enabled on the IOPB7/$\overline{\text{RING}}$/$\overline{\text{PFAIL}}$/$\overline{\text{RESET2}}$ pin 1 0: $\overline{\text{RESET2}}$ is enabled on the IOPD2/EXWINT24/$\overline{\text{RESET2}}$ pin 1 1: $\overline{\text{RESET2}}$ input is disabled, and $\overline{\text{RESET2}}$ events will not be generated (default)
15	**Reserved.**

[1] When expansion memory is used for a shared BIOS implementation, set the Force MBTA Zero bit to 0, do not clear the Host Boot Block bit, and use a flash device with a boot block as expansion memory.

The protection word is read by hardware from the Information Block during the internal reset process. This information is used to set the host access restrictions described in Section 5.3.5 on page 301, the core access restrictions defined in this section, and the JTAG access protection described in OBD and PROG environment access restrictions; see Table 30 on page 221.

There is no protection over the core’s reading the flash contents. Writes to a write-protected page are ignored. When enabled, a core interrupt may be generated on write-protect violations.

Core-Initiated Accesses

There is no read protection on core-initiated read accesses. Write protection is provided with default settings and firmware override as follows:

- Main memory block - per 8 Kbyte block
- Information Block Factory parameters and protection word are always write protected

This protection is defined using FCWP0 register. All blocks are defined as write protected on reset. The blocks within the Core Boot Block and Host Boot Block are read only (RO) to prevent accidental alteration of their contents by the core.

Special Erase state is not supported for core-initiated accesses.

Host-Initiated Accesses

Host access to the flash Main Block is controlled by the information in the protection word and is controlled and further restricted by the core. Section 5.3.5 on page 301 details the host access limitations. The host may not access the Information Block.

Embedded Controller Modules (Continued)

JTAG-Initiated Accesses

JTAG-initiated accesses for both read and write is restricted based on the value held in the Flash Contents Protect field of the protection word.

When any bit in the Flash Contents Protect field is 0, no reads or writes are allowed. In OBD or DEV environment, the device remains reset until a special erase is performed. The only operation allowed in this case is a special erase of the flash contents.

When all bits in the Flash Contents Protect field are 1, all flash operations are allowed. The entire flash may be read, and except for the Factory Parameters page Of the Information Block, the entire flash may be erased and programed.

Parallel Flash Interface-Initiated Accesses

Parallel Flash interface-initiated accesses are subject to the same protection scheme as JTAG-initiated accesses. When any bit in the Flash Contents Protect field is 0, only special erase operations are allowed and all other flash access operations are ignored.

See also Section 4.16.5 on page 226.

4.16.7 Flash Interface Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Table 34. Flash Interface Register Map

Mnemonic	Register Name	Type
IBAI	Information Block Access Index	R/W
IBD	Information Block Data	R/W
FCWP0	Flash Core Write Protect	R/W
FLCR	Flash Memory Control	R/W
FLSR	Flash Memory Status	R/W1C
FLPSLR	Flash Memory Pre-Scaler	R/W
FLSTART	Flash Memory Start Time	R/W
FLTRAN	Flash Memory Transition Time	R/W
FLPROG	Flash Memory Program Time	R/W
FLPERASE	Flash Memory Page Erase Time	R/W
FLSERASE	Flash Memory Section Erase Time	R/W
FLEND	Flash Memory End Time	R/W
FLSEND	Flash Memory Section Erase End Time	R/W
FLRCV	Flash Memory Recovery Time	R/W

Information Block Access Index Register (IBAI)

This register defines the address bits 7-0 for read or write transaction to the Information Block.

Location: 00 $F880_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**								**Indirect Memory Address 7-0**							

Bit	Description
7-0	**Index Address 7-0.** The address is of a byte address in the flash Information Block. Access to words is allowed only to word-aligned addresses.
15-8	**Reserved.**

Information Block Data Register (IBD)

This register holds the data for read or write transaction from/to the Information Block. Byte or word reads and writes to this register are allowed to access a byte or word pointed to by IBA register. Note the program and erase limitations imposed by the flash protection scheme.

Location: 00 $F882_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Information Block Data**															

Bit	Description
15-0	**Information Block Data.**

Flash Core Write Protect Register 0 (FCWP0)

This register controls write protection of the Main Block per 8 Kbyte blocks. It is cleared (00_{16}) on reset.

Location: 00 $F890_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**FC0WP15-0**															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description
15-0	**Core Write Protect.** Each bit controls an 8 Kbyte block. For bit i, the block's start address is i x 00 2000_{16}. 0: Program and erase are inhibited for this block 1: Program and erase are allowed for this block Bits within the Core Boot Block and Host Boot Block, as defined by the protection word, are ignored, and the flash memory is always write protected.

Embedded Controller Modules (Continued)

Flash Memory Control Register (FLCR)

This register controls the on-chip flash operation modes. It is cleared (00_{16}) on reset. The core can write this register only when HLOCK bit in SMCCST register is 0 (see Section 5.3.9 on page 307). Otherwise, writes are ignored.

Location: 00 $F8C0_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**SMER**	**PMER**	**PE**	**IENPROG**	**Reserved**			**LOWPWR**
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	**Reserved**							**DISVRF**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**LOWPWR (Low Power Mode).** 0: On-chip flash is in Normal mode 1: On-chip flash is in Low Power mode This bit may be set (1) only when the core domain's clock is up to f_{FLPMAX}. It may be altered only when FRE bit in SZCFG1 register is set (1).
3-1	**Reserved.**
4	**IENPROG (Interrupt Enable for Program).** 0: Interrupt request disabled. 1: An interrupt request to the ICU is enabled (see FMLFULL in the Flash Memory Status register).
5	**PE (Program Enable).** This bit is impact core program operations only. 0: Flash Programing is disabled (default) 1: Flash Programing is enabled (if PMER and SMER are cleared)
6	**PMER (Page Erase).** When set (1) a valid write to the on-chip flash erases the entire page pointed to by the write address. This bit affects core erase operations only. This bit may be modified, only when the FMBUSY bit is cleared (0); otherwise, results are undefined. 0: Flash Page Erase is disabled (default) 1: Flash Page Erase is enabled (if PE and SMER are cleared).
7	**SMER (Section Erase)**. When set (1), a valid write to the on-chip flash erases the entire section pointed to by the write address. Write to a Main Block address results in a Main Block section erase. Write to an Information Block address results in an Information Block section and its corresponding Main Block section erase. This bit affects core erase operations only. This bit may be modified only when the FMBUSY bit is cleared (0); otherwise, results are undefined. 0: Flash Section Erase is disabled (default) 1: Flash Section Erase is enabled (if PMER and PE are cleared).
8	**DISVRF (Disable Verify)**. This bit affects core program and erase operations only. 0: Verify flash contents after program or erase is enabled (default). 1: Disable the automatic verification of flash contents after program or erase. This speeds up the program or erase speed.
15-9	**Reserved.**

Flash Memory Status Register (FLSR)

This register provides the on-chip flash status. It is cleared (00_{16}) on reset.

Location: 00 $F8C2_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**DERR**	**FMLFULL**	**FMBUSY**	**PERR**	**EERR**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	R/W1C	**EERR (Erase Error).** Indicates an error in erasing the flash by the core. It is cleared by writing 1 to it. Writing 0 is ignored. This bit may be cleared only when the FMBUSY bit is cleared (0). 0: No program error detected (default) 1: An error occurred during erase
1	R/W1C	**PERR (Program Error).** Indicates an error in writing to the flash by the core. It is cleared by writing 1 to it. Writing 0 is ignored. This bit may be cleared only when the FMBUSY bit is cleared (0). 0: No program error detected (default) 1: An error occurred during program
2	RO	**FMBUSY.** 0: The embedded flash is ready to accept operations 1: The embedded flash is not accessible since the PC87591x clocks are not ready or the embedded flash is busy programing or erasing
3	RO	**FMLFULL (Double-Buffer Full).** Indicates that the core can not write to the flash since its buffer is full. 0: There is at least one word empty in the flash double buffer. If IENPROG is set (1), in the Flash Memory Control register, an interrupt request to the ICU is active. 1: The double-buffer is full
4	R/W1C	**DERR (Data Loss Error).** Indicates that a double-buffer overrun error has occurred during core access. It is cleared by writing 1 to the DERR bit. Writing 0 is ignored. This bit may be cleared only when the FMBUSY bit is cleared (0). 0: No data loss detected (default) 1: A write to a full buffer was detected
7-5		**Reserved.**

Flash Memory Pre-Scaler Register (FLPSLR)

The FLPSLR register selects the pre-scaler divider ratio. FLPSLR defaults to (00_{16}) on reset.

Location: 00 $F8C4_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**				**FTDIV**			
Reset	0	0	0	0	0	0	0	0

Bit	Description
3-0	**FTDIV (Pre-Scaler Divider Value).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.
7-4	**Reserved.**

Embedded Controller Modules (Continued)

Flash Memory Start Time Register (FLSTART)

The FLSTART register controls the program/erase start delay time. FLSTART defaults to (14_{16}) on reset.

Location: 00 $F8C6_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	FTSTART							
Reset	0	0	0	1	0	1	0	0

Bit	Description
7-0	**FTSTART (Start Delay Count).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for required values.

Flash Memory Transition Time Register (FLTRAN)

The FLTRAN register controls some program/erase transition timings. FLTRAN defaults to (28_{16}) on reset.

Location: 00 $F8CA_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	FTTRAN							
Reset	0	0	1	0	1	0	0	0

Bit	Description
7-0	**FTTRAN (Transition Count).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.

Flash Memory Program Time Register (FLPROG)

The FLPROG register controls the programing pulse width. FLPROG defaults to ($0A_{16}$) on reset.

Location: 00 $F8D0_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	FTPROG							
Reset	0	0	0	0	1	0	1	0

Bit	Description
7-0	**FTPROG (Programing Pulse Width).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.

Embedded Controller Modules (Continued)

Flash Memory Page Erase Time Register (FLPERASE)

The FLPERASE register controls the page erase pulse width. FLPERASE defaults to (09_{16}) on reset.

Location: 00 $F8D2_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**FTPER**							
Reset	0	0	0	0	1	0	0	1

Bit	Description
7-0	**FTPER (Page Erase Pulse Width).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.

Flash Memory Section Erase Time Register (FLSERASE)

The FLSERASE register controls the section erase pulse width. FLSERASE defaults to (09_{16}) on reset.

Location: 00 $F8D4_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**FTSER**							
Reset	0	0	0	0	1	0	0	1

Bit	Description
7-0	**FTSER (Section Erase Pulse Width).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.

Flash Memory End Time Register (FLEND)

The FLEND register controls the program/page erase end delay time. FLEND defaults to (14_{16}) on reset.

Location: 00 $F8D8_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**FTEND**							
Reset	0	0	0	1	0	1	0	0

Bit	Description
7-0	**FTEND (End Delay Count).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.

Flash Memory Section Erase End Time Register (FLSEND)

The FLSEND register controls the section erase end delay time. FTSEND defaults to (32_{16}) on reset.

Location: 00 $F8DC_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**FTSEND**							
Reset	0	0	1	1	0	0	1	0

Bit	Description
7-0	**FTSEND (Section Erase End Delay Count).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.

Flash Memory Recovery Time Register (FLRCV)

The FLRCV register controls the recovery time required between two accesses. FTRCV defaults to (04_{16}) on reset.

Location: 00 $F8DE_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**FTRCV**							
Reset	0	0	0	0	0	1	0	0

Bit	Description
7-0	**FTRCV (Recovery Time).** This register should be configured based on the currently set core frequency. See Table 33 on page 223 for details of required values.

4.17 POWER MANAGEMENT CONTROLLER (PMC)

The Power Management Controller (PMC) module improves the efficiency of PC87591x operation by adjusting the chip's power consumption to the level of activity required by the application. This module works together with the High-Frequency Clock Generator (HCFG) and the core to control the activity of the PC87591x. It also interacts with the Multi-Input Wake-Up (MIWU), Interrupt Control Unit (ICU) and Debugger interface for wake-up events.

4.17.1 Features

- Three core domain power modes:
  - Active
  - Idle
  - Power Off
- Two clock inputs:
  - High-frequency clock (HFCLK)
  - Low-frequency clock (LFCLK)
- Power mode switching by software and/or hardware control
- High-frequency clock source Enable/Disable control
- Other core domain modules are controlled with power mode indications

4.17.2 The Core Domain Power Modes

Table 35 summarizes the main properties of the three modes and shows the activity levels of clocks while in the various power states.

Table 35. Core Domain Power Mode Summary

Mode	HFCG	LFCG	CLK	LFCLK[1]	V_{CC} Supply
Active[2]	On	On	HF	LF	On
Idle	On or Off[3]	On	Off	LF	On
Power Off	Off	On	Off	Off	Off

1. The RTC and TWM modules always work from the LF oscillator.
2. The core may execute the WAIT instruction while in Active mode to reduce power consumption while no core activity is required. This state is referred to as "Active Executing WAIT" in some places in the specification, but it is not a separate power state as far as clocks are concerned.
3. Can be turned off by software but depends also on SuperI/O clock domain activation.

Active Mode

In Active mode, the core domain operates at the frequency generated by the HFCG. This frequency may be changed dynamically using the Load, Fast, Load96 or Fast96 operations in the HFCG module. The module's respective enable/disable bits control module activity.

In this mode, power consumption can be reduced by selectively disabling modules and/or by the core executing the WAIT instruction. When WAIT is executed, the core stops executing new instructions until it receives an interrupt signal.

After reset, the PC87591x is in Active mode.

Idle Mode

In Idle mode, the clock is stopped for most of the core domain. Only the PMC and a limited number of core domain modules (such as the TWD) continue to operate at the low-frequency oscillator rate; they can wake up the core domain and resume instruction execution when required.

For modules that are active in Idle mode, details of their activity are included in the module's description.

Wake-up events are generated by the MIWU module according to the enabled internal and external events.

Power Off Mode

When V_{CC} power is turned off, the core domain reaches its lowest activity level. The contents of the memories (except for the on-chip flash) and registers are not preserved in this mode. Applying power to the core domain should be done using the V_{CC} power-up reset sequence.

Embedded Controller Modules (Continued)

A battery supply pin (V_{BAT}) provides power to the RTC and low-frequency clock oscillator (LFCO), allowing these parts to continue functioning even in Power Off mode. Some registers are reserved and events recorded. These registers and events are maintained by V_{PP}, as noted.

Details of the activity of each battery-operable module are provided in the module's specification.

4.17.3 Switching Between Power Modes

The switching from one power mode to another is done using the protocols described below.

Figure 77 shows the three power modes of the core domain and the transitions between them.

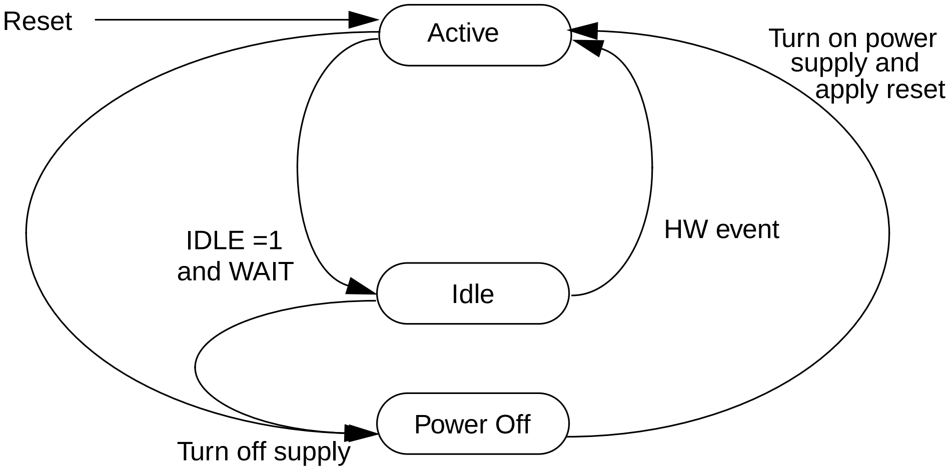


Figure 77. Power Modes and Transitions

Wake-Up Event

Some of the power-up switchings are based on receiving a wake-up event. This event has three possible sources:

- A maskable event (from MIWU)
- A Non-Maskable Interrupt (NMI)
- An ISE interrupt (from the Debugger interface)

The wake-up is identified by a high level on the maskable event and/or a low-to-high transition on NMI or ISE interrupts. Once a wake-up event is detected, it is latched until an interrupt acknowledge bus cycle is detected or a reset is applied.

Decreasing Power Consumption

Entering Idle Mode

Enter Idle mode by setting (1) IDLE bit in PMCSR register and then executing the WAIT instruction. WBPSM must be set before executing WAIT.The HFCG may be disabled to further reduce the power consumption. This is done by writing 1 to DHF in PMCSR register before executing WAIT.

Entering Power Off Mode

Switch to Power Off mode by turning off the supply to the V_{CC} pins of the PC87591x Note that V_{DD} must be turned off as well.

The $\overline{\text{PFAIL}}$ input may be used to interrupt the PC87591x so that context saving to a non-volatile memory can be completed and write operations to the RTC can be stopped before the power to the PC87591x is disconnected.

Increasing Activity

Fast Wake-Up from Idle Mode to Active

A hardware wake-up event causes the core domain to switch directly from Idle mode to Active mode. The following sequence is performed:

1. HDF in PMCSR register is cleared, thus enabling the high-frequency clock (if it was disabled).
2. After waiting for the high-frequency clock to become active (OHFC in PMCSR register is set), the core domain switches to Active mode.

When in Active mode, Idle bit in PMCSR register is cleared.

Embedded Controller Modules (Continued)

If the core was executing a WAIT instruction, it resumes operation by entering an interrupt routine (an enabled interrupt in the ICU, NMI or ISE).

Exit from Power Off

When in Power Off, activity can be resumed only by switching to Active mode. This is done by applying V_{CC} power to the PC87591x. The Power-Up reset sequence described in “VCC Power-Up Reset” on page 66 should be applied.

Power Mode Switch Protection

The PMC module includes a mechanism that protects the PC87591x from malfunctions caused by missing or unstable clock signals.

Clock Toggling Indication

OHFC and OLFC bits in PMCSR register indicate the current status of the high- and low-frequency clock inputs, respectively. The current status is based on indications from the HFCG and LFCG modules.

The PMC does not use the high-frequency clock when the OHFC bit is 0; it does not use the low-frequency clock when OLFC is 0.

During reset, the PC87591x clock does not toggle until OHFC is 1. During power mode change, if there is a request to switch to a non-stable or non-toggling clock, the power mode change stalls.

4.17.4 The Power Management Controller Status Register (PMCSR)

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

PMCSR is a byte-wide, read/write register that selects the Active or Idle modes. In addition, it controls the operation of the HFCG by enabling or disabling the high-frequency core clock domain. On reset, all non-reserved bits are cleared. PMCSR format is shown below.

Location: 00 $FF80_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**OLFC**	**OHFC**	**WBPSM**	**Reserved**		**IDLE**	**DHF**	**Reserved**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**Reserved.**
1	**DHF (Disable High-Frequency Oscillator).** When cleared (0), the HFCG is enabled. In Active mode, the HFCG is enabled regardless of the DHF value. If in Idle mode, DHF can be used to reduce power consumption. When DHF=1, the HFCG is disabled and the high-frequency clock is not generated. In Power off mode, the HFCG is disabled regardless of the DHF value. DHF is cleared by the hardware when a hardware wake-up event is detected.
2	**IDLE.** When set, the core domain enters Idle mode on the execution of a WAIT instruction. WBPSM must be set before executing the WAIT instruction. This bit can be set and cleared by software; it is cleared by hardware when a hardware wake-up event is detected.
4-3	**Reserved.**
5	**WBPSM (Wait Before Entering Power Save Mode).** When set, the switch from Active to Idle mode is done by setting the IDLE bit and executing a WAIT instruction. In addition, if DHF is set, the high-frequency oscillator is disabled only after the WAIT instruction is executed and Idle mode is entered.
6	**OHFC (Oscillating High-Frequency Clock).** 0: Indicates that the high-frequency clock received by the PMC is either disabled, not available or not producing a stable clock. When OHFC is cleared, the PMC does not switch to Active mode. 1: Indicates that the high-frequency clock received by the PMC is available and producing a stable clock.
7	**OLFC (Oscillating Low-Frequency Clock).** 0: Indicates that the low-frequency clock received by the PMC is either disabled, not available or not producing a stable clock. When the OLFC is cleared, the PMC does not switch from Active mode to Power Save or Idle modes. 1: Indicates that the low-frequency clock received by the PMC is available and producing a stable clock.

Embedded Controller Modules (Continued)

4.17.5 Usage Hints

The hints below apply when Idle mode is used with a disabled HFCG.

1. When disabling HFCG in Idle mode, on wake-up, a frequency clock may be generated that differs from the selected setting due to temperature variations in the working environment during the idle period. For details on the resulting deviation from the nominal frequency, refer to “tCLKINTwk” on page 385. To avoid any failures that may result from waking up to a higher frequency in case the access time to the internal or external memory is marginal using the current BIU configuration, follow the procedures exactly.

   Before entering Idle mode, configure SZCFGi (where i=0-2) for an additional Wait clock cycle (see Section 4.1.10 on page 83 for details on SZCFGi configuration).
2. After waking up from Idle mode, wait 0.5 seconds before returning to the optimal SZCFG0 configuration.

4.18 HIGH-FREQUENCY CLOCK GENERATOR (HFCG)

The HFCG generates the high-frequency clock (OSCCLK) based on the system's 32.768 KHz clock signal. The HFCG output is derived from OSCCLK and generates the host domain clock and the core domain clock. Clock generation is controlled by the core domain's PMC module (see Section 4.17 on page 235) and the host domain's SuperI/O configuration (see Section 6 on page 335).

To generate OSCCLK, the HFCG includes a programmable frequency multiplier. The core domain clock is derived from OSCCLK via a programmable pre-scaler; its frequency is in the range of 4 MHz to 20 MHz. The host domain clock is derived from OSCCLK via a pre-scaler that divides by 2, generating an output of 48 MHz (see Figure 78).

4.18.1 Features

- Programmable frequency multiplier for a wide range of output frequencies
- Core domain clock and host domain clock generation
- Programmable pre-scaler to derive the core domain clock from OSCCLK
- Separate enable/disable for core domain clock and host domain clock
- On V_{CC} power-up:
  - — 4 MHz default core domain clock frequency is set
  - — Host domain clock is disabled
- On WATCHDOG reset and Debugger Interface reset:
  - — If host domain clock is enabled, the 48 MHz clock monitor is initiated
  - — If host domain clock is disabled, the 4 MHz default core domain clock frequency is set

4.18.2 Functional Description

Figure 78 shows the HFCG blocks.

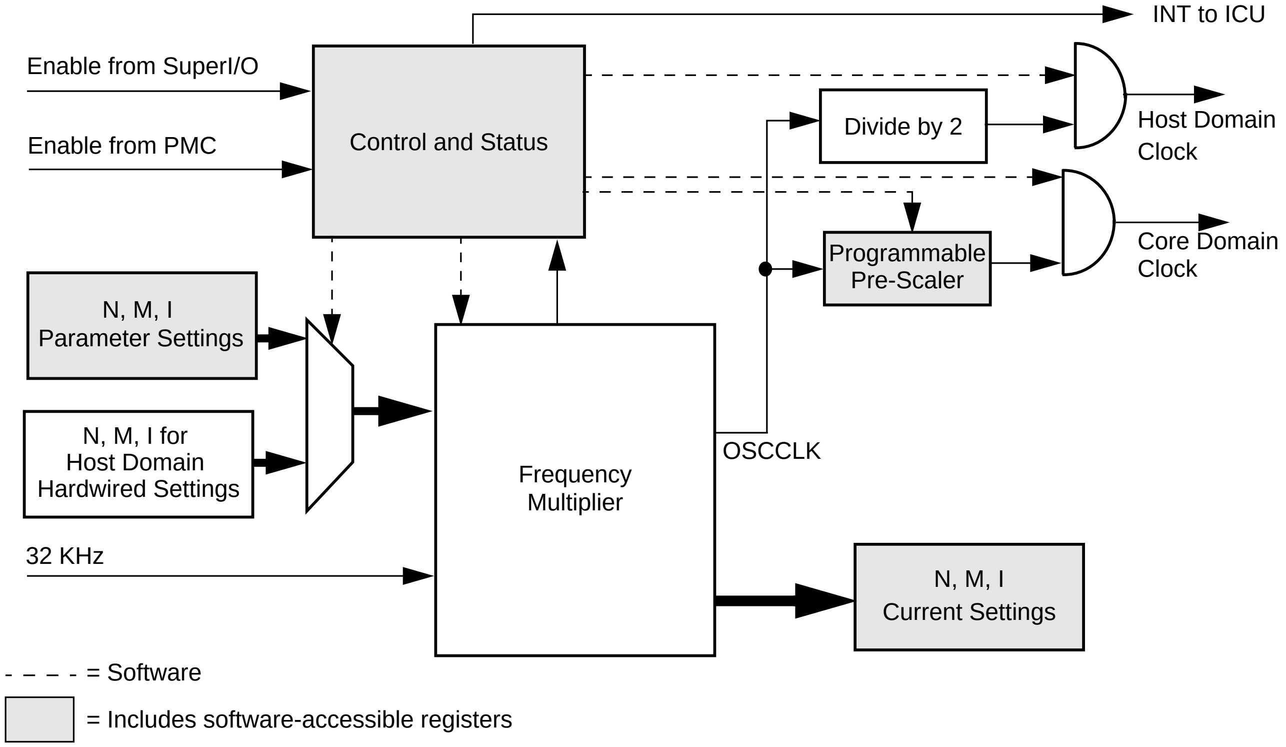


Figure 78. HFCG Schematic Diagram

The frequency multiplier includes a 5-bit variable (N) and a 14-bit variable (M); these variables define the OSCCLK frequency. There are two software methods to set the frequency of OSCCLK (by changing N and M); the method used depends on the HFCG state, as described in Section 4.18.3. The methods are:

Software Method 1. Write new values (HFCGML, HFCGMH and HFCGN registers) to a buffer and enable the programmable frequency setting. Either a normal or fast clock setting may be used. The programmable pre-scaler of the core domain clock is automatically set to a divide by 1; see "PMC Enabled SuperI/O Disabled State" on page 240.

Embedded Controller Modules (Continued)

Software Method 2. Enable the hardwired frequency setting (M and N are set to generate a 96 MHz OSCCLK). Either a normal or fast clock setting may be used. The programmable pre-scaler of the core domain clock is set according to the HFCGP register; see “SuperI/O Enabled State” on page 242.

During a frequency change, the OSCCLK output is low to prevent the system from using an unstable clock.

The HFCG is designed to be tightly coupled with the PMC. The HFCG receives two enable signals: one from the PMC and the other from the host domain. The PMC can enable or disable core domain clock generation while in Idle mode; it enables the core domain clock in Active mode. The SuperI/O configuration enables or disables clock generation for the host domain according to the host processor requests.

4.18.3 HFCG States

There are three groups of HFCG states:

- PMC Enabled SuperI/O Disabled: OSCCLK is programmable (set by hardware or by software method 1). The core domain clock is enabled; the host domain clock is disabled.
- SuperI/O Enabled PMC Enabled/Disabled: OSCCLK is fixed at 96 MHz (set by hardware or by software method 2). The host domain clock is enabled; the core domain clock is either enabled or disabled depending on PMC.
- Disabled: OSCCLK is disabled. Both core domain clock and host domain clock are disabled.

Transitions between the states are controlled by either hardware or firmware. Figure 79 illustrates the states and the hardware or software transitions between them. Some of the software settings and transitions are protected to improve the system’s durability with regard to software errors; see details in the following sections.

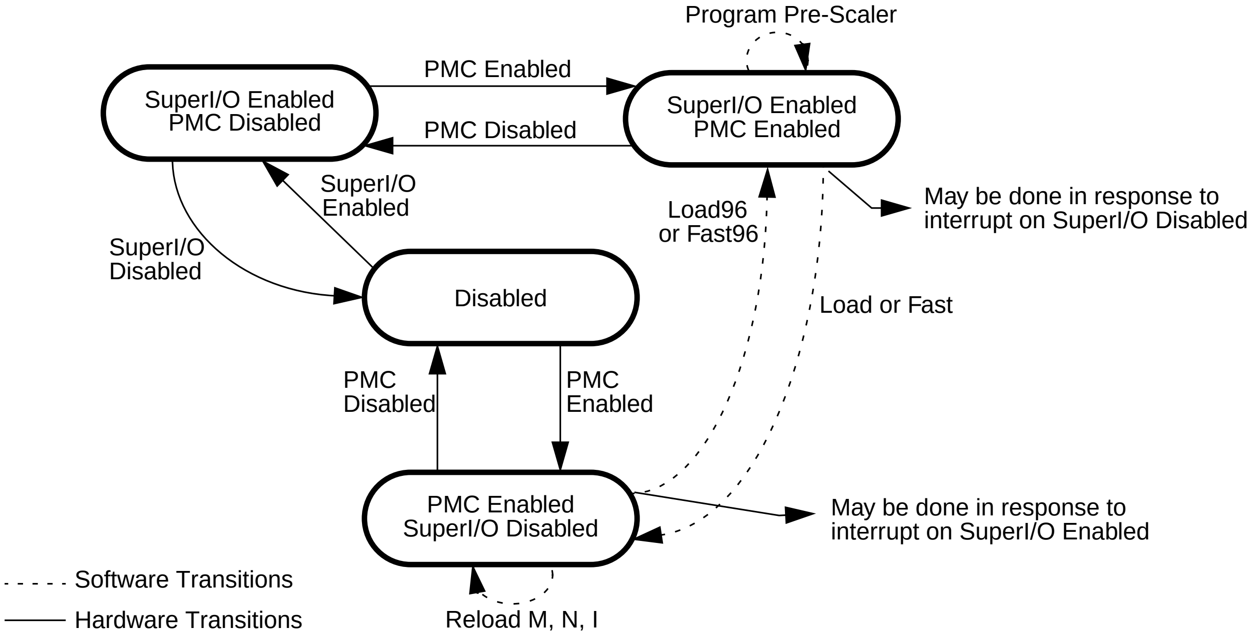

Program Pre-Scaler
PMC Enabled
SuperI/O Enabled PMC Disabled
PMC Disabled
SuperI/O Enabled PMC Enabled
SuperI/O Enabled
Load96 or Fast96
May be done in response to interrupt on SuperI/O Disabled
SuperI/O Disabled
Disabled
Load or Fast
PMC Disabled
PMC Enabled
PMC Enabled SuperI/O Disabled
May be done in response to interrupt on SuperI/O Enabled
- - - - - Software Transitions
——— Hardware Transitions
Reload M, N, I

Figure 79. HFCG States and Transitions

PMC Enabled SuperI/O Disabled State

Normal Clock Setting. This operation enables changing the clock frequency while in PMC Enabled SuperI/O Disabled state or switching from SuperI/O Enabled PMC Enabled state to PMC Enabled SuperI/O Disabled state.

To change the OSCCLK frequency, load the N and M variables with new values. M is loaded in two parts by writing to HFCGML and HFCGMH registers.

Load the new setting (N and M values, simultaneously) into the frequency multiplier. The core writes the new variables into a data input buffer. Then a command loads the new values into the frequency multiplier. The command also loads simultaneously the programmable pre-scaler with 0 (set to a divide by 1). Note that HFCGP register does not change its contents.

To set a new clock frequency:

1. Write the N value to HFCGN register.
2. Write the low byte of the M value to HFCGML register.
3. Write the upper bits of the M value to HFCGMH register.
4. Set LOAD in the HFCGCTRL1 register to 1.

Embedded Controller Modules (Continued)

When the new HFCGML, HFCGMH and HFCGN register values are loaded, the HFCG holds OSCCLK low until the frequency multiplier locks onto the target frequency and the new frequency stabilizes. The core domain clock is also low during this time. This automatic locking process can take several milliseconds to complete.

Frequencies within the range of 4 MHz to 20 MHz are valid. See Table 36 for a sampling of selected frequencies and their corresponding M and N values.

Table 36. Frequencies of Selected Settings

Frequency[1] (MHz)	HFCGMH	HFCGML	HFCGN
4.00 (default)	03_{16}	CF_{16}	08_{16}
5.00	03_{16}	92_{16}	06_{16}
6.00	03_{16}	92_{16}	05_{16}
7.00	07_{16}	81_{16}	09_{16}
8.00	$0A_{16}$	$7C_{16}$	$0B_{16}$
9.00	$0A_{16}$	$B9_{16}$	$0A_{16}$
10.00	$0A_{16}$	$B9_{16}$	09_{16}
12.00	11_{16}	$2A_{16}$	$0C_{16}$
14.00	19_{16}	07_{16}	$0F_{16}$
16.00	11_{16}	29_{16}	09_{16}
18.00	11_{16}	29_{16}	08_{16}
20.00	17_{16}	$D6_{16}$	$0A_{16}$

[1] This value is referred to as $t_{CLKINTnom}$ in the AC specifications.

Fast Clock Setting. This operation enables changing the clock frequency while in PMC Enabled SuperI/O Disabled state or switching from SuperI/O Enabled PMC Enabled state to PMC Enabled SuperI/O Disabled state.

The HFCG maintains an internal I variable. The I variable is defined by two byte-wide registers: HFCGIL and HFCGIH. If new M and N values are loaded, the frequency multiplier automatically searches for the I value needed to lock onto the target frequency. The locking process can take several milliseconds to complete. The I variable can be recorded for a given M and N set of values and used later to reduce the time needed for frequency locking.

To record the I value:

1. Read the low byte of the I value via HFCGIL register.
2. Check if IVLID bit in HFCGCTRL1 register is set to 0. If yes, repeat steps 1 and 2.
3. Read the upper six bits of the I value via HFCGIH register.

To fast load a new setting, load the M and N values and the corresponding I value. Then set FAST bit in HFCGCTRL1 register to 1. The FAST command loads (simultaneously) the new values into the frequency multiplier and loads the programmable pre-scaler with 0 (set to a divide by 1). Note that HFCGP register does not change its contents. The frequency multiplier quickly locks onto the target frequency without searching for a new I value.

To fast set a new clock frequency:

1. Write the N value to HFCGN register.
2. Write the low byte of the M value to HFCGML register.
3. Write the upper bits of the M value to HFCGMH register.
4. Write the low byte of the I value to HFCGIL register.
5. Write the upper six bits of the I value to HFCGIH register.
6. Set FAST bit in HFCGCTRL1 register to 1.

Changes in temperature or voltage may cause variations in the value of I for a given output frequency. If these changes occur in the interval between recording I and its use, the output frequency generated following a fast frequency setting may differ from the target frequency. However, after some time, the output frequency converges to the desired frequency.

SuperI/O Enabled State

Normal Clock Setting. This operation enables switching from PMC Enabled SuperI/O Disabled state to SuperI/O Enabled PMC Enabled state.

To generate a 96 MHz OSCCLK, a command loads the hardwired M and N values (simultaneously) into the frequency multiplier. At the same time, the command loads the programmable pre-scaler with the value held in HFCGP register. To set a 96 MHz clock frequency and the required core clock.

1. Write the HFCGP value.
2. Set LOAD96 bit in HFCGCTRL1 register.

The HFCGN, HFCGML and HFCGMH registers are ignored and left unchanged when switching to SuperI/O Enabled states.

Fast Clock Setting. This operation enables fast switching from PMC Enabled SuperI/O Disabled state to SuperI/O Enabled PMC Enabled state.

The HFCG maintains an internal I variable for the 96 MHz clock. The I variable is defined by two byte-wide registers: HFCGIL and HFCGIH. In a LOAD96 operation, the frequency multiplier automatically searches for the I value needed to lock onto the target frequency. The locking process can take several milliseconds to complete. The I variable can be recorded for the 96 MHz setting and used later to reduce the time needed for frequency locking.

To record the 96 MHz I value:

1. Write the required HFCGP value.
2. Enter any of the SuperI/O Enabled states using the Normal clock setting or Hardware clock setting process.

To fast set a 96 MHz clock frequency:

1. Write the required HFCGP value.
2. Set FAST96 bit in HFCGCTRL1 register to 1.

The HFCGN, HFCGML and HFCGMH registers are ignored and left unchanged when switching to SuperI/O Enabled states.

4.18.4 The Programmable Pre-Scaler: Core Domain Clock Generation

The core domain clock (CLK) is driven from OSCCLK via a 5-bit pre-scaler. When in PMC Enabled SuperI/O Disabled state, the pre-scaler divides by 1. In SuperI/O Enabled PMC Disabled state, the pre-scaler is programmable, as defined in HFCGP register. In other states, the core-domain clock is disabled.

The core domain clock may be set in the range of 4 MHz to 20 MHz.

When LOAD96 or FAST96 bit in HFCGCTRL1 register is set (1), the pre-scaler is set to the value held in HFCGP register (core frequency = 96 MHz / (HFCGP +1)).

When in SuperI/O Enabled PMC Enabled state, a write to HFCGP register changes the core’s frequency at the next cycle of the pre-scaler.

When LOAD or FAST bit in HFCGCTRL1 register is set (1), the pre-scaler is automatically set to a divide by 1. The contents of HFCGP are unchanged.

When in SuperI/O Enabled PMC Enabled state or SuperI/O Enabled PMC Disabled state, WATCHDOG reset or Debugger Interface reset sets HFCGP to its reset value and initializes the pre-scaler using this value; see Section 4.18.6 on page 243 for details about setting the pre-scaler during state transitions.

When in Disabled state or PMC Enabled and SuperI/O Disabled state, WATCHDOG reset or Debugger Interface reset keeps the pre-scaler in the divide by 1 operation and loads the HFCGP to its reset value.

In all states, on V_{CC} power-up, WATCHDOG reset or Debugger Interface reset, HFCGP is set to its reset value and the pre-scaler is set to a divide by 1.

4.18.5 State Transitions

The following section describes the actions taken by the HFCG during state transitions. Unless explicitly specified, the actions are initiated by hardware.

Transition to PMC Enabled SuperI/O Disabled State

- When transitioning from Disabled state:
  - OSCCLK defaults to a frequency set by the input data buffer (according to the most recent LOAD, FAST or Reset command).
  - The pre-scaler defaults to a divide by 1.
- When transitioning from SuperI/O Enabled PMC Enabled state:
  - Software sets OSCCLK and the pre-scaler by the LOAD or FAST command (software method 1).
- The host domain clock is disabled (low).
- The core domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized.

Embedded Controller Modules (Continued)

When transitioning from SuperI/O Enabled PMC Enabled state, there may be a transition period in which the core domain clock is still toggling according to the setting in SuperI/O Enabled PMC Enabled state. This occurs until software method 1 is performed by the interrupt handler of the SuperI/O Disabled event.

Transition to SuperI/O Enabled PMC Disabled State

- OSCCLK defaults to 96 MHz
- The host domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized
- The core domain clock is disabled (low)

Transition to SuperI/O Enabled PMC Enabled State

- When transitioning from SuperI/O Enabled PMC Disabled state:
  - — OSCCLK defaults to 96 MHz
  - — The pre-scaler is set according to HFCGP
- When transitioning from PMC Enabled SuperI/O Disabled state:
  - — Software sets OSCCLK and the pre-scaler by the LOAD96 or FAST96 command (software method 2)
- The host domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized
- The core domain clock is enabled and starts toggling as soon as the frequency multiplier has stabilized

When transitioning from PMC Enabled SuperI/O Disabled state, there may be a transition period in which the core domain clock is still toggling according to the setting in PMC Enabled SuperI/O Disabled state. This occurs until software method 2 is performed by the interrupt handler of the SuperI/O Enabled event. During this transition period, the host domain clock is still disabled.

4.18.6 48 MHz Clock Monitor

While in one of the SuperI/O Enabled states, a WATCHDOG reset or Debugger Interface reset does not interfere with the operation of the SuperI/O. Thus, if the 48 MHz clock is correctly toggling, it is not interfered with. Toggling is monitored by the 48 MHz clock monitor. If a violation is detected, the 48 MHz monitor error flag is set and the HFCG goes through a complete reset, after which it is put into PMC Enabled SuperI/O Disabled state, ignoring the host domain enable signal.

The 48 MHz clock monitor is initiated when a WATCHDOG reset or Debugger Interface reset occurs while the HFCG is in a SuperI/O Enabled state. The monitor checks that the host domain clock frequency is in the range of 48 MHz ±1%.

If the host domain clock frequency is in the correct range, the HFCG continues generating the 96 MHz OSCCLK, but the programmable pre-scaler is set to a default of divide by 24. The HFCG is in SuperI/O Enabled PMC Enabled state.

The core reset routine is responsible for switching to SuperI/O Enabled PMC Enabled state and/or communicating the failure to the host software.

4.18.7 HFCG Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

HFCG Register Map

Mnemonic	Register Name	Type
HFCGCTRL1	HFCG Control 1	Varies per bit
HFCGML	M Low Byte Value	R/W
HFCGMH	M High Byte Value	R/W
HFCGN	N Value	R/W
HFCGIL	I Low Byte Value	R/W
HFCGIH	I High Byte Value	R/W
HFCGP	HFCG Pre-Scaler	R/W
HFCGCTRL2	HFCG Control 2	Varies per bit

Embedded Controller Modules (Continued)

HFCG Control Register 1 (HFCGCTRL1)

The HFCGCTRL1 register sets the frequency multiplier's operating parameters. On V_{CC} power-up, WATCHDOG reset and Debugger Interface reset, it is initialized to $0C_{16}$.

Location: 00 $FFA0_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**FAST96**	**LOAD96**	**IVLID**	**OHFC**	**PENABLE**	**FAST**	**LOAD**
Reset	0	0	0	0	1	1	0	0

Bit	Type	Description
0	WO	**LOAD (Load M and N Values).** Write 1 to this bit to perform a normal frequency change by loading the HFCGML, HFCGMH and HFCGN buffer data to the frequency multiplier. The pre-scaler is automatically set to a divide by 1. The bit always reads back 0. Results are undefined when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96.
1	WO	**FAST (Load M, N and I Values).** Write 1 to this bit to perform a fast frequency change by loading the HFCGML, HFCGMH, HFCGN, HFCGIH and HFCGIL input buffer data in the frequency multiplier. The pre-scaler is automatically set to a divide by 1. The bit always reads back 0. Results are undefined when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96.
2	RO	**PMC ENABLE (Enable Core Domain Clock).** Provides the status of the PMC enable/disable command. Any data written to this bit is ignored. 0: Disabled 1: Enabled
3	RO	**OHFC (Output Clock Status).** Indicates when the HFCG is oscillating and produces a stable clock. Any data written to this bit is ignored. 0: Not oscillating 1: Oscillating with stable output
4	RO	**IVLID (I Value Valid).** 0: Data read is invalid; repeat HFCGIL register read operation 1: Data read is valid; HFCGIH can be read
5	WO	**LOAD96 (Load M and N Values for 96 MHz).** Write 1 to this bit to perform normal locking on 96 MHz frequency by loading the hardwired variables M and N to the frequency multiplier. The pre-scaler is set according to the value held in HFCGP register. The bit always reads back 0. Results are undefined when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96.
6	WO	**FAST96 (Load M, N and I Values** for 96 MHz**).** Write 1 to this bit to perform fast locking on 96 MHz frequency by loading the hardwired variables M and N and registers HFCGIH and HFCGIL to the frequency multiplier. The pre-scaler is set according to the value held in HFCGP register. The bit always reads back 0. Results are undefined results when more than one of the following bits is written with 1 at the same time: LOAD, FAST, LOAD96, FAST96.
7		**Reserved.**

Embedded Controller Modules (Continued)

HFCGM Low Value Register (HFCGML)

The HFCGML register contains the lower eight bits of the frequency multiplier M value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGML value of the currently set frequency. On V_{CC} power-up, WATCHDOG reset and Debugger Interface reset, it is loaded with CF_{16}.

Location: 00 $FFA2_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**HFCGM7-0**							
Reset	1	1	0	0	1	1	1	1

Bit	Description
7-0	**HFCGM7-0.** Contains the lower eight bits of the M value.

HFCGM High Value Register (HFCGMH)

The HFCGMH register contains the upper bits of the frequency multiplier M value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGMH value of the currently set frequency. On V_{CC} power-up, WATCHDOG reset and Debugger Interface reset, it is loaded with 03_{16}.

Location: 00 $FFA4_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**HFCGM**							
Reset	0	0	0	0	0	0	1	1

Bit	Description
7-0	**HFCGM.** Contains the upper bits of the M value.

HFCGN Value Register (HFCGN)

The HFCGN register is a byte-wide, read/write register containing five bits of the frequency multiplier N value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGN value of the currently set frequency. On V_{CC} power-up, WATCHDOG reset and Debugger Interface reset, it is loaded with 08_{16}.

Location: 00 $FFA6_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**HFCGN4-0**				
Reset	0	0	0	0	1	0	0	0

Bit	Description
4-0	**HFCGN4-0.** Contains the five bits of the N value.
7-5	**Reserved.**

Embedded Controller Modules (Continued)

HFCGI Low Value Register (HFCGIL)

The HFCGIL register contains the lower eight bits of the frequency multiplier I value. Data written to the register is stored in the setup buffer. Reading the register returns the value of its first 8 bits. (IVLID bit in HFCGCTRL1 register indicates if the data is valid.)

Location: 00 FFA8$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**HFCGI7-0**							

Bit	Description
7-0	**HFCGI7-0.** Contains the lower eight bits of the I value.

HFCGI High Value Register (HFCGIH)

The HFCGIH register is a byte-wide, read/write register containing the upper six bits of the frequency multiplier I value. Data written to the register is stored in the setup buffer. Reading the register returns the HFCGIH value of the currently set frequency.

Location: 00 FFAA$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**HFCGI13-8**					

Bit	Description
5-0	**HFCGI13-8.** Contains the upper six bits of the I value.
7-6	**Reserved.**

HFCG Pre-Scaler Register (HFCGP)

The HFCGP register is a byte-wide, read/write register. It allows the core clock to be derived from the 96 MHz clock.

On reset HFCGP is initialized to 17$_{16}$.

Location: 00 FFAC$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**HFCGP4-0**				
Reset	0	0	0	1	0	1	1	1

Bit	Description
4-0	**HFCGP4-0 (Pre-Scaler Divider Value).** The divider of OSCCLK is the number defined by HFCGP(4-0) + 1, e.g., a value of 00$_{16}$ results in a divide by 1; a value of 1F$_{16}$ results in a divide by 32. Results are undefined when the pre-scaler holds a value that yields a core domain clock frequency higher than 20 MHz or lower than 3 MHz. The pre-scaler is set according to the value held in HFCGP register when LOAD96 bit or FAST96 bit is set (1). Reading HFCGP register returns the last value written to it.
7-5	**Reserved.**

Embedded Controller Modules (Continued)

HFCG Control Register 2 (HFCGCTRL2)

The HFCGCTRL2 register sets the frequency multiplier's operating parameters. On V_{CC} power-up, WATCHDOG reset and Debugger Interface reset, it is initialized to 00_{16}.

Location: 00 $FFAE_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			96MON	**MONERR**	**SCESTP**	**SCESTR**	**SENABLE**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	RO	**SIO ENABLE (Enable Host Domain Clock).** Provides the status of the SuperI/O enable/disable command. Any data written to this bit is ignored. 0: Disabled 1: Enabled
1	R/W1C	**SCESTR (SIO Clock Enable Start).** Indicates that a rising edge was detected on the SENABEL signal. When set, an interrupt is sent to the ICU (level high). Cleared by writing 1 to it. This interrupt may be masked only in the ICU. 0: No rising edge on SENABLE was detected (default) 1: A rising edge on SENABLE was detected and an interrupt is sent to the ICU
2	R/W1C	**SCESTP (SIO Clock Enable Stop).** Indicates that a falling edge was detected on the SENABEL signal. When set, an interrupt is sent to the ICU (level high). Cleared by writing 1 to it. This interrupt may be masked only in the ICU. 0: No falling edge on SENABLE was detected (default) 1: A falling edge on SENABLE was detected and an interrupt is sent to the ICU
3	R/W1C	**MONERR (Monitor Error).** Indicates that a 48 MHz monitor error was detected during the last WATCHDOG or Debugger Reset process. Note that this bit is cleared by all resets except WATCHDOG and Debugger Interface resets. Once set, this bit is cleared by writing 1 to it. 0: No error detected (default) 1: 48 MHz monitor detected an out of range frequency
4	RO	**96MON (96 MHz Oscillations On).** Indicates the oscillation frequency at which the HFCG is operating. 0: HFCG is oscillating as defined by the HFCGM and HFCGN. The core clock is identical to HFCG frequency. 1: HFCG is oscillating at 96 MHz. The core clock is pre-scaled as defined by HFCGP.
7-5		**Reserved.**

4.19 THE DEBUGGER INTERFACE

The Debugger Interface module links between a debugger running on a host machine and the PC87591x.

The Debugger interface associates a processor number with the processor core. The number associated with the core is 0.

The Debugger interface can also program the on-chip flash.

4.19.1 Features

- Debugger communication via an IEEE 1149.1b 1994 JTAG serial bus
- Test Access Port (TAP) for JTAG serial bus
- Hardware reset signal assertion by debugger command
- Configurable ABORT event
- Interrupt signal ($\overline{\text{TINT}}$) to the debugger indicates a waiting message
- Waiting message to on-chip processor by non-maskable ISE interrupt
- 8-word (16-byte) Rx (downstream) data link
- 8-word (16-byte) Tx (upstream) data link
- On-chip flash support
  - Read, Program and Erase
  - Read protect option to secure contents

4.19.2 Structure

The Debugger interface consists of a TAP, ISE Interrupt Control, Rx and Tx data links and a PC87591x reset-by-command source circuit. Figure 80 shows these functional blocks.

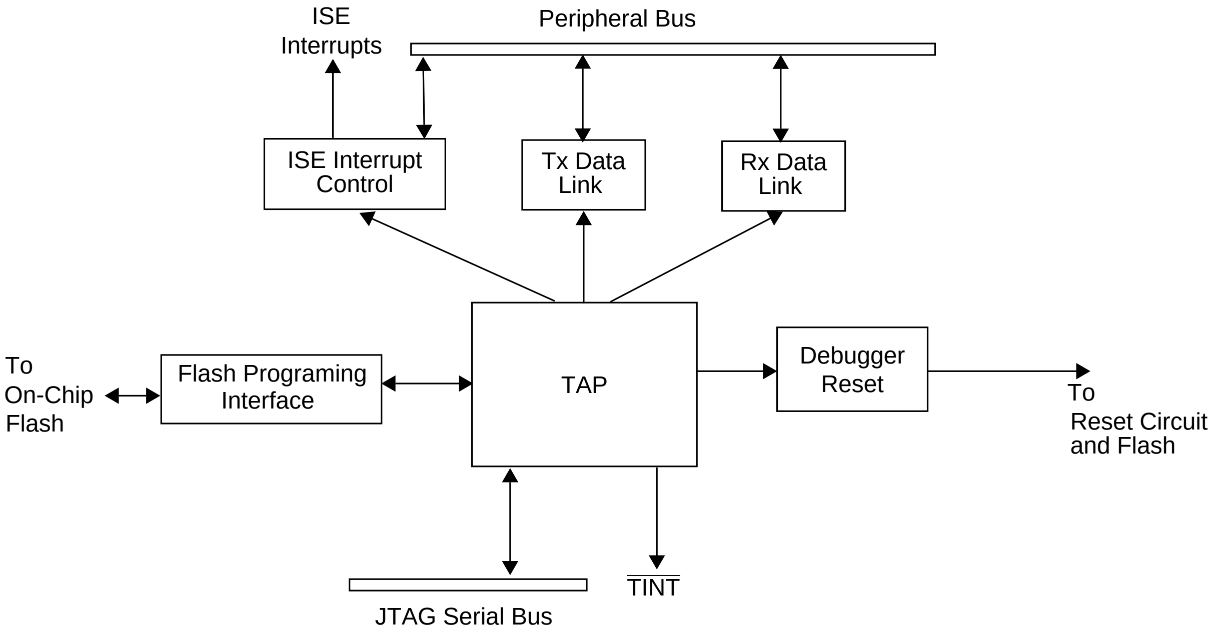


Figure 80. Debugger Interface

The TAP block has an IEEE 1149.1b 1994 standard interface to a JTAG serial bus. In addition, it uses $\overline{\text{TINT}}$ to notify the host/debugger that a message is waiting upstream. The TAP and the JTAG form the debugging communication channel and port for the PC87591x. They operate as follows:

- The TAP copies downstream messages (from the debugger to one of the on-chip processors) to the Rx data link, to be read by the relevant processor via the peripheral bus.
- A processor writes upstream messages (from an on-chip processor to its debugger) to the Tx data link for upstream transmission via the JTAG serial bus.
- The ISE interrupt control generates ISE interrupts to the processor in cases of ABORT or a waiting message in the Rx data link.
- The debugger reset circuit generates a PC87591x reset by a debugger command. This is in addition to the Power-Up and Hardware reset sources.
- Flash access (control, status, read, write and erase) interface and control is provided. The Debugger interface generates a reset signal when JTAG access to the flash is enabled.

4.19.3 Debugger Interface Functional Description

The Debugger interface supports four operating modes: Rx session, Tx session, chip RESET and ABORT. Some of these modes can be active simultaneously for the same or different processors. The ISE interrupt control block includes hooks to control these conditions.

Rx Session (Sending Data Downstream)

A message sent downstream by the debugger to a processor is called an Rx session. In an Rx session, a debugger uses both the TAP and the JTAG to monitor the “busy” indication for the Rx data link. Following a “not busy” indication, a message is sent to a processor via the JTAG, TAP and Rx data link.

One of the internal ISE interrupts is asserted (if not active) according to the PID field of the instruction currently loaded into the TAP IR register. The signaled processor accesses the data link via the peripheral bus, reads the message length (optional) and fetches it from the Rx data buffer.

At the end of the data transfer, the processor turns off the “busy” indication of the Rx data link.

Tx Session (Sending Data Upstream)

A message sent upstream by a processor to the debugger is called a Tx session. In a Tx session, one of the processors tries to own the Tx data link by accessing the Tx semaphore DBGTXLOC register. If successful, it writes a message body to the Tx data buffer and a message length, in words, to DBGTXST register.

After completion of the data buffer update, the processor sets ASSERT bit in DBGTINT register to 1. This signals the debugger with an active-low pulse on $\overline{\text{TINT}}$. The debugger reads the data using both the TAP and the JTAG.

At the end of the data transfer, the semaphore circuit is set to “not busy” and $\overline{\text{TINT}}$ is released.

Chip RESET

The PC87591x is reset by a dedicated TAP instruction.

ABORT

Either a TAP instruction or a bit-set operation in DBGABORT register generates an ABORT. Asserting an ISE interrupt together with a non-zero ABORT_i bit in DBGISESRC register signals an ABORT operation. The ISE interrupt control circuit asserts the ISE interrupt according to the pre-programed mask bits in ABORT_MASK register. A dedicated circuit, together with a set of registers in the ISE Interrupt Control, clears the ISE requests after they have been served.

Rx Data Link

The Rx data link consists of an 8-word read/write data buffer, DBGRXD0 to DBGRXD7 registers and the Status (DBGRXST) register.

On PC87591x reset, DBGRXST register is set to its reset value. On TAP reset, the data link maintains its values.

When the TAP controller is in Update-DR state and the current IR is SCAN_RX, DBGRXDX registers are updated from the TAP Data Shift (DBGDATA) register. Data is valid to the processor only while BUSY bit in DBGRXST register is set. In this case (i.e., Update-DR of SCAN_RX), the TAP controller copies the PID and length fields from its IR to the Status register and sets BUSY bit to 1 in this state.

A processor may turn off the BUSY bit by writing 1 to it. BUSY can be cleared even when TCK is not toggling. The Rx data link functions in Active or Idle modes.

DBGDATA length is set to the length field of the SCAN_RX instruction before Capture_DR state (see Section 4.19.7 on page 260). No parallel load is executed in Capture-DR state.

Embedded Controller Modules (Continued)

TX Data Link

The Tx Data link consists of an 8-word, read/write data buffer, DBGTXD0 to DBGTXD7 registers, a read/write Status register (DBGTXST), a read/write semaphore lock register (DBGTXLOC) and a write-only $\overline{\text{TINT}}$ control register (DBGTINT).

On Power-Up reset, the Tx Data link is reset (negating any pending message), DBGTXLOC and DBGTXST are set to their reset values and $\overline{\text{TINT}}$ is released (1). On Warm and Internal reset, any partial message (i.e., $\overline{\text{TINT}}$=1) is negated by setting DBGTXLOC and DBGTXST to their reset values. Messages that were completed (i.e., $\overline{\text{TINT}}$=0) are maintained for transmission to the host by maintaining DBGTXLOC and DBGTXST values.

On TAP reset, the data link maintain its values.

DBGTXD registers are captured by the TAP data shift register (DBGDATA) in Capture-DR state of the TAP controller when the current Information Register (IR) is SCAN_TX. The DBGDATA length is set dynamically, according to the length field of DBGTXST register before the Capture_DR state of the SCAN_TX operation (see Section 4.19.7 on page 260). The TAP IR register captures the values of PID and MSG_LEN fields of DBGTXST register when the TAP controller is in Capture-IR state. No parallel load is executed in Update-DR state.

The semaphore is implemented by DBGTXLOC register. A write operation, to PID field of this register, of a value other than '1111' changes the contents of this field only when the PID field equals '1111'. This field returns to '1111' in one of the following ways:

- Write operation of '1111' to PID field of DBGTXLOC register from the peripheral bus while $\overline{\text{TINT}}$ is not asserted
- Update-DR state of the TAP controller when the current instruction loaded into TAP IR register is SCAN_TX (this action take place at the rising edge of TCK)
- On Warm or Internal reset if $\overline{\text{TINT}}$=1 and on Power-Up reset

A processor can access DBGTXDi registers in Active mode only. For a processor to gain ownership of the Tx link, it must capture the DBGTXLOC semaphore using the following sequence:

1. Verify that the value of PID in DBGTXLOC register is '1111'.
2. Write the processor PID code to DBGTXLOC register.
3. Read DBGTXLOC. If the PID field is equal to the value that was written, the data link is granted. If not, repeat step 1.

A processor should access DBGTXDi and DBGTXST registers only after successfully gaining ownership over the Tx link by using the above sequence.

The $\overline{\text{TINT}}$ signal is an active-low pulse asserted by the Tx data link when ASSERT bit in DBGTINT register is written with 1. It is de-asserted, together with the semaphore indication in Update-DR state of the TAP controller, when the current instruction loaded into TAP IR register is SCAN_TX.

Access to DBGTXLOC, DBGTXST and DBGTXDi registers should be done only while $\overline{\text{TINT}}$ is not asserted. $\overline{\text{TINT}}$ negation can be identified by the release of PID ('1111').

Debugger Reset Circuit

Chip reset is asserted in the Update-DR state of the TAP controller when the current instruction is ASSERT_DBG_RST. This triggers a Debugger reset, as described in "ASSERT_DBG_RST" on page 260. This circuit is functional in Active or Idle modes. It is functional, while TCK is not toggling, one cycle after exit from Update-DR state.

ISE Interrupt Control

The ISE interrupt control module sends ISE interrupt requests to the processor core. It consists of the write-only DBGABORT register, the read/write DBGISESRC register, and the DBGMASKS shift register.

The DBGISESRC register is cleared on PC87591x reset. During TAP reset, the values of these registers are maintained. The DBGMASKS value is modified only in Update-DR state of the TAP controller when the current instruction is SCAN_ABORT_MASK.

The ISE interrupt control module issues an ISE interrupt request to a specific processor or multiple processors, together with the matched bit in DBGISESRC register, according to the MESSAGE or ABORT event.

In a MESSAGE event, an ISE interrupt is requested for a specific processor according to the PID field of the SCAN_RX instruction. The request is issued (together with DBGISESRC bit assertion) if the current instruction loaded in TAP IR is SCAN_RX and the TAP controller is in Update-DR state (rising edge of TCK).

An ABORT event occurs when SCAN_RX is executed with a PID of all 1s (ISE and ABORT_i bits in DBGISESRC register are asserted) or when processor bits P_i are set in DBGABORT register by one of the processors. That is, ISE and DBGISESRC bits are asserted with the write operation itself; if DBGABORT is written with some 0 bits, the PIDs related to these bits do not get the ISE. In an ABORT event, the assertion of each ISE interrupt and DBGISESRC bit depends on its masking bit in DBGMASKS register.

Each ISE interrupt is cleared when ABORT_i and RX_i in DBGISESRC register are both 0. Any bit in DBGISESRC register is cleared by writing 1 to it (writing 0 is ignored). If there is a new source activity in the same write cycle during which DBGISESRC register of a specific processor is cleared, the ISE interrupt control asserts ISE again, together with its source bit.

The ISE interrupt is an active-high pulse. For nested ISE sources, the ISE remains asserted, and the ISE interrupt control module sets the new source bit to 1.

The DBGMASKS register is not available to the peripheral bus; it is accessed only from the JTAG serial bus when the SCAN_ABORT_MASK instruction is loaded into the TAP IR.

The ISE signal, together with its source bit, should be asserted in Active or Idle modes for wake-up purposes when the source is a debugger message or debugger abort. Other functionality should be consistent while changing modes (i.e., during wake-up). Full functionality of this module is maintained even when TCK is not toggling.

Clock Synchronization

Some operations are related to TCK, others to the PC87591x main clock and yet others are asynchronous. The PC87591x logic guarantees correct operation and a meta-stable protected interface in all its operating modes (i.e., Active and Idle).

The core may access the Debugger interface registers only in Active mode.

4.19.4 Flash Interface Functional Description

The Flash interface enables access to the flash through the JTAG serial bus without any core involvement. This enables a firmware-independent update of flash contents.

The Flash interface is divided into three parts:

- Flash Control and Status: Controls flash operations and returns flash status.
- Flash Address: Stores the address for flash operations and includes an auto-increment mechanism for speeding up the flash update process.
- Flash Data: Handles flash data for both read and write operations.

Enabling Access to the Flash

The flash can be accessed through the JTAG serial bus in OBD and DEV environments. In IRE and PROG environments, access cannot be enabled (i.e., write operations to the JTAG Access Enable (JTFIE) bit in the Flash Control and Status register, FL_CT_ST, are ignored).

To enable Flash interface access to the flash, use the following sequence:

1. Set Instruction Register (IR) to Flash Control Scan Instruction.
2. Scan in a code with the Flash JTFIE bit in FL_CT_ST register is set to enable access to the PC87591x.
3. Repeat step 2 until the FMBUSY bit in FL_CT_ST register is cleared.

Enabling the flash puts the PC87591x in Reset state. It remains in this state, with the clocks enabled and any core or host activity disabled, until the Flash interface is again disabled. To disable the flash, clear the JTFIE bit. This releases the PC87591x from reset and enables the core to start executing the code stored in the flash.

Flash Contents Protection

The flash has a contents protection mechanism that protects the flash contents from read operations and prevents them from being altered during write transactions. When this mechanism is set, only a special erase that erases the entire flash can be performed.

When the Blank Flash flag is set, it indicates to the core that the flash has no valid code and that the core should remain in Reset state until the protection information is changed. Both this flag and the Access Enable flag are part of the flash protection byte and should be programed after the flash is loaded with data. See Section 4.16.6 on page 226.

Flash Data Read

To read the contents of the flash, perform the following sequence:

1. Enable flash access (if not already enabled).
2. Set the flash to read as follows:
   a. Set the IR to Flash Control and Status Scan instruction.
   b. Scan in a read operation code (8000_{16}).
3. Set the address as follows:
   a. Set IR to Flash Address Scan instruction.
   b. Scan the address of the memory location to be read into the chip.

Embedded Controller Modules (Continued)

4. Read the data as follows:
   a. Set IR to Flash Data Read Scan instruction.
   b. Scan out of the chip the first word of the data.
   c. Repeat step b for additional data words for consecutive addresses.

Flash Data Erase

Pages, sections or the entire flash device (special erase) can be erased. If the flash is locked (FLAEN in the Flash Status register is set), a special erase must be performed.

To select and perform a flash erase, perform the following sequence:

1. Enable flash access (if not already enabled).
2. Set the desired erase mode as follows:
   a. Set IR to Flash Control and Status Scan instruction.
   b. Scan in a code of 8100_{16} (page erase), 8200_{16} (section erase) or 8400_{16} (special erase).
3. Set the address as follows:
   a. Set IR to Flash Address Scan instruction.
   b. Scan an address into the memory block to be erased, specifying the correct block.
4. Erase the page or section as follows:
   a. Set IR to Flash Data Erase Scan instruction.
   b. Scan the erase command into the chip with 1_{16} data.
   c. Repeat the scan with 0_{16} data until the Busy status flag indicates erase completion. The Error bit indicates if the erase was successful. In case of an error, the failing address is provided in Flash Address register.
   d. To erase additional pages or sections, repeat the Set Address (3) and Erase (4) operations.
5. Erase the entire flash device (special erase) as follows:
   a. Set IR to Flash Data Erase Scan instruction.
   b. Scan the erase command into the chip with 1_{16} data.
   c. Repeat the scan with 0_{16} data until the Busy status flag indicates erase completion. The Error bit indicates if the erase was successful. In case of an error, the failing address is provided in Flash Address register.
   d. If there is no error indication, repeat steps b and c two more times.

Flash Data Write

Writing data to the flash should be done after properly erasing the page (see “Flash Page Organization and Erase Options” on page 221 for flash write after erase limitations).

To write data, perform the following sequence:

1. Enable flash access (if not already enabled).
2. Set the flash to Program mode as follows:
   a. Set IR to Flash Control and Status Scan instruction.
   b. Scan in a code of 8080_{16} for program enable.
3. Set the address as follows:
   a. Set IR to Flash Address Scan instruction.
   b. Scan an address into the memory address to be written.
4. Write the block as follows:
   a. Set IR to Flash Data Write Scan instruction.
   b. Scan the write data into the chip with the data byte/word and Valid bit set.
   c. Repeat the scan instruction as long as the Valid bit is cleared, checking for the returned Valid status flag to indicate a ‘not busy’ status. The Error bit indicates if an error occurred during the write process of the previous word. On successful write completion, the address is incremented to the next address. In case of an error, the failing address is provided in Flash Address register; further writes are disabled until the error flag is cleared. The exact source of the error is provided in FL_CT_ST register. The error may be cleared using FL_CT_ST register.
5. Repeat the write operation (step 4) for writing data to successive addresses.
6. Check that the write is completed by reading the Flash Status bit until it indicates that the flash is not busy; to do this:
   a. Set IR to Flash Control and Status Scan instruction.
   b. Scan in a code of 8080_{16}, maintaining the program enable.
   c. Check the capture status for completion of the last write (the FMBUSY bit should be cleared).
   d. Repeat steps b and c until the flash is not busy.

Embedded Controller Modules (Continued)

Flash ID Read

To identify the device version, perform the following sequence:

1. Enable flash access (if not already enabled).
2. Read the Device ID (SID and SRID register values), using the Device ID Scan, as follows:
   a. Set the IR to Device ID Scan instruction.
   b. Scan the LF_ID chain.

4.19.5 Test Access Port (TAP)

This section defines the top-level design of the TAP, as illustrated in Figure 81. Subsequent sections provide detailed TAP design requirements.

The TAP includes the following elements:

- TAP signals
- TAP controller
- Instruction Register (IR)
- Data registers (Debugger and Flash interfaces).

The Instruction and data registers have separate shift register-based paths connected in parallel. These registers have a common serial data input and a common serial data output connected to the TAP TDI and TDO signals, respectively.

The TAP controller selects between TDI and TDO as the alternative instruction and data register paths.

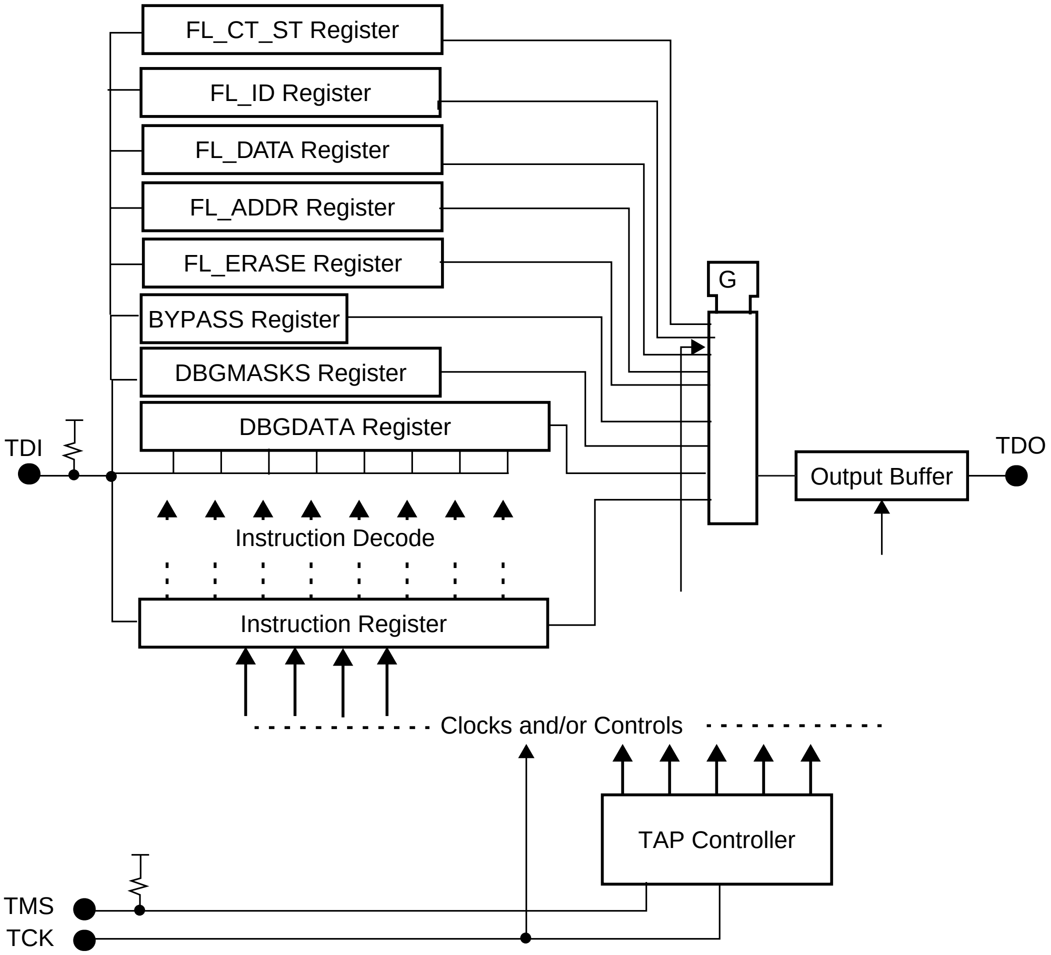


Figure 81. TAP Block Diagram

Embedded Controller Modules (Continued)

Further Information

The TAP is based on the test logic in the IEEE 1149.1b-1994 specification. However, since its purpose is to facilitate communication, it does *not* support IEEE 1149.1b-1194 testing facilities. It is used here to benefit from off-the-shelf bus controller cards and software and potential future enhancements to the test scheme.

This document includes the relevant rules of this specification. See the following documents for further information.

- For further details and examples of the standard, see *IEEE Standard Test Access Port and Boundary-Scan Architecture, May 21, 1990*.
- For further details of test bus chips and equipment, see the relevant manufacturer datasheets and application notes, e.g., *SCANTM Data book, National Semiconductor 400102.*
- For technical background, refer to text books on the subject, for example, *Colin M. Maunder and Rodham E. Tulloss, "The Test Access Port and Boundary-Scan Architecture", IEEE Computer Society Press Tutorial*.

TAP Signals

The TAP interface signals to the JTAG serial bus are: Test Clock Input (TCK), Test Mode Select Input (TMS), Test Data Input (TDI) and Test Data Output (TDO). The TAP controller is reset at Power-Up reset only; see Section 3.2 on page 65 for further information.

TCK. TCK provides the clock for the JTAG serial bus and the TAP controller. Stored-state devices in the TAP maintain their state indefinitely after the signal applied to TCK is stopped.

TMS. The TAP controller next state is set by the TMS value and its current state. The TAP samples the signal presented at TMS on the rising edge of TCK.

Circuitry, fed from TMS, produces the same response to the application of a logic 1 as for a non-driven input.

TDI. This is the data and instructions serial input. The TAP samples the signal presented at TDI on the rising edge of TCK.

Circuitry fed from TDI produces the same response to the application of a logic 1 as for a non-driven input.

When data is shifted from TDI towards TDO, test data received at TDI appears without inversion at TDO following a number of rising and falling edges of TCK. This is determined by the length of the instruction or Test Data register selected.

TDO. This is the data and instructions serial output. The signal driven through TDO changes its state only after the falling edge of TCK.

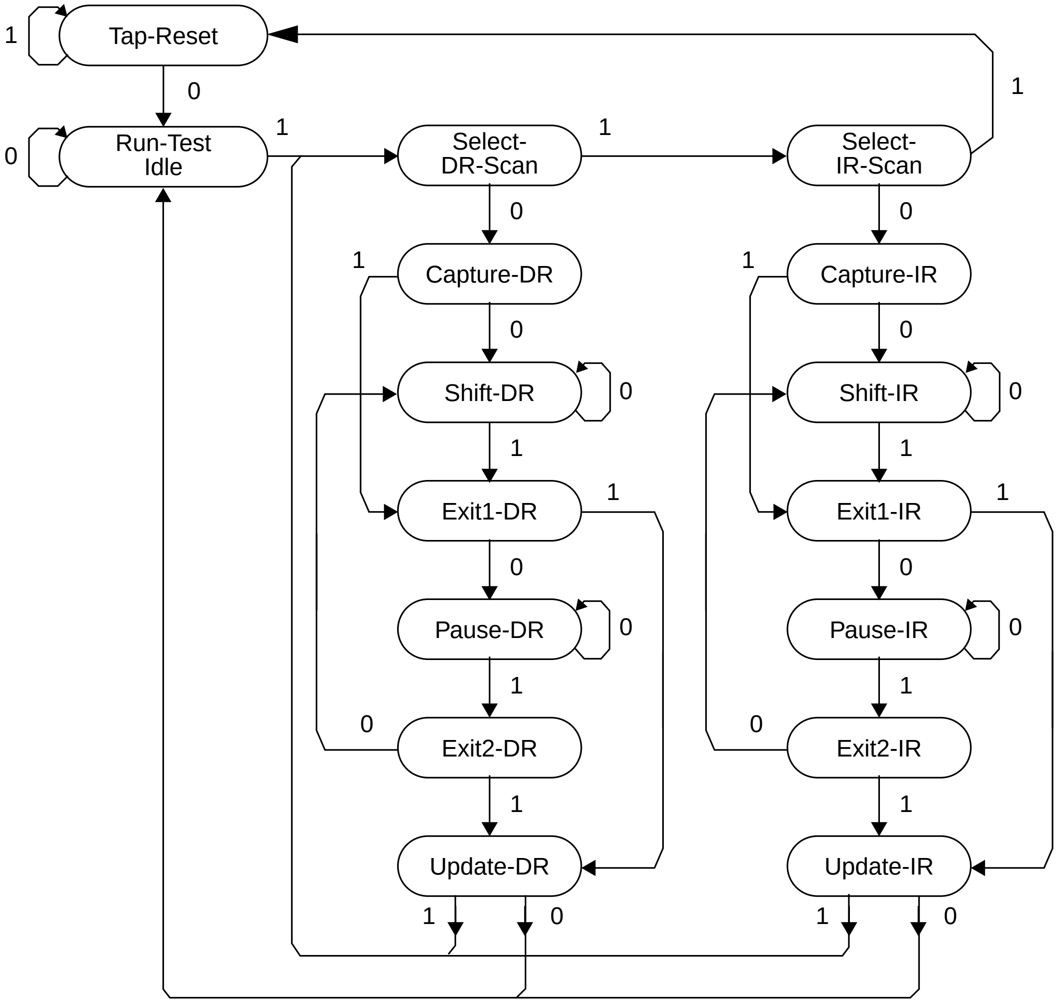
The TDO driver is set to its inactive drive state, except while data or an instruction is being scanned.

TAP Controller

The TAP controller is a synchronous, finite state machine that responds to changes in the TMS and TCK signals and controls the sequence of operations of the PC87591x reset, ISE interrupt control and data link circuitry.

Figure 82 shows the TAP controller state diagram.

Embedded Controller Modules (Continued)



Note: The value for each state transition represents the signal on TMS for a rising edge at TCK.

Figure 82. TAP Controller State Diagram

For a detailed description of the controller states, see the IEEE 1149.1b-1994 specifications.

All state transitions of the TAP controller occur based on the value of TMS on the rising edge of TCK. TAP actions (described in Sections 2.1.3 to 2.1.7) occur either on the rising or falling edge of TCK in each controller state, as shown in Figure 83.

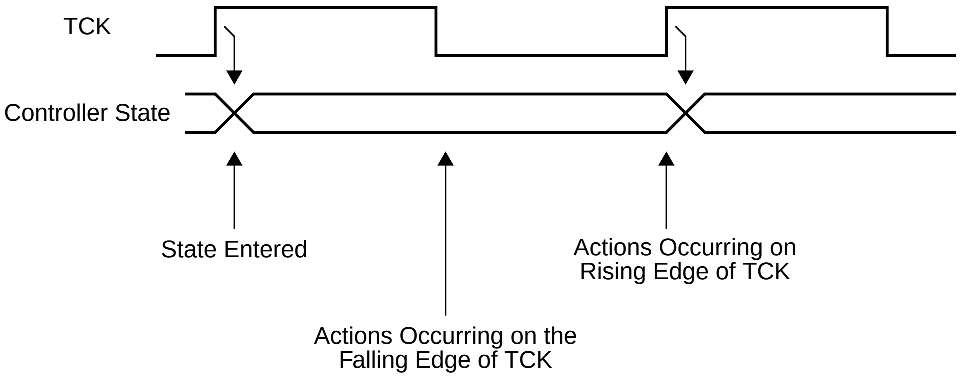

TCK
Controller State
State Entered
Actions Occurring on Rising Edge of TCK
Actions Occurring on the Falling Edge of TCK

Figure 83. Timing of Actions in a Controller State

Embedded Controller Modules (Continued)

TAP Controller Operation

The TAP controller changes state only in response to the following events:

- A rising edge of TCK
- Power-up

The TAP controller generates signals to control the operation of the TAP registers and associated PC87591x reset, ISE interrupt control and data link circuitry.

The TDO output buffer and the circuitry that selects the register output fed to TDO are controlled as shown in Table 37. Changes at TDO, as defined in Table 37, occur on the falling edge of TCK after entry into the state.

Table 37. TAP Operation in Each Controller State

Controller State	Register Selected to Drive TDO	TDO Driver Level
Tap-Reset	Undefined	Inactive
Run-Test-Idle	Undefined	Inactive
Select-DR-Scan	Undefined	Inactive
Select-IR-Scan	Undefined	Inactive
Capture-IR	Undefined	Inactive
Shift-IR	Instruction	Active
Exit1-IR	Undefined	Inactive
Pause-IR	Undefined	Inactive
Exit2-IR	Undefined	Inactive
Update-IR	Undefined	Inactive
Capture-DR	Undefined	Inactive
Shift-DR	Test Data	Active
Exit-DR	Undefined	Inactive
Pause-DR	Undefined	Inactive
Exit2-DR	Undefined	Inactive
Update-DR	Undefined	Inactive

TAP Controller Initialization

Circuitry built into the PC87591x forces the TAP controller into TAP-Reset state at power-up in an asynchronous manner.

The TAP controller must not be initialized by operation of any system input, such as a system reset.

When TMS is equal to 1 for five consecutive TCK cycles, this forces the controller into TAP-Reset state from any state.

4.19.6 TAP Instruction Register

The Instruction Register (IR) allows an instruction to be shifted into the Debugger interface. The instruction selects the mode of operation, the data register to be addressed or both.

In any serial scan operation, the JTAG serial bus controller transmits and receive vectors of the same length. This serial scan chain property is used to transmit status bits from the device to the bus controller in any instruction scan operation. The Instruction register in the PC87591x allows status information generated within the data links to be examined and supports flash programing, as shown in Table 38 on page 257.

Design and Construction

The IR uses a shift register-based design with a parallel input for register cells other than the two nearest to the serial output. An instruction, shifted into the register, is latched at the completion of the shifting process.

The IR includes 12 shift register-based cells capable of holding instruction data.

An instruction that is shifted into the register is latched so that the related changes occur only in the Update IR and TAP-Reset states.

Embedded Controller Modules (Continued)

Data is not inverted between the serial input and serial output of IR.

The IR parallel input status bits, loaded at the Capture IR state of any instruction scan operation, are shown below.

Bit	11 7	6	5	5	3	2	1	0
Name	**MSG_LEN**	**PID**				**RX_BUSY**	**0**	**1**
Reset								

Bit	Description
0	Fixed value (1).
1	Fixed value (0).
2	**RX_BUSY (Receive Busy).** Busy indication from the Rx data link.
6-3	**PID (Processor ID).** Contains the processor ID from the Tx data link. When $\overline{\text{TINT}}$ is set to 1 (not active), PID field is ‘1111’, indicating that the Tx link has no valid data. When $\overline{\text{TINT}}$ is set to 0 (active), PID field indicates the value of PID field in DBGTXLOC register. Note that this value is latched when $\overline{\text{TINT}}$ becomes active and is held until the Rx data is read by the host.
11-7	**MSG_LEN (Message Length).** Contains the Tx message length in words. When the Tx status word PID field is not ‘1111’, the MSG_LEN holds a copy of the MSG_LEN field in DBGTXST register. When the size of this field is less than five bits, the MSG_LEN MSBs in the status word are forced to 0.

Instruction Register Operation

Table 38 shows the behavior of IR in each TAP controller state.

All actions resulting from an instruction terminate when a new instruction is transferred to the parallel output of the IR (i.e., in Update-IR or TAP-Reset states).

All operations of the shift register stages occur on the rising edge of TCK after entry into a TAP controller state.

Table 38. Instruction Register Operation in Each Controller State

Controller State	Shift Register Stage	Parallel Output
TAP-Reset	Undefined	Set to give BYPASS instruction
Capture-IR	Load status data (table 2-2)	Retain last state
Shift-IR	Shift towards serial output	Retain last state
Exit1-IR Exit2-IR Pause-IR	Retain last state	Retain last state
Update-IR	Retain last state	Load from shift register stage
All other states	Undefined	Retain last state

The data present at the parallel output of the IR is latched from the shift register stage on the falling edge of TCK in Update-IR state.

After entry into TAP-Reset state as a result of TAP controller clocked operation, the BYPASS instruction is latched onto the IR output on the falling edge of TCK.

Embedded Controller Modules (Continued)

Instructions

Instructions are entered serially into the Debugger interface logic during an IR scan by using the IR. See Table 39 for the required binary codes.

Table 39. IR Instruction Binary Codes

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Instruction
x	x	x	x	x	x	x	x	x	1	1	1	BYPASS
x	x	x	x	x	x	x	x	x	0	0	0	BYPASS (Reserved for Scan)
x	x	x	x	x	x	x	x	x	1	1	0	BYPASS
0	0	0	0	0	0	0	0	0	0	0	1	Flash Control Scan
0	0	0	0	0	0	0	0	1	0	0	1	Device I/D Scan
0	0	0	0	0	0	0	1	0	0	0	1	Flash Address Scan
0	0	0	0	0	0	1	1	0	0	0	1	Flash Data Read Scan
0	0	0	0	0	1	0	1	0	0	0	1	Flash Data Write Scan
0	0	0	0	0	1	1	1	0	0	0	1	Flash Erase Scan
other									0	0	1	BYPASS
L4	L3	L2	L1	L0	PID3	PID2	PID1	PID0	0	1	0	SCAN_RX
x	x	x	x	x	x	x	x	x	0	1	1	SCAN_TX
x	x	x	x	x	x	x	x	x	1	0	1	SCAN_ABORT_MASK
x	x	x	x	x	x	x	x	x	1	0	0	ASSERT_DBG_RST

Note:

- 1. D0 is the nearest to the serial output.
- 2. “X” means ignore.
- 3. Debugger abort is generated by SCAN_RX when PID=‘1111’.

Data registers not selected by the current instruction do not interfere with the operation of the on-chip system logic or with the selected data registers.

Each instruction enables a single serial data register path to shift data between TDI and TDO in Shift-DR state, as shown in Table 40.

Instruction codes that are not required to control test logic are equivalent to the BYPASS instruction.

BYPASS

The BYPASS instruction operates the BYPASS register. This register contains a single shift register stage and provides a minimum-length serial path between the TDI and TDO pins of a component when no test operation is needed for that component. This allows more rapid movement of test data to and from other board components that are required to perform test operations.

The BYPASS instruction selects BYPASS register to be connected for serial access between TDI and TDO in the Shift_DR state.

If the BYPASS instruction is selected, all other data registers continue their normal functionality.

Flash Support Instructions

Flash Control Scan

The Flash Control Scan instruction switches the data scan path to FL_CT_ST register. During this instruction, the length of the scan chain is 16 bits. In this case, the Control and Status register of the flash can be scanned. In the Capture-DR state of the TAP controller, the Status register is latched into the FL_CT_ST register chain. During the Update-DR state, the Control register is loaded with the contents of FL_CT_ST register. Some bits are shared between the two registers and operate as read/write or read/write 1 to clear.

Device I/D Scan

The Device I/D Scan instruction switches the data scan path to FL_ID register. During this instruction, the length of the scan chain is 32 bits. In this case, SID and REVID registers of the SuperI/O are read and their contents are sampled in the Capture-DR state. No updates are made during the Update-DR state.

Flash Address Scan

The Flash Address Scan instruction switches the data scan path to FL_ADDR register. During this instruction, the length of the scan chain is 32 bits. In this case, the Address register of the flash can be scanned. In the Capture-DR state of the TAP controller, the flash address is latched into the FL_ADDR chain. During the Update-DR state, if the Block ID field is other than ‘0000’, Flash Address register is loaded with a new value from FL_ADDA and the Valid bit in the data register is cleared. After Flash Address register is loaded, a read operation from the array is performed, the result is stored in the Flash Read Data buffer, and the Data Read buffer Valid bit is set.

Flash Data Read Scan

The Flash Data Read Scan instruction switches the data scan path to FL_DATA register. During this instruction, the length of the scan chain is 18 bits. In the Capture-DR state, the Flash Read Data buffer and the two status bits are loaded into the shift register and the Valid bit is cleared. The Error bit indicates any error that occurred during the read operation.

If the Valid bit read was cleared, the data and error fields should be ignored. If the Valid bit read was set, the address is incremented after the Capture-DR state. A data read from the memory is performed, placing the data in the Flash Read Data buffer and setting the Valid bit.

When the flash address is in the flash main block or Information block, the data read is one word and the address increment is two.

Flash Data Write Scan

The Flash Data Write Scan instruction switches the data scan path to FL_DATA register. During this instruction, the length of the scan chain is 18 bits. In the Capture-DR state, the Flash Read Data buffer and the two status bits are loaded into the shift registers and the Valid bit is cleared. The Error bit indicates any error that occurred during the write operation.

If the Valid bit read was cleared, the data and error fields should be ignored. If the Valid status (RO) bit is cleared and the Valid control bit is set (WO) during the Update-DR state, the data is loaded into the Flash Write Data buffer, starting the data write sequence to the flash. When the write is completed, a flash read is performed, placing the data in the Flash Read Data buffer. The data is then compared to the Flash Write Data buffer; if the two are different, the Program Error flag in the Control Status register and the Error flags in the data register are set. The Valid bit read (RO) indicates when another word may be written to the data write buffer.

When the flash address is in the flash main block or Information block, the data written is one word (16 bits) and the address increment is two.

FMBUSY is set at the beginning (Update-DR) of the write process; it is cleared when the write to the flash is completed. The Flash error flags are updated in case an error occurs during the write operation.

Flash Erase

The Flash Erase instruction switches the data scan path to FL_ERASE register. During this instruction, the length of the scan chain is two bits. In the capture-DR state, the Flash Busy and Error bits are read. If the Busy bit is cleared, the Error bit indicates any failure in the erase process.

During the Update-DR state of this instruction, the Start (WO) bit is loaded into the Control register. If the Start bit and any of the Erase bits are set, the respective erase process starts. The Flash Address and Block ID are used to indicate the page or sector to erase in the Page and Sector Erase commands, respectively.

When the erase is completed, the Busy bit is set. The Error flag indicates any failure that occurs in the flash erase procedure.

The FMBUSY and FMERR bits in Flash Status register are updated in parallel to the Erase instruction Busy and Error flags.

Debugger Interface Instructions

SCAN_RX

The SCAN_RX instruction switches the data scan path to DBGDATA register. The DBGDATA length is set to L0 to L4; see “Debug Data Register (DBGDATA)” on page 261. The result of the Capture-DR state of the TAP controller is unpredictable. A parallel load of data from DBGDATA register to the DBGRXD Rx data buffer is done in Update-DR state.

The controller sets the RX_BUSY indication in DBGRXST register to 1 in Update-DR state. The PID and message fields of the SCAN_RX instruction are available for read access, through the peripheral bus, from DBGRXST register. The ISE interrupt control block asserts the ISE interrupt and RX_i bit in DBGISESRC register, according to the PID index.

DEBUGGER ABORT

This operation has no dedicated operation code. It is performed using the SCAN_RX instruction with the PID field is ‘1111’. Following SCAN_RX mode, the ISE interrupt control block asserts ISE interrupts, together with ABORT_i bits in DBGISESRC register, according to the MASKS values in DBGMASKS register. The assertion is triggered during the TCK rising edge during Update-IR state. In this case, there is no RX_BUSY indication and no change in the contents of DBGRXST register (i.e., this format of SCAN_RX may be issued with a busy Rx data link).

SCAN_TX

The SCAN_TX instruction switches the data scan path to DBGDATA register. In Capture-DR state, DBGDATA register captures the values from the Tx data buffer DBGTXD. No parallel load is performed in Update-DR state. The length of DBGDATA register is set to the value of MSG_LEN in DBGTXST register (see “Debug Data Register (DBGDATA)” on page 261). No parallel load is performed in Update-DR state.

The Controller sets DBGTXLOC to all 1s and releases $\overline{\text{TINT}}$ on Update-DR state. This operation clears the semaphore and enables the data link for a new transaction. $\overline{\text{TINT}}$ is asserted, as described in “TX Data Link” on page 250.

Embedded Controller Modules (Continued)

SCAN_ABORT_MASK

The SCAN_ABORT_MASK instruction switches the data scan path to ABORT_MASK register. In Capture-DR state, a parallel update of the shift register occurs with the DBGMASKS values. Parallel load is performed in Update-DR state to update the DBGMASKS values.

ASSERT_DBG_RST

A Debugger reset is asserted in Update-DR state. Serial data is switched to the BYPASS register.

4.19.7 TAP Data Registers, Debugger Interface

Bit Arrangement and Mapping

Figure 84 shows the bit allocation for a parallel load operation between any pair of serial shift registers and a peripheral bus addressable register.

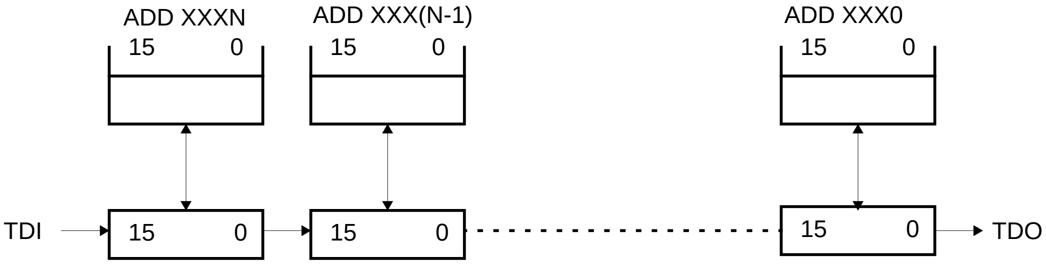


Figure 84. Bit Allocation Arrangement

Functionality in Various TAP Controller States

When data is shifted through a data register, data applied to TDI appears without inversion at TDO following an appropriate number of TCK transitions, when the TAP controller is in Shift-DR state.

The data register connected between TDI and TDO shifts data one stage towards TDO after each rising edge of TCK in Shift-DR state.

In TAP-Reset state, all data registers either perform their system function (if one exists) or do not interfere with the operation of the on-chip system logic.

If, in response to the current instruction, a data register loads data from a parallel input, the data is loaded on the rising edge of TCK following entry into Capture-DR state.

If the data register connected between TDI and TDO in response to the current instruction is provided with latched parallel data outputs, the data is latched into the parallel output buffers on the falling edge of TCK, during Update-DR or Run-Test/Idle states, as appropriate.

If, in response to a current instruction, no operation of a selected data register is required in a given controller state, the register retains its last state unchanged.

When the TAP controller state machine is in TAP-Reset state during Power-Up reset, IR register is reset.

Data registers that are not selected by the current instruction are set to perform their functions, as described in Section 4.19; specifically, see “TX Data Link” on page 250, “Debugger Reset Circuit” on page 250 and “ISE Interrupt Control” on page 250.

Table 40. Data Register Operation in Each Controller State

Controller State	Action
Capture-DR	Load data at parallel input into shift register stage. Parallel output registers, or latch, retains last state.
Shift-DR	Shift data towards serial output. Parallel output register, or latch where provided, retains state.
Exit1-DR Exit2-DR Pause-DR	Retain last state.
Update-DR	Load parallel output register or latch from the shift register stage. Shift register stage retains state.
All other controller states	Registers that have a parallel output maintain the last state of this output; otherwise undefined.

Embedded Controller Modules (Continued)

Debug Bypass Register (BYPASS)

The BYPASS register provides a minimum length serial path for data movement between TDI and TDO. This path can be selected when no other data register needs to be accessed.

The BYPASS register consists of a single shift register stage.

When the current instruction selects the BYPASS register for inclusion in the serial path between TDI and TDO, the shift register stage is set to 0 on the rising edge of TCK following entry into Capture-DR state. The BYPASS register is accessed from the JTAG serial bus only.

Debug Data Register (DBGDATA)

This register is the shift register element of DBGRXD and DBGTXD. It is accessed from the JTAG serial bus only. Figure 85 shows the parallel load data scheme.

The DBGDATA register consists of shift register stages according to the data buffer length. The actual length of the data value, in a scan operation, is set before Capture-DR state when the current instruction is SCAN_RX or SCAN_TX. In the first case, it is set according to SCAN_RX L0 to L4; in the latter case, it is set according to the value of MSG_LEN in DBGTXST register.

The actual length of the register is 16*(length+1), where “length” is the binary positive number created by the length field (with L4 as MSB).

A length longer than the maximum data buffer is mapped to maximum length. A length shorter than the maximum data buffer results in loading the data to/from the smallest addresses in the data buffer. Maximum length for this design is 128 shift register stages. Note that TDO is always fixed; the TDI “insertion-point” changes according to the actual length.

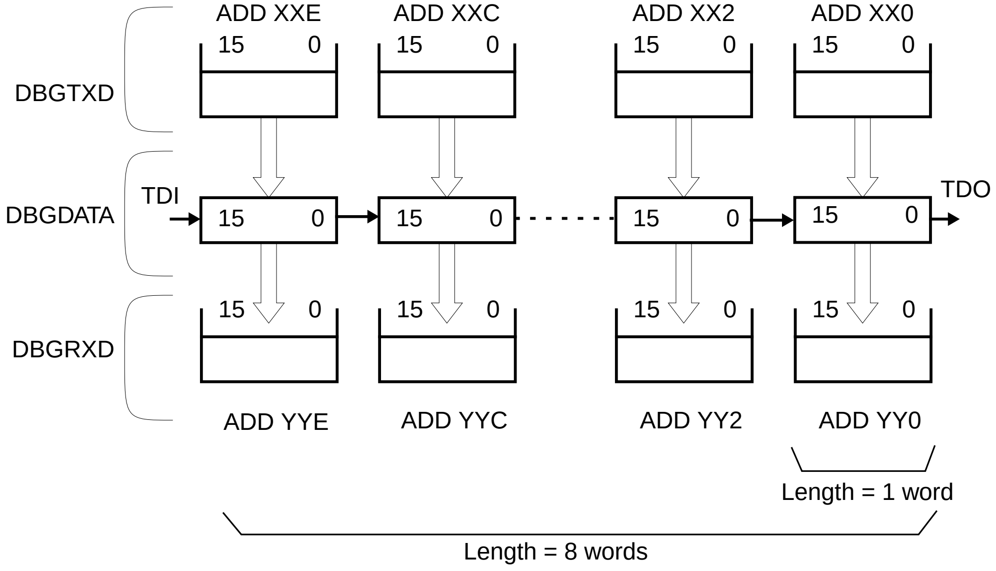


Figure 85. DBGDATA Connection to the Data Links

Debug Abort Mask Register (DBGMASKS)

This is a serial shift register, with parallel output and parallel input (for read and write of ABORT_MASK register), accessed by the JTAG serial bus while the SCAN_ABORT_MASK instruction is set to TAP IR. The register is not addressable from the peripheral bus. The non-reserved bits of ABORT_MASK register are preset to 1 on Power-Up reset.

The ABORT_MASK register consists of shift register stages with 16 bits. Bits 1 to 15 are reserved and should be written with 0. A bit value of 1 enables the processor to abort; a bit value of 0 disables it.

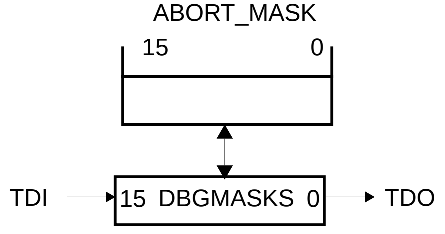


Figure 86. ABORT_MASK and DBGMASKS Register Interaction

4.19.8 TAP Data Registers, Flash Interface

This section describes the flash interface registers available to the host through the flash interface. The bit ordering is as described in “TAP Data Registers, Debugger Interface” on page 260.

Flash Control and Status Register (FL_CT_ST)

FL_CT_ST is a shift register that enables reading from the Flash Status register and writing to the Flash Control register. The Flash Status register is read in Capture-DR state; the Flash Control register is updated in Update-DR state. The connection of the register to the serial chain is shown in Figure 87. The Control and Status registers are reset at Power-Up and TAP resets only.

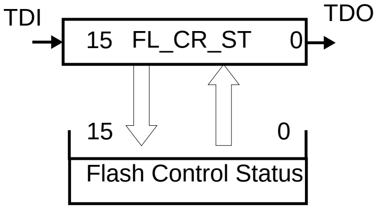


Figure 87. Flash Control and Status Register

Flash Status

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**FMPRGE**	**BLFF**	**FLAEN**	**FMDERR**	**FMBFULL**	**FMBUSY**	**FMPERR**	**FMEERR**
Reset	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8
Name	**JTFIE**	**DISVRF**	**Reserved**			**FMSPERE**	**FMSERE**	**FMPERE**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	R/W1C	**FMEERR (Flash Memory Erase Error).** Indicates an error in erasing to the flash. It is cleared by writing 1 to FMEERR bit in the Control register. This bit may be cleared only when FMBUSY bit is cleared (0). 0: No program error detected (default) 1: An error occurred during erase This bit may be cleared only when FMBUSY bit is cleared (0).
1	R/W1C	**FMPERR (Flash Memory Program Error)**. Indicates an error in writing to the flash. It is cleared by writing 1 to FMPERR bit in the Control register. This bit may be cleared only when FMBUSY bit is cleared (0). 0: No program error detected (default) 1: An error occurred during program
2	RO	**FMBUSY (Flash Memory Busy).** 0: The embedded flash is ready to accept operations if no error flag is set 1: The embedded flash is not accessible because the PC87591x clocks are not ready or the embedded flash is busy programing or erasing
3	RO	**FMBFULL (Flash Double-Buffer Full).** 0: There is at least one empty word in the flash double buffer 1: The double-buffer is full

Embedded Controller Modules (Continued)

Bit	Type	Description
4	R/W1C	**FMDERR (Flash Data Loss Error).** Indicates that a double-buffer overrun error occurred. The bit is cleared by writing 1 to FMDERR bit in Control register. It may be cleared only when FMBUSY bit is cleared (0). 0: No data loss detected (default) 1: A write to a full buffer was detected
5	RO	**FLAEN (Flash Access Enabled).** 0: The flash memory protection is set; read, write and erase (except special erase) are disabled 1: The flash memory access through the JTAG serial bus is enabled
6	RO	**BLFF (Blank Flash Flag).** 0: The Blank Flash flag is cleared and the core is allowed to execute from it 1: The Blank Flash flag is set and the core is kept in reset, even when flash access is enabled
7	R/W	**FMPRGE (Flash Memory Program Enable).** 0: Programing the flash memory using JTAG write instructions is disabled (default) 1: Programing the flash memory using JTAG write instructions is enabled
8	R/W	**FFMPERE (Flash Memory Page Erase Enable).** 0: Flash page erase disabled (default) 1: Flash page erase using the JTAG Erase instruction is enabled
9	R/W	**FMSERE (Flash Memory Sector Erase Enable).** 0: Flash sector erase disabled (default) 1: Flash sector erase using the JTAG Erase instruction is enabled
10	R/W	**FMSPERE (Flash Memory Special Erase Enable).** 0: Flash special erase operation disabled (default) 1: Flash special erase operation using the JTAG Erase instructions is enabled
13-11		**Reserved.**
14	R/W	**DISVRF (Disable Verify).** 0: Enables verification of flash contents after program or erase (default) 1: Disables the automatic verification of flash contents after program or erase; this speeds up the program or erase speed.
15	R/W	**JTFIE (JTAG To Flash Interface Enabled).** 0: Flash access through the JTAG is disabled. Write of 1 to any other bit in this register is ignored, all error flags are cleared, and all other JTAG access flash transactions are ignored (i.e., Capture-DR returns invalid data, Update-DR is ignored) (default). 1: Interface to the flash through the JTAG is enabled

Flash Identification Register (FL_ID)

FL_ID is a shift register that contains device identification information. This information can help to verify the use of the correct flash programing algorithm and to match downloaded information to the device properties.

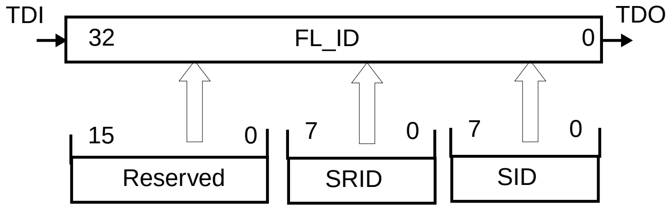


Figure 88. Flash Identification Register

Type: Read Only

Bit	31	30	29	28	27	26	25	24	23	2	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**																**SRID**								**SID**							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ID values															

Embedded Controller Modules (Continued)

Bit	Description
7-0	**SID (SuperI/O Identification).** Returns the value of SID register (see “SuperI/O ID Register (SID)” on page 344)
15-8	**SRID (SuperI/O Revision ID).** Returns the value of SID register (see “SuperI/O Revision ID Register (SRID)” on page 346)
31-16	**Reserved.**

Flash Address Register (FL_ADDR)

FL_ADDR is a shift register that enables reading from and writing to flash addresses. The address includes two fields: the Flash Address and the Block Identifier (main block or information block). In Capture-DR state, the next address for read and/or program is read. In case of a program or page erase fail, the offset within the section of the failing memory location is returned. In Update-DR state, the address register is loaded with a new address. The connection of the register to the serial chain is shown in Figure 89. The address register is reset at Power-Up and TAP resets only.

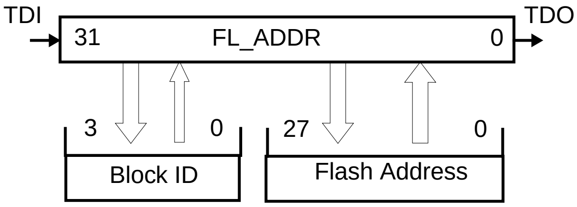


Figure 89. Flash Address Register

Type: R/W

Bit	31	30	29	28	27	26	25	24	23	2	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Block ID**				**Flash Address**																											
Reset	0				0																											

Bit	Type	Description
27-0	R/W	**Flash Address.** Holds the flash address in bytes, starting from address 00000_{16}. The flash last address depends on the block size. All addresses must be word aligned. Note that the field may hold values larger than the actual flash size. Addresses beyond the flash size are reserved and should not be specified.
31-28	R/W	**Block ID.** Holds an ID for a specific block. The ID field values are: **Bits 31 30 29 28** — **Description** 0 0 0 0: For Capture-DR, reserved; for Update-DR, no load of Address and Block ID registers (default) 0 0 0 1: Flash Main Block 0 0 1 0: Flash Information Block Other: Reserved

Flash Data Register (FL_DATA)

FL_DATA is a shift register that enables scanning data for flash read and write. This register is used during both flash programing and flash read. It includes two fields: a 16-bit data field and two Control/Status bits. In Capture-DR state, data read from the array (for either a read instruction or as a confirmation of a write cycle) is loaded from the Flash Read buffer into FL_DATA register. The Valid and Error bits indicate if program and read operations are successfully completed. In Update-DR state, data is loaded into the Flash Write Data buffer if the Valid bit in the data shifted in is set. The connection of the register to the serial chain is shown in Figure 89. The address register is reset at Power-Up and TAP resets only.

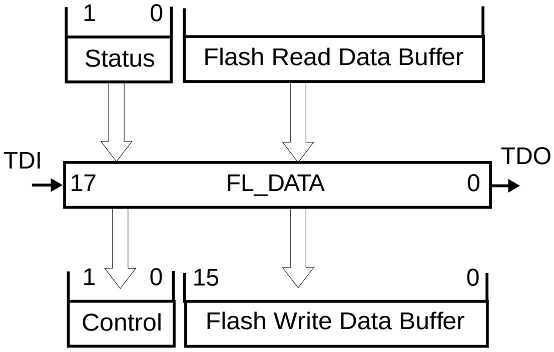


Figure 90. Flash Data Register

Type: Varies per bit

Bit	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Error**	**Valid**	**Data**															
Reset																		

Bit	Type	Description
15-0	R/W	**Data.** In read, returns the data read from the flash memory. Following a previous read instruction or a set address, the data is read from the array with the currently held address. Following a previous write instruction, the data is read back from the array as part of the write verification. During write instruction, data scanned into this field, if the Valid bit is set, is loaded into the Flash Write Data buffer and used for the next flash programing action.
16	RO WO	**Valid.** Indicates that the data read or written is valid. On Read (Capture-DR): 0: Data is not valid, since the read was not completed or the write buffer is not ready to accept new data. 1: Indicates that data in the Data Read buffer is valid. This data may be the result of a read (following a previous read instruction) or the result of a failed write. Note that when a 1 is read from this bit, it is automatically cleared to 0. In a read instruction, a read of 1 triggers a new read operation from memory with the successive address. During flash write sequences, indicates that the write data buffer is ready to accept new data; the bit is cleared when new data is written to the buffer. On Write (Update-DR): This bit must be written with 0 during a read instruction or when the flash is not ready to accept new data (i.e., when the Valid bit in the Capture-DR is cleared). 0: Data is not valid. Do not update the Flash Data Write buffer. 1: Indicates that valid data is written to the Flash Data Write buffer. In write instruction, this triggers the flash to be programed with new data.
17	RO	**Error.** Indicates that an error occurred during the last programing cycle. The bit is set when the error is detected. 0: Last write was completed successfully (default) 1: An error was detected during the last write. This may be due to a failed write, a double buffer error or to an attempt to violate restrictions put by the flash contents protection mechanism. For more details about the error type, see “Flash Control and Status Register (FL_CT_ST)” on page 262. When set, the error is cleared by clearing any set error flag in FL_CT_ST register.

Embedded Controller Modules (Continued)

Flash Erase Register (FL_ERASE)

FL_ERASE is a shift register that starts a flash erase operation and returns the status of the erase operation. The type of erase performed depends on the settings in the Control and Status register. The connection of the register to the serial chain is shown in Figure 91. The FL_ERASE register is reset at Power-Up and TAP resets only.

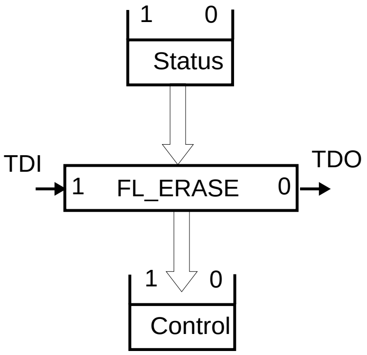


Figure 91. Flash Erase Register

Type: Varies per bit

Bit	1	0
Name	Error	Start/Busy
Reset	0	0

Bit	Type	Description
0	RO WO	**Busy.** On read (Capture-DR), indicates the status of the flash array. 0: Flash is idle and is ready for an erase command to be started 1: Flash is busy executing a previous erase or write command; any command given at this state is ignored. **Start.** On write (Update-DR), starts a flash erase process. 0: Do not start a new erase process 1: Start a new erase process
1	RO	**Error.** Indicates that an error occurred during the last erase cycle. This bit has valid data only when the Busy bit is cleared. 0: Last erase was completed successfully (default) 1: An error was detected during the last erase. This may be due either to a failed erase or protection being set. More details about the error type are available by reading the Control and Status register. When a value of 1 is read, this bit is cleared.

4.19.9 Core Registers, Debugger Interface

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Core Register Map

Mnemonic	Register Name	Type
DBGRXD0-14	Debug Receive Data Registers 0, 2, 4, 6, 8, 10, 12 and 14	RO
DBGRXST	Debug Receive Status Register 15 0	Varies per bit
DBGTXD0-14	Debug Transmit Data Registers 0, 2, 4, 6, 8, 10, 12 and 14	R/W
DBGTXLOC	Debug Transmit Lock Register	R/W
DBGTXST	Debug Transmit Status Register	R/W
DBGTINT	Debug TINT Assert Register	WO
DBGABORT	Debug Abort Generate Register	WO
DBGISESRCA	Debug ISE Source Register A	R/W

Debug Receive Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 (DBGRXD0-14)

DBGRXD0 to DBGRXD14 is a group of eight word-wide read-only registers. The DBGRXD0-14 registers are written from the JTAG serial bus. They can only be read from the peripheral bus. A representative DBGRXDi register is shown below.

Location: Channel 0 - 00 FDC0_{16}
Channel 2 - 00 FDC2_{16}
Channel 4 - 00 FDC4_{16}
Channel 6 - 00 FDC6_{16}
Channel 8 - 00 FDC8_{16}
Channel 10 - 00 FDCA_{16}
Channel 12 - 00 FDCC_{16}
Channel 14 - 00 FDCE_{16}

Type: RO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RX_DATAi**															

Bit	Description
15-0	**Receive Data (RX_DATA).** Data bits 16*i through 16*i+15 of the Rx data link data buffer.

Embedded Controller Modules (Continued)

Debug Receive Status Register (DBGRXST)

DBGRXST is a byte-wide register updated by the TAP controller to reflect the PID and MSG_LEN fields of the SCAN_RX instruction. Bits 1 to 7 of this register are read-only bits; data written to them is ignored. The register format is shown below.

Location: 00 FDE0$_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**MSG_LEN**			**PID**				**RX_BUSY**

Bit	Type	Description
0	R/W	**RX_BUSY (Receive Busy).** Data link busy indication. This bit can be cleared by writing 1 to it. Writing 0 is ignored. 0: Not Busy 1: Busy
4-1	RO	**PID (Processor Index).** A write operation to this field is ignored. **Bit Value (Decimal) Description** 0: Processor ID index 1-14: Invalid 15: All processors abort
7-5	RO	**MSG_LEN.** The message length equals (MSG_LEN+1). Write operations to this field are ignored.

Debug Transmit Data Registers 0, 2, 4, 6, 8, 10, 12 and 14 (DBGTXD0-14)

DBGTXD0 to DBGTXD14 is a group of eight word-wide read/write registers. The DBGTXD0-14 registers are written by a processor from the peripheral bus. A representative DBGTXDi register is shown below.

Location: Channel 0 - 00 FDD0$_{16}$
Channel 2 - 00 FDD2$_{16}$
Channel 4 - 00 FDD4$_{16}$
Channel 6 - 00 FDD6$_{16}$
Channel 8 - 00 FDD8$_{16}$
Channel 10 - 00 FDDA$_{16}$
Channel 12 - 00 FDDC$_{16}$
Channel 14 - 00 FDDE$_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**TX_DATAi**															

Bit	Description
15-0	**TX_DATAi (Transmit Data i).** Data bits i*16 through i*16+15 of the transmit buffer.

Embedded Controller Modules (Continued)

Debug Transmit Lock Register (DBGTXLOC)

DBGTXLOC is a byte-wide read/write register. The PID field of this register is used as a semaphore for locking the Tx channel for a specific processor. If $\overline{\text{TINT}}$ is not active (1), this register is loaded with F_{16} on Warm or Internal reset. If $\overline{\text{TINT}}$ is active (0), it is not affected by a Warm or Internal reset. At Power-Up reset, the register is always loaded with $0F_{16}$. The register format is shown below.

Location: 00 $FDE4_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**				**PID**			
Reset	0	0	0	0	1	1	1	1

Bit	Description
3-0	**PID (Processor Index).** Indicates which processor currently has control over the Tx channel. When $\overline{\text{TINT}}$ becomes active, the value of PID is locked. On the Update_DR state of SCAN_TX instruction, PID is reset to F_{16} after the host reads the data. 0_{16}-E_{16}: Processor ID index. When the PID field holds any of these values, write operations of values other than F_{16} are ignored. F_{16}: Semaphore free indication. When the PID field holds this value, write operations may capture the Tx for processor use.
7-4	**Reserved.**

Debug Transmit Status Register (DBGTXST)

DBGTXST is a byte-wide read/write register. This register is written by a processor to indicate the message length, which is used by the Tx data link to set the length of the serial shift register and by the processor software to define the message length parameter. While $\overline{\text{TINT}}$ is inactive, this register may be written to at any time. While $\overline{\text{TINT}}$ is active, the contents of this register are locked and can be read by the host via the TAP controller status word MSG_LEN field. While $\overline{\text{TINT}}$ is inactive, DBGTXST is cleared on reset. While $\overline{\text{TINT}}$ is active, only Power-Up reset clears DBGTXST. The register format is shown below.

Location: 00 $FDE2_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**MSG_LEN**		
Bit	0	0	0	0	0	0	0	0

Bit	Description
2-0	**MSG_LEN (Message Length).** The number of words in a message equals (MSG_LEN+1).
7-3	**Reserved.**

Embedded Controller Modules (Continued)

Debug TINT Assert Register (DBGTINT)

This is a byte-wide write-only register used to assert $\overline{\text{TINT}}$. The register format is shown below.

Location: 00 FDE6$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**							**ASSERT**

Bit	Description
0	**ASSERT (Assert $\overline{\text{TINT}}$ Control).** Writing 1 to this bit asserts the $\overline{\text{TINT}}$ output. $\overline{\text{TINT}}$ is de-asserted during Update_DR state of SCAN_TX or during Power-Up reset.
7-1	**Reserved.**

Debug Abort Generate Register (DBGABORT)

DBGABORT is a word-wide write-only register, used to generate an ABORT. A processor may generate an ABORT to a processor with a PID index of i by modifying its P_i bit to 1. Writing 0 to P_i does not result in ISE assertion. The register format is shown below.

Location: 00 FDE8$_{16}$

Type: WO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**								**Res**	**Res**	**Res**	**Res**	**Res**	**Res**	**Res**	**P_0**

Bit	Type	Description
0	WO	**P_0.** The ABORT source activation. 0: Processor ID i does not get an ABORT 1: Processor ID i gets an ABORT
15-1		**Reserved.** Reserved for future expansion.

Debug ISE Source Registers A (DBGISESRCA)

DBGISESRCA is a word-wide read/write register that indicates the ISE sources. The register is cleared on reset. Writing 1 to a bit in this register clears it. Writing 0 to a bit does not change its value. The DBGISESRCA register format is shown below.

Location: 00 FDEA$_{16}$

Type: R/W S

Bit	15	14	13	12	11	10	9	8
Name	**Reserved**	**Reserved**	**Reserved**	**Reserved**	**Reserved**	**Reserved**	**Reserved**	**Reserved**
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**ABORT_0**	**RX_0**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**RX_i.** The ISE source is the Rx data link. Turn on with ISE assertion. Turn off by writing a byte to DBGISESRC containing 1 in the RX_i bit location. 0: Not an ISE source 1: The ISE source is the Rx data link
1	**ABORT_i.** The ISE source is an ABORT request by the debugger or one of the processors. This bit is set by either an abort command from the debugger (SCAN_RX with PID='1111') or by a write of a 1 to the processor's respective bit in DBGISECA register. 0: Not an ISE source 1: The ISE source is an ABORT event
7-2	**Reserved.** These bits are reserved for future expansion.

4.19.10 Usage Hints

JTAG Performance

For best performance, the PC87591x should be the only device in the scan path. Since the Debugger interface is used in the development phase of the product, this should not be a problem. Keep the scan path as short as possible.

JTAG Clock Rate

For reliable communication over the JTAG serial bus, the error probability should be extremely low. This can be achieved by connecting the TAP to the JTAG bus controller card, paying attention to the TCK frequency, signal timing, this cable type and cable length (for long cables, transceivers may be required).

Communication Lockout

Some erroneous operations by the host may cause a lockout of the communication hardware. Examples of such cases are:

- Performing a data scan when the JTAG instruction is SCAN_RX with a PID value of '1111'
- Performing a data scan with a JTAG instruction with a PID value that is not in the system (i.e., PID > 0)
- Performing a SCAN_TX when the PID value is '1111'

A reset instruction is guaranteed to recover from all of these locked cases, but it resets the entire PC87591x.

4.20 DEVELOPMENT SYSTEM SUPPORT

The PC87591x supports code development and debug in the On Board Development (OBD) and Development (DEV) environments. OBD environment is used for debug of the code in the final production board. DEV environment is used in Application Development Boards (ADBs) and ISE systems.

4.20.1 Features

- ISE clipping support via a TRI-STATE strap input (TRIS)

Features available in OBD and DEV environments:

- Debugger interface via the JTAG based Debugger Interface module
- Internal ISE interrupt generation by the Debugger Interface module
- Internal Reset generation by the Debugger Interface module
- Ability to prevent real-time events from interfering with operation of ADB monitor
- Core-integrated hardware breakpoint

Features available in DEV environment:

- Optional use of break line input signal
- Status Information to trace internal activities and implement debug features such as hardware breakpoint and traces
- Use of SRAM in the ADB for fast download of code during development

Feature available in OBD environment:

- Use of on-chip flash for executing the final code

4.20.2 The ISE Interrupt

The core ISE interrupt is an edge-triggered, non-maskable interrupt that is triggered on the rising edge of the Debugger interface output. The ISE interrupt is asserted by the Debugger Interface module for RX or ABORT events to the core.

The ISE interrupt is enabled in DEV and OBD environments when ON bit in DBGCFG register is set to 1; otherwise, it is held inactive.

4.20.3 Break Line and Reset Output Interrupt

The $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ signal is available in DEV environment. It has two functions: $\overline{\text{BRKL}}$ interrupt input and $\overline{\text{RSTO}}$ reset indication. Multiplexing between the two functions is done based on bus activity as defined below.

$\overline{\text{BRKL}}$ Function

The $\overline{\text{BRKL}}$ interrupt input is enabled in DEV environment when ON and BRKLE in DBGCFG register are both set (1); otherwise, it is held inactive.

When $\overline{\text{BRKL}}$ is active during an instruction fetch, it indicates to the core a request to break on the execution of the instruction (if the instruction is to be executed). This enables implementing multiple hardware breakpoints using external hardware.

$\overline{\text{RSTO}}$ Function

$\overline{\text{RSTO}}$ is a pulse output that is driven low when the PC87591x is in reset due to any of its sources (i.e., any reset to the core). This output may be used to set any required defaults in the development system.

$\overline{\text{RSTO}}$ and $\overline{\text{BRKL}}$ Selection

The $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ signal serves as input to the PC87591x whenever there is activity on the bus i.e., while either $\overline{\text{SELIO}}$, $\overline{\text{SEL0}}$ or $\overline{\text{SEL1}}$ are active. If $\overline{\text{SELIO}}$, $\overline{\text{SEL0}}$ or $\overline{\text{SEL1}}$ are inactive, the system must stop driving the $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ signal and hold it high using a pull-up resistor.

When there is no activity on the bus, $\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$ may serve as output. It is driven low during an internal reset. There may be a delay from reset start to the signal being driven low, but it is guaranteed to be low for at least three CLK cycles.

4.20.4 TRIS Strap Input Pin

The TRIS strap input signal is used by ISEs to allow clipping on a PC87591x while it is mounted in the production system.

The TRIS input is sampled at V_{CC} Power-Up reset while the device is in IRE or OBD environments. When TRIS is low (0), the PC87591x acts normally. When TRIS is high (1), all PC87591x outputs, except for DAC outputs and 32KX2, are put in TRI-STATE.

Section 2.3 on page 53 describes the strap input handling.

4.20.5 Freezing Events

The PC87591x can prevent real-time events from interfering with the operation of the ADB monitor and changing the status of the PC87591x. This is done by disabling maskable interrupts and setting FREEZE bit; the FREEZE bit freezes the WATCHDOG counter and disables destructive read operations.

Disabling Maskable Interrupts

Clearing the core I or E bits in PSR register disables the maskable interrupts. The I bit is cleared automatically whenever a trap or interrupt occurs and after reset.

Freezing the WATCHDOG Counter

To freeze the WATCHDOG counter, FREEZE bit in DBGCFG register must be set to 1 on entering a TMON routine (the bit must be cleared before returning to the application). Setting FREEZE bit prevents the WATCHDOG from generating the reset that occurs if WATCHDOG is not cleared on time (see Section 4.10 on page 164). The WATCHDOG counter keeps its value while it is frozen and resumes counting after FREEZE is cleared.

If an application fails to touch the WATCHDOG in time and a reset event is generated before or while FREEZE bit is set, the PC87591x receives the reset.

Disabling Additional Modules

The two MFT16 and the two ACB modules may be frozen by the FREEZE bit. Freezing is enabled when the respective bit in DBGFRZEN register is set; the bits can be set to meet specific needs of different applications.

Disabling Destructive Reads

When FREEZE in DBGCFG register is set (1), destructive reads do not change the system state (i.e., they return the read data but do not clear or set bits or send signals). This allows the debugger to present the values of these bits. NMISTAT is an exception to this rule and is not affected by FREEZE. Core accesses to Host domain registers (using the “Core Access to Host Controlled Modules”) may also be destructive but are not affected by FREEZE. Note that host operations continue without any FREEZE bit impact.

4.20.6 Monitoring Activity During Development

In DEV environment, information is available for monitoring on-chip activities and implementing debug features in the development system.

Bus Status Signals

The Bus Status signals (BST2-0) indicate if a transaction on the core bus was issued and, if so, the type of transaction.

The BST2-0 signals reflect activity on the core bus. For word accesses involving 8-bit expansion memory, the core bus cycle triggers two external bus cycles. The first external bus cycle is flagged as a T1 cycle of the core bus. The second is not flagged as a T1 cycle, i.e., BST2-0 is 000. See Table 41.

Table 41. Core Bus Transaction Encoding

BST	Core Bus Transaction Type
000	Not a T1 cycle, except when the core waits for an interrupt following WAIT instruction execution
001	Core waits for an interrupt following WAIT instruction execution
010	T1 of an interrupt acknowledge bus cycle
011	T1 of a data transfer of a non-core bus master
100	T1 of a sequential instruction fetch
101	T1 of a non-sequential instruction fetch
110	T1 of a core data transfer
111	T1 of an exception data transfer

Embedded Controller Modules (Continued)

Transaction Effect on the External Bus

The following core bus transactions are reflected on the external bus:

- Accesses to external zones of expansion memory, off-chip base memory and accesses that use the I/O Expansion protocol are indicated by the active state of the $\overline{\text{SEL0}}$, $\overline{\text{SEL1}}$ and $\overline{\text{SELIO}}$ signals, respectively, and are described by the address and data buses and the status signals, BST2-0.
- Accesses to on-chip memories and peripheral modules can be observed using the “Core Bus Monitoring Bus Cycles” (see “Core Bus Monitoring” on page 82). They accesses are indicated by an inactive state for the $\overline{\text{SEL0}}$, $\overline{\text{SEL1}}$ and/or $\overline{\text{SELIO}}$ signals. They are described by addresses A0-19, the byte-enable BE0-1 signals, the CBRD signal and the BST2-0 signals.
- BE0 is high when a lower memory byte (a byte in an even address) is accessed. BE1 is high when a higher memory byte (a byte in an odd address) is accessed.
- CBRD is high when the transaction is a read operation and low when it is a write operation.

Pipe Status Signals (PFS and PLI)

The PFS indicates the completion of an instruction in the core. The Pipe Long Instruction (PLI) signal indicates the size of the completed instruction, where 0 = word instruction and 1 = double-word instruction (see Figure 92). If an instruction flushes the pipeline, the fetch for the next instruction (BST=101) is issued during the cycle following the instruction’s PFS, or later.

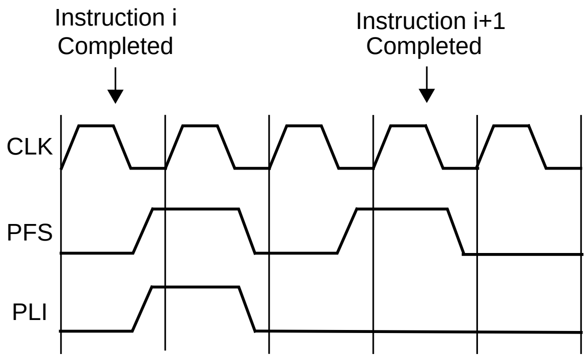


Figure 92. Pipe Status Signal (PFS and PLI)

4.20.7 On-Chip Hardware Breakpoint

The core provides two types of hardware breakpoints:

- Address Match - Detection of a matched address for the current executed instruction (PC value)
- Data Match - Detection of a read or write transaction for a matched memory location

For a detailed description of the core breakpoint mechanism, refer to the CR16B User Manual.

4.20.8 CR16B Development Support Registers

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

CR16B Development Support Register Map

Mnemonic	Register Name	Type
DBGCFG	Debug Configuration	R/W
DBGFRZEN	Debug Freeze Enable	R/W

Debug Configuration Register (DBGCFG)

The DBGCFG register controls the configuration of debug support features. On reset, DBGCFG is cleared (0).

Only the software development tools may access DBGCFG. This enables application software to be binary compatible in all environments.

Location: 00 FF16$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**BRKLE**	**FREEZE**	**ON**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**ON.** 0: In IRE environment, this bit is always cleared to 0; any data written to it is ignored. The ON bit becomes read-only when DBGL bit in DCR register in the core is set. 1: In OBD and DEV environments, enables the following debug support features: - ISE interrupt signal - Use of other bits in DBGCFG register
1	**FREEZE.** 0: No effect 1: When ON is 1, the WATCHDOG timer is stopped from counting. All bits that use destructive reads (i.e., bits set or cleared by read operations and other events triggered by reads) become indifferent to reads. An exception is NMISTAT register, which is always affected by reads. The DBGFRZEN register controls the impact of FREEZE bit on a group of modules, enabling the group to be indifferent to the FREEZE being set. FREEZE has no effect when ON is 0.
2	**BRKLE (Break Line Enable).** 0: Break Line input is ignored by core 1: In DEV environment, when BRKLE is set, the $\overline{\text{BRKL}}$ input signal is passed to the core
7-3	**Reserved.**

Debug Freeze Enable Register (DBGFRZEN)

The DBGFRZEN register enables the freeze operation to be performed on specific modules during debug. When the relevant bit is set, it enables the freeze of activities in the respective module when both FREEZE and ON are set in DBGCFG register. On reset, DBGFRZEN is loaded with FF_{16}.

Location: 00 $FF18_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**HIFEN**	**USARTFEN**	**ACB2FEN**	**ACB1FEN**	**MFT2FEN**	**MFT1FEN**
Reset	1	1	1	1	1	1	1	1

Bit	Description
0	**MFT1FEN (MFT16 1 Freeze Enable).** 0: FREEZE in DBGCFG register has no effect on the MFT16 1 1: Freezes the MFT16 1 when FREEZE is set
1	**MFT2FEN (MFT16 2 Freeze Enable).** 0: FREEZE in DBGCFG register has no effect on the MFT16 2 1: Freezes the MFT16 2 when FREEZE is set
2	**ACB1FEN (ACB1 Freeze Enable).** 0: FREEZE in DBGCFG register has no effect on the ACB 1 interface 1: Freezes the ACB 1 interface when FREEZE is set
3	**ACB2FEN (ACB2 Freeze Enable).** 0: FREEZE in DBGCFG register has no effect on the ACB 2 interface 1: Freezes the ACB 2 interface when FREEZE is set
4	**USARTFEN (USART Freeze Enable).** 0: FREEZE in DBGCFG register has no effect on the USART interface 1: Freezes the USART interface when FREEZE is set
5	**HIFEN (Host Interface Freeze Enable).** 0: FREEZE in DBGCFG register has no effect on the Host Interface module 1: Freezes the Host interface when FREEZE is set
7-6	**Reserved.**

5.0 Host Controller Interface Modules

This chapter describes functions that serve as an interface between the host and core domains. The functions are:

- Keyboard and Mouse Controller Interface (legacy 60_{16}, 64_{16}); see Section 5.1
- Two Power Management (PM) channels compliant with ACPI EC specifications; see Section 5.2 on page 285
- Shared Memory mechanism; see Section 5.3 on page 297
- Core Access to SuperI/O modules; see Section 5.4 on page 313
- Mobile System Wake-Up functions; see Section 5.5 on page 318

5.1 KEYBOARD AND MOUSE CONTROLLER INTERFACE

The PC87591x supports a standard Keyboard and Mouse Controller interface. This interface implements legacy ports 60_{16} and 64_{16}.

5.1.1 Features

- Intel 8051SL-compatible Host interface
  - 8042 KBD standard interface (ports 60_{16}, 64_{16})
  - Legacy IRQ: IRQ1 (KBD) and IRQ12 (mouse) support
  - Fast Gate A20 and Fast Reset via firmware
- Configured using two logical devices: Keyboard and Mouse

5.1.2 General Description

The PC87591x supports a keyboard/mouse communication channel that uses the standard command/status register and data registers. It uses either polling- or interrupt-driven communication schemes with the host and/or core. The hardware is designed to allow a race-free interface between the host and the PC87591x.

The keyboard and mouse channel consists of three registers:

- DBBOUT - can be written by the core and read by the host processor.
- DBBIN - can be written by the host processor and read by the core.
- STATUS - can be read by both core and host processors. It has five bits (2, 4-7) that are written by the core. Three other bits are controlled by the hardware to indicate the status of DBBIN and DBBOUT registers.

Host Addresses

The host processor accesses the PC87591x Keyboard/Mouse Host Interface registers at two addresses in the host address space. These addresses are defined by two internal chip-select signals specified in the PC87591x host configuration registers; see Section 6.1.10 on page 348). Legacy settings of these addresses are 60_{16} and 64_{16} for the status/command and data registers, respectively.

Table 42 describes the register mapping to the host processor I/O space. For simplicity, the Host Interface module specification refers to the legacy addresses.

Table 42. Mapping of the Host Interface Registers to the Host Processor

Port	Legacy Address	Internal Chip Select	Type	Register Name	Mnemonic
Keyboard and Mouse	60_{16}	Keyboard/Mouse Data	Write	Data	DBBIN (A2=0)
	64_{16}	Keyboard/Mouse Command	Write	Command	DBBIN (A2=1)
	60_{16}	Keyboard/Mouse Data	Read	Data	DBBOUT
	64_{16}	Keyboard/Mouse Command	Read	Status	STATUS

Core Interrupts

The Host Interface module generates four level (high) interrupts to the core ICU. These can be used by the firmware for interrupt-driven control of the keyboard/mouse and/or PM channels.

Host Interrupts

The PC87591x Host Interface supports two interrupts to the host processor: Keyboard interrupt (legacy IRQ1) and Mouse interrupt (legacy IRQ12). These interrupts may be controlled by hardware according to the host interface buffer status or by the PC87591x firmware toggling the bit value.

5.0 Host Controller Interface Modules (Continued)

The Host configuration module assigns host interrupts to IRQ numbers (see Section 6.1.10 on page 348). These interrupts are IRQ1 and IRQ12 for keyboard and mouse IRQs, respectively.

When IRQ1 and/or IRQ12 are disabled (OBFKIE and/or OBFMIE bits in HICTRL register are cleared), the firmware can control the IRQ1 and/or IRQ12 signals by writing to the signal's respective bit in HIIRQC register.

When IRQ1 and/or IRQ12 are controlled by hardware (OBFKIE and/or OBFMIE bits in HICTRL register are set to 1), interrupts to the host are generated according to the status of Output Buffer Full (OBF) flag.

In normal polarity mode (IRQNPOL in HIIRQC register is set to 0), the PC87591x supports two types of interrupts: Legacy Edge and Level. When an Edge interrupt is selected (IRQM in HIIRQC register is set), the interrupt signal default value is high (1). When an interrupt signal must be sent (i.e., the corresponding OBF flag is set), a negative pulse is generated. The pulse width is determined by IRQM field in HIIRQC register.

When the IRQ signals are set as level interrupts (IRQM in HIIRQC register is set to 0), the interrupt signal is usually low (0) and is asserted (1) to indicate that the respective OBF flag is set. The signal is de-asserted (0) when the output buffer is read (i.e., OBF flag is cleared).

Note that IRQ1 and IRQ12 have the same OBF flag but are not asserted together. Either IRQ1 or IRQ12 is set, depending on the internal register written (HIKDO or HIMDO, respectively).

In negative polarity mode (IRQNPOL in HIIRQC register is set to1), the IRQ signal behavior is inverted from the behavior described above.

The PC87591x firmware can read the values of the IRQ1 and IRQ12 signals by performing a read operation from IRQ1B and IRQ12B bits in HIIRQC register.

Figure 93 illustrates the effect of the different control bits on the IRQ signals.

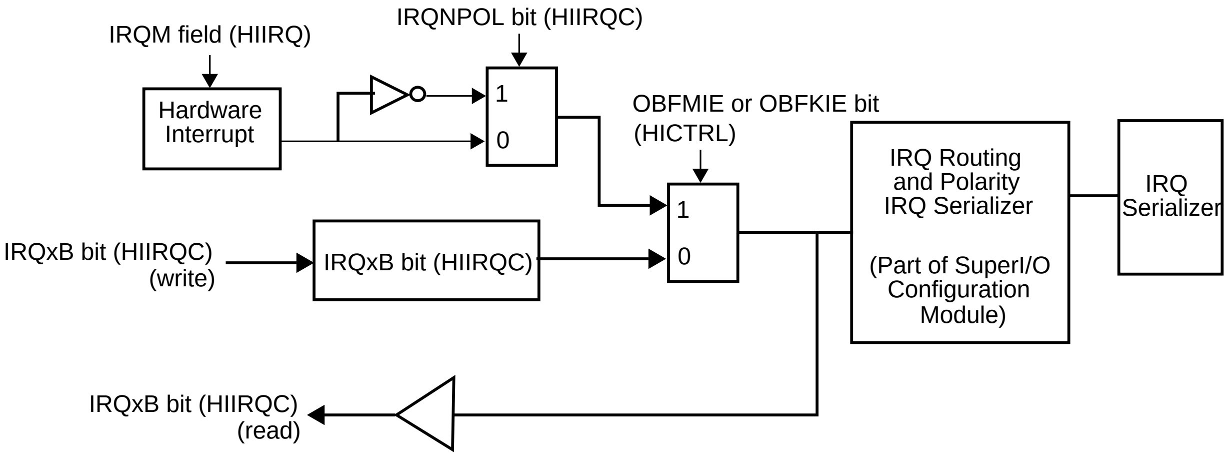

IRQM field (HIIRQ)
IRQNPOL bit (HIIRQC)
Hardware Interrupt
1
0
OBFMIE or OBFKIE bit (HICTRL)
1
0
IRQ Routing and Polarity IRQ Serializer
(Part of SuperI/O Configuration Module)
IRQ Serializer
IRQxB bit (HIIRQC) (write)
IRQxB bit (HIIRQC)
IRQxB bit (HIIRQC) (read)

Figure 93. IRQx (IRQ1, IRQ11 or IRQ12) Control Diagram

Keyboard/Mouse Channel (60_{16}, 64_{16})

The Host Interface of the PC87591x is compatible with the legacy 8042 host interface. It is based on two registers: Command/Data and Status. The Host Interface logic generates interrupts to the host processor and core according to the status of the input and output data buffers. Figure 94 provides a schematic description of the Host Interface Keyboard/Mouse channel.

5.0 Host Controller Interface Modules (Continued)

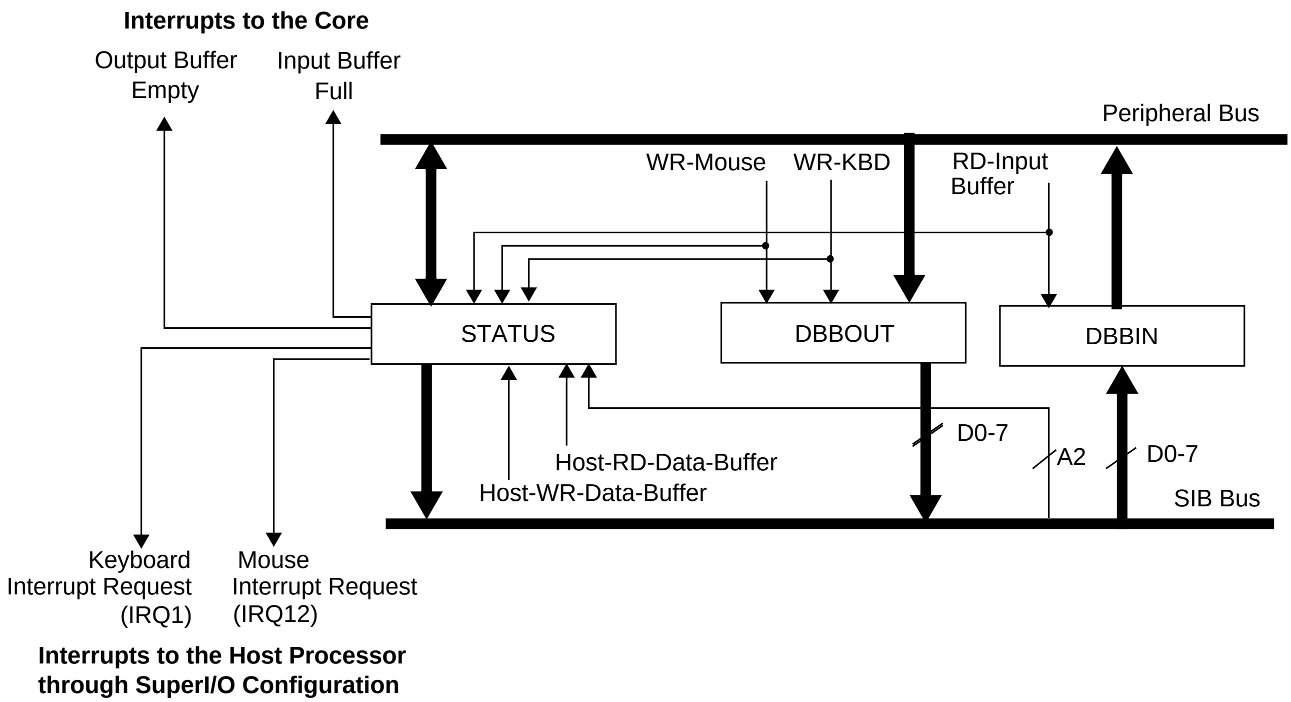


Figure 94. Host Interface Keyboard/Mouse Channel (Ports 60,64) Block Diagram

Status Read

Both the host and the core can read the status of the KBC data buffers. Bits 2 and 4 to 7 can be written by the core. The host processor should read address 64_{16} to obtain the contents of the Status register. The core software can obtain this information by reading/writing the HIKMST register. The format of the Status register is identical for both the host and the core (see “Host Interface Keyboard/Mouse Status Register (HIKMST)” on page 284).

Host Data Write to Host Interface Keyboard/Mouse Channel

The data buffer has two latches; one serves as an input buffer and the other as an output buffer. When writing to address 60_{16} or 64_{16}, the following sequence of events occurs: the data is written to the Data In latch (DBBIN), IBF bit in the Status register is set and bit 3 (A2) in the Status register indicates to the core which address (command or data) was written to. When writing to address 60_{16} (legacy A2=0), bit 3 of the Status register is cleared. When writing to address 64_{16} (legacy A2=1), bit 3 of the Status register is set.

The core identifies that data is present in the input buffer by either polling IBF bit of the Status register or acknowledging an interrupt when the input buffer interrupt is enabled (IBFCIE in HICTRL register is set to 1).

When the input buffer is full, reading the Status register identifies which address was written to (i.e., check A2 of HIKMST register). The core can then read the data from the input buffer (HIKMDI). The IBF status bit is cleared when the data input buffer is read by the core.

Host Data Read from Host Interface Keyboard/Mouse Channel

The output data latch (DBBOUT) is written by the core when it needs to send data to the host. The OBF flag in the Status register (OBF in HIKMST register) is set to indicate that data is available in DBBOUT. DBBOUT should be written only when this bit is cleared.

The PC87591x supports polling and interrupt communication schemes with the host. Both Keyboard interrupt (IRQ1) and Mouse interrupt (IRQ12) are supported.

The core firmware writes to HIKDO register data addressed to the keyboard driver (i.e., generate IRQ1). A write to HIKDO stores the data to DBBOUT and sets OBF bit. If the IRQ1 interrupt is enabled (OBFKIE in HICTRL register is set to 1), it is also sent according to the interrupt mode (IRQM field and IRQNPOL bit in HIIRQC register).

The core firmware writes data addressed to the mouse driver (IRQ12) to HIMDO register. A write to HIMDO stores the data in DBBOUT and sets OBF bit. If the IRQ12 interrupt is enabled (OBFMIE in HICTRL is set to 1); the IRQ12 interrupt is also sent according to the interrupt mode (IRQM field and IRQNPOL bit in HIIRQC register).

5.0 Host Controller Interface Modules (Continued)

The host processor identifies that data is present in the output buffer by either polling the Status register (reading address 64_{16}) or responding to IRQ1 or IRQ12. When this data is available, the host can read it using a read operation from address 60_{16}. Reading from address 60_{16} clears the OBF flag. In addition, when the host interrupt is in level mode (IRQM in HIIRQC register is set to 000_2) and the hardware interrupt is enabled, IRQ1 or IRQ12 are de-asserted (low if IRQNPOL in HIIRQC register is set to 0).

The core can read OBF bit to identify when the output buffer is empty and ready for a new data transfer. When the Output Buffer Empty interrupt to the core is enabled (OBECIE in HICTRL register is set to 1), the interrupt signal to the ICU is set high if the output buffer is empty (OBF=0).

5.1.3 Host Interface Registers

The module has four registers, described below. The base address for each may be configured individually. For legacy operation, they should be configured to 60_{16} and 64_{16} (see Section 6.1.10 on page 348).

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Host Interface Register Map

Offset	Mnemonic	Register Name	Type
60_{16}	DBBOUT	Data Out Buffer	R
64_{16}	STATUS	Status	R
60_{16}	DBBIN	Data In Buffer	W
64_{16}	COMAND	Command In Buffer	W

Data Out Buffer Register (DBBOUT, Legacy 60_{16})

This register allows the host to read DBBOUT register while clearing OBF bit in the Status register. If the host interrupts are level (IRQM in HIIRQC register is 000_2), the interrupt is de-asserted. If the core interrupt on output buffer empty is enabled (OBECIE in HICTRL register is set to 1), reading this register asserts it (high).

Location: As defined in LDN 06_{16} registers, index 60_{16} and 61_{16}

Type: R

Bit	7	6	5	4	3	2	1	0
Name	**Keyboard/Mouse DBBOUT Data**							

Bit	Description
7-0	**Keyboard/Mouse DBBOUT Data.**

Status Register (STATUS, Legacy 64_{16})

This register provides the status of the host interface keyboard channel buffers (DBBIN and DBBOUT) and the value of messages sent by the core using the Status bits to the host. The Status register can also be read by the core as HIKMST. The Status register is cleared (00_{16}) on reset.

Location: As defined in LDN 06_{16} registers, index 62_{16} and 63_{16}

Type: R

Bit	7	6	5	4	3	2	1	0
Name	ST3-ST0				A2	F0	IBF	OBF
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**OBF (Output Buffer Full).** The bit is set when the keyboard/mouse channel's DBBOUT is written to the core (i.e., writing to HIKDO or HIMDO registers). The bit is cleared by a host processor read from the keyboard/mouse channel output buffer (60_{16}).
1	**IBF (Input Buffer Full).** The bit is set when the keyboard/mouse channel's DBBIN is written to the host processor (i.e., writing to either address 60_{16}, data or address 64_{16}, control). The bit is cleared when the core reads the input buffer (HIKMDI).
2	**F0 (Flag 0).** A general-purpose flag that can be set or cleared by the core firmware.
3	**A2 (A2 Address).** Holds the value of the A2 signal in the last write operation of the host to the keyboard/mouse channel's input buffer (i.e., A2=0 for Data In Buffer write and A2=1 for Command In Buffer write).
7-4	**ST3-ST0 (Status).** Four general-purpose flags that can be set or cleared by the core firmware.

Data In Buffer Register (DBBIN, Legacy 60_{16})

This register allows the host to write to DBBIN register while setting Status register bit IBF and clearing Status register bit A2 bit. If the core interrupt on IBF is enabled (IBFCIE in HICTRL register is set to 1), writing to this register asserts it (high).

Location: As defined in LDN 06_{16} registers, index 60_{16} and 61_{16}

Type: W

Bit	7	6	5	4	3	2	1	0
Name	Keyboard/Mouse DBBIN Data							

Bit	Description
7-0	**Keyboard/Mouse DBBIN Data.**

Command In Buffer Register (COMAND, Legacy 64_{16})

This register allows the host to write to DBBIN register while setting IBF and A2 bits in the Status register. If the core interrupt on IBF is enabled (IBFCIE in HICTRL register is set to 1), writing to Data In Buffer asserts it (high).

Location: As defined in LDN 06_{16} registers, index 62_{16} and 63_{16}

Type: W

Bit	7	6	5	4	3	2	1	0
Name	Keyboard/Mouse DBBIN Data							

Bit	Description
7-0	**Keyboard/Mouse DBBIN Data.**

5.0 Host Controller Interface Modules (Continued)

5.1.4 Core Interface Registers

Some register bits affect PM channel 1 to achieve firmware compatibility with the PC87570.

For a summary of the abbreviations used for Register Type, see Section 2 on page 34.

Core Interface Register Map

Mnemonic	Register Name	Type
HICTRL	Host Interface Control	R/W
HIIRQC	Host Interface IRQ Control	R/W
HIKMST	Host Interface Keyboard/Mouse Status	R/W
HIKDO	Host Interface Keyboard Data Out Buffer	WO
HIMDO	Host Interface Mouse Data Out Buffer	WO
HIKMDI	Host Interface Keyboard/Mouse Data In Buffer	RO

Host Interface Control Register (HICTRL)

The HICTRL is used in setting host interface mechanism options. On reset, non-reserved bits of HICTRL are cleared.

Location: 00 FEA0$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**PMICIE**	**PMIOCIE**	**PMIHIE**	**IBFCIE**	**OBECIE**	**OBFMIE**	**OBFKIE**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**OBFKIE (Output Buffer Full Keyboard Interrupt Enable).** 0: IRQ1 is controlled by IRQ1B bit in HIIRQC register 1: Enables the Output Buffer Full interrupt to the keyboard driver of the host processor (IRQ1). The interrupt is triggered by a core write to HIKDO register and sent according to IRQM field and IRQNPOL bit in HIIRQC register.
1	**OBFMIE (Output Buffer Full Mouse Interrupt Enable).** 0: IRQ12 is controlled by IRQ12B bit in HIIRQC register 1: Enables the Output Buffer Full interrupt to the mouse driver in the host processor (IRQ12). The interrupt is triggered by the core write to HIMDO register and sent according to IRQM field and IRQNPOL bit in HIIRQC register.
2	**OBECIE (Output Buffer Empty Core Interrupt Enable).** 0: Interrupt signal is low 1: Enables the Output Buffer Empty interrupt to the core ICU for the keyboard/mouse channel. The interrupt signal is active when the output buffer is empty (i.e., the interrupt signal is set (1) when OBF bit is cleared).
3	**IBFCIE (Input Buffer Full Core Interrupt Enable).** 0: Interrupt signal is low 1: Enables the Input Buffer Full interrupt to the core ICU for the keyboard/mouse channel. The interrupt signal is active when the input buffer is full (i.e., the interrupt signal is set (1) when IBF bit is set).
4	**PMI**HIE (**PMI** Channel 1 Host Interrupt Enable). 0: IRQ11 is controlled by IRQ11B bit in HIIRQC register 1: Enables the Output Buffer Full interrupt of PM channel 1 in PC87570 Compatible mode, to drive the host processor interrupt. The interrupt is noted as IRQ11 and may be routed to any of the IRQs, to SMI or to the SCI events. The interrupt is triggered by a core write to HIPM0DO register and sent according to IRQM field and IRQNPOL bit in HIIRQC register.

Bit	Description
5	**PMOCIE (PM Channel 1 Output Buffer Empty Core Interrupt Enable).** 0: Interrupt signal is low 1: Enables the PM Output Buffer Empty interrupt to the core ICU for PM channel 1 in PC87570 Compatible mode. The interrupt signal is active when the output buffer is empty (OBF bit is cleared in the PM channel status register).
6	**PMICIE (PM Channel 1 Input Buffer Full Core Interrupt Enable).** 0: Interrupt signal is low 1: Enables the PM input buffer full interrupt to the core ICU for PM channel 1 in PC87570 Compatible mode. The interrupt signal is active when the input buffer is full (IBF bit is set in the PM channel status register).
7	**Reserved.**

Host Interface IRQ Control Register (HIIRQC)

The HIIRQC register controls the IRQ signals mode of operation. On reset, HIIRQC is set to 07_{16}.

Location: 00 $FEA2_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**IRQNPOL**	**IRQM**			**IRQ11B**	**IRQ12B**	**IRQ1B**
Reset	0	0	0	0	0	1	1	1

Bit	Description
0	**IRQ1B (Host Interrupt Request 1 Control Bit).** When the IRQ1 signal is configured for direct control by the firmware (OBFKIE in HICTRL register is 0), IRQ1B bit is output to the IRQ1 signal. When read, IRQ1B bit returns the current value of the IRQ1 pin. The IRQ1 signal value an be read regardless of the state of OBFKIE bit.
1	**IRQ12B (Host Interrupt Request 12 Control Bit).** When the IRQ12 signal is configured for direct control by the firmware (OBFMIE in HICTRL register is 0), IRQ12B bit is output to the IRQ12 signal. When read, IRQ12B bit returns the current value of the IRQ12 pin. The IRQ12 signal value can be read regardless of the state of the OBFMIE bit.
2	**IRQ11B (Host Interrupt Request 11 Control Bit).** When PM channel 1 is in PC87570 Compatible mode and its host interrupt is configured for direct control by the firmware (PMHIE in HICTRL register is 0), IRQ11B bit is output to the IRQ11 signal. When read, IRQ11B bit returns the current value of the IRQ11 signal. The IRQ11 signal value can be read regardless of the state of PMHIE bit; see Section 5.2 on page 285 for details about the PM channel 1 interrupt scheme.
5-3	**IRQM (IRQ Mode).** Sets the hardware-controlled IRQ signals to work in Level or Pulse mode and defines the pulse width in Pulse mode. When IRQM = 000_2, the IRQ signals function in Level mode. In this mode, when IRQNPOL = 0, the signal's default value is low, and a high level is set to issue an interrupt (the respective OBF is set). When IRQM ≠ 0, the host interrupts are in Pulse mode. When IRQNPOL = 0, the signal's default value is high and toggles low to issue an interrupt (i.e., when the respective output buffer register is written). The pulse width is as follows: **Bits** **5 4 3** **Pulse Width** 0 0 0: Level Interrupt 0 0 1: 1-Cycle Pulse 0 1 0: 2-Cycle Pulse 0 1 1: 4-Cycle Pulse 1 0 0: 8-Cycle Pulse 1 0 1: 16-Cycle Pulse Other: Reserved

5.0 Host Controller Interface Modules (Continued)

Bit	Description
6	**IRQNPOL (Negative Polarity).** When IRQNPOL is cleared, the IRQ (IRQ1, IRQ11, IRQ12) signal polarity is compatible with the standard ISA bus interface (as specified in the IRQM field). When hardware IRQ generation is enabled (HICTRL register bits OBFKIE for IRQ1 and IRQ12; PMHIE for IRQ11), the interrupt output is inverted if IRQNPOL is set.
7	**Reserved.**

Host Interface Keyboard/Mouse Status Register (HIKMST)

The HIKMST register provides the status of the Host Interface keyboard channel buffers (DBBIN and DBBOUT) and a way for the PC87591x to send status bits to the host. This register can also be read by a host processor read operation from address 64_{16}. On reset, the register is cleared.

Location: 00 $FEA4_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**ST3-ST0**				**A2**	**F0**	**IBF**	**OBF**
Reset	0				0	0	0	0

Bit	Description
0	**OBF (Output Buffer Full).** The bit is set when the keyboard/mouse channel's DBBOUT is written by the core (i.e., writing to HIKDO or HIMDO register). The bit is cleared by a host processor read from the keyboard/mouse channel output buffer (60_{16}). This read-only bit is ignored when writing to this register.
1	**IBF (Input Buffer Full).** The bit is set when the keyboard/mouse channel's DBBIN is written by the host processor, i.e., writing to either address 60_{16} (data) or address 64_{16} (control). The bit is cleared by a core read of the input buffer (HIKMDI). This read-only bit is ignored when writing to this register.
2	**F0 (Flag 0).** A general-purpose flag that can be set or cleared by the core firmware.
3	**A2 (A2 Address).** Holds the value of the A2 signal in the last write operation of the host to the keyboard/mouse channel's input buffer (i.e., indicates A2 value during write to address 60_{16} or 64_{16}). This read-only bit is ignored when writing to this register.
7-4	**ST3-ST0 (Status Bits).** Four general-purpose flags that can be set or cleared by the core firmware.

Host Interface Keyboard Data Out Buffer Register (HIKDO)

The HIKDO register allows the core firmware to write to DBBOUT register while setting OBF bit in the Status register. If IRQ1 interrupt is enabled, it is sent. If the core interrupt on output buffer empty is enabled (OBECIE in HICTRL register is 1), writing to HIKDO de-asserts it (low).

Location: 00 $FEA6_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**Keyboard DBBOUT Data**							

Bit	Description
7-0	**Keyboard DBBOUT Data.**

Host Interface Mouse Data Out Buffer Register (HIMDO)

The HIMDO register allows the core firmware to write to the DBBOUT register while setting OBF bit in the Status register. If an IRQ12 interrupt is enabled, it is sent. If the core interrupt on output buffer empty is enabled (OBECIE in HICTRL register is 1), writing to HIMDO de-asserts it (low).

Location: 00 $FEA8_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**Mouse DBBOUT Data**							

Bit	Description
7-0	**Mouse DBBOUT Data.**

Host Interface Keyboard/Mouse Data In Buffer Register (HIKMDI)

The HIKMDI register allows the core firmware to read to the DBBIN register while clearing IBF bit in the Status register. If the core interrupt on IBF is enabled (IBFCIE in HICTRL register is 1). Reading from HIKMDI de-asserts it (low).

Location: 00 $FEAA_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Keyboard/Mouse DBBIN Data**							

Bit	Description
7-0	**Keyboard/Mouse DBBIN Data.**

5.2 POWER MANAGEMENT (PM) CHANNELS

The PC87591x implements two PM communication channels, both of which are compliant with ACPI specifications for an embedded controller interface. The PC87591x may be configured to work in either Private Interface or Shared Interface mode. The PM interface has two identical channels, which are individually configured and used. If using only one channel, Shared Interface mode should be used; if using both channels, Private Interface mode may be used.

The description below refers to a single channel. The signals may have the suffix ‘n’ to indicate the channel number, either 1 or 2, where applicable. Registers in the core domain have the prefix HIPMn to indicate the channel number. Host domain registers are identified by the logical device to which they belong.

Note: When working in PC87570 Compatible mode, only channel 1 may be used.

5.2.1 Features

- Two operation modes
  - PC87570 Compatible
  - Enhanced PM
- ACPI embedded controller interface compliant support
  - Shared interface
  - Private interface
- PM channel registers (legacy 62_{16}, 66_{16})
  - Command/Status
  - Data
- PM interrupt using
  - IRQ
  - SMI
  - SCI

5.2.2 General Description

The PM channel has two modes of operation:

- PC87570 Compatible (available for channel 1 only) supports software previously written for the PC87570.
- Enhanced PM includes a mechanism that facilitates easier generation of SCI and SMI interrupts to the host.

Figure 95 is a schematic diagram of the PM channel.

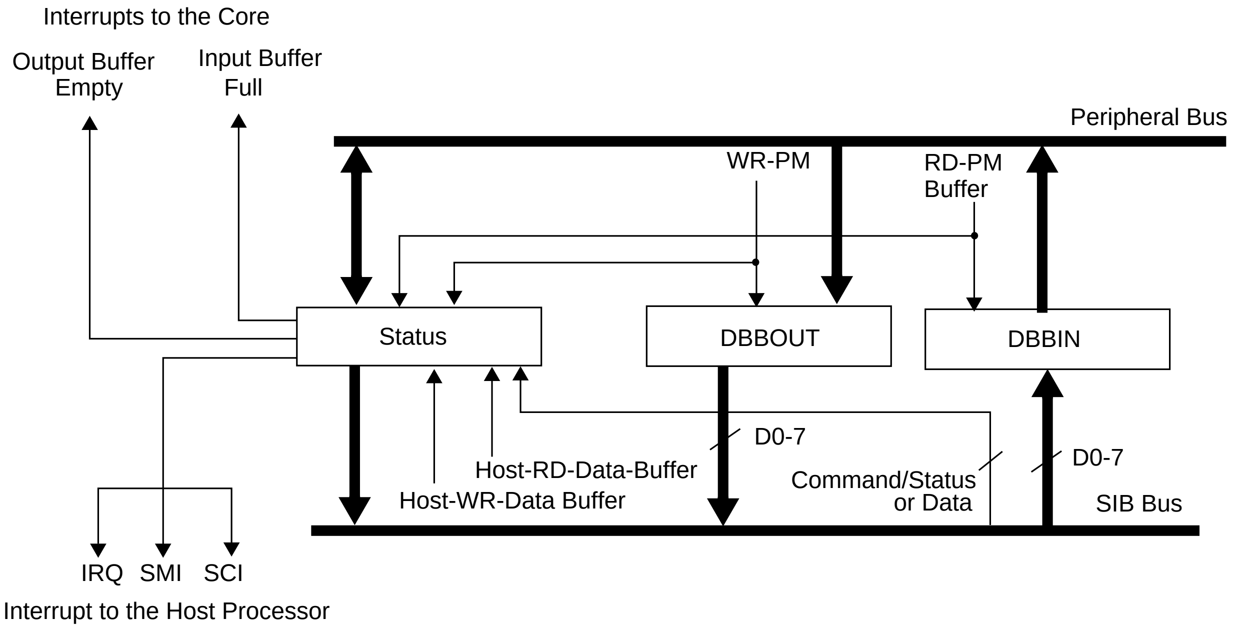


Figure 95. Host Interface PM Channel n Block Diagram

Data Registers

The PM channel has three registers.

- DBBOUT - can be written to by the core and read by the host processor. Multiple addresses in the core address space enable generating an IRQ, SMI or SCI interrupt on Output Buffer Full (OBF).
- DBBIN - can be written to by the host processor and read by the core.
- STATUS - can be read by both the core and the host processor. It has five bits (bits 2 and 4-7) that are written to by the core directly or, in Enhanced PM mode, via the control and configuration register. Three other bits are controlled by hardware to indicate the status of DBBIN and DBBOUT registers.

Host Addresses

The host processor accesses the PC87591x PM channel interface registers at two addresses in the host address space. These addresses are defined by two internal chip select signals specified in the PC87591x Host configuration registers; see Section 6.1.13 on page 355 and Section 6.1.14 for channels 1 and 2, respectively. Legacy setting of these addresses is 62_{16} and 66_{16} for channel 1 Status/Command and Data registers, respectively.

Table 43 shows the register mapping to the host processor I/O space. For simplicity, the Host Interface module specification refers to the legacy addresses.

5.0 Host Controller Interface Modules (Continued)

Table 43. Host Interface Registers to Host Processor Mapping

Port	Legacy Address[1]	Configuration Register Index	PM Internal Chip Select	Type	Register Name	Mnemonic
PM Channel n	62_{16}	Index 60_{16}, 61_{16}	Data	Write	Data	DBBIN
	66_{16}	Index 62_{16}, 63_{16}	Command/Status	Write	Command	DBBIN
	62_{16}	Index 60_{16}, 61_{16}	Data	Read	Data	DBBOUT
	66_{16}	Index 62_{16}, 63_{16}	Command/Status	Read	Status	STATUS

1. The legacy address serves as an example only. Do not assign the same address for both channels.

Core Interrupts

For each channel, the Host Interface module generates two level (high) interrupts to the core ICU (see Figure 96). The firmware can use these for interrupt-driven control of the PM channels.

In PC87570 Compatible mode (EME in HIPMnCTL register is set to 0), interrupts are enabled using HICTRL register bits PMOCIE and PMICIE for output buffer empty and input buffer full interrupts, respectively.

In Enhanced PM mode (EME in HIPMnCTL register is set to1), interrupts are enabled using HIPMnCTL register bits OBEIE and IBFIE for output buffer empty and input buffer full interrupts, respectively.

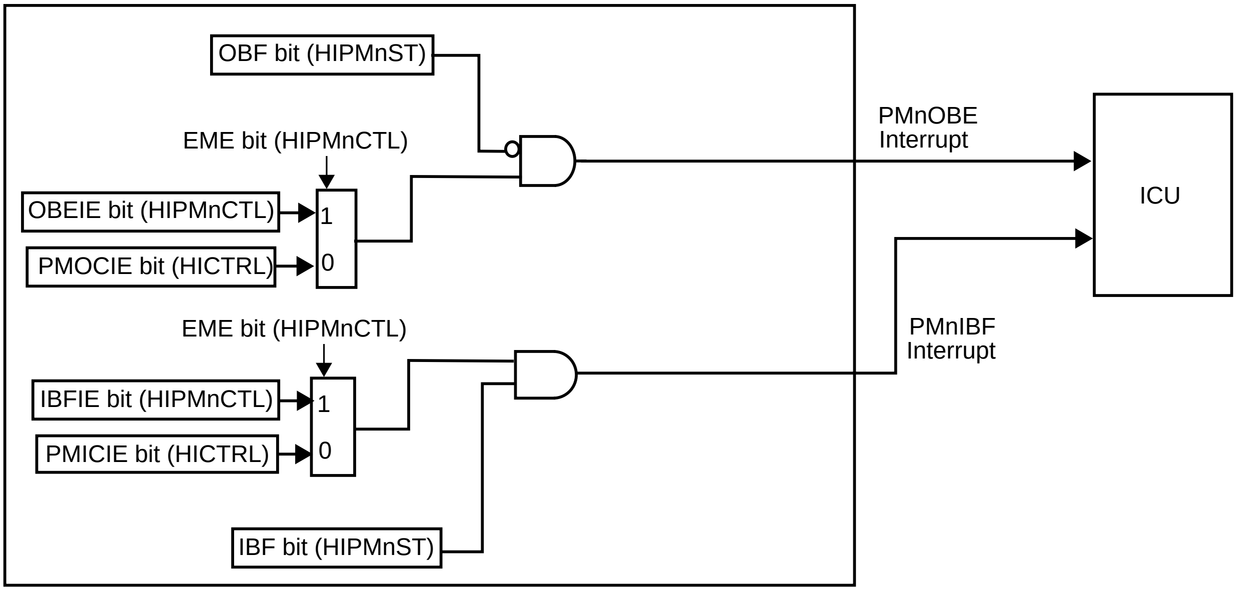


Figure 96. Core Interrupt Request for PM Channel n

Host Interrupt Generation Modes

The Host Interface module generates three types of interrupts to the host: regular IRQ, SMI and SCI. The interrupt schemes are designed to meet ACPI requirements for host interrupts. Two interrupt modes are supported: PC87570 Compatible and Enhanced PM.

PC87570 Compatible Mode

PC87570 Compatible mode uses the same method for IRQ generation as the PC87570. It is available only for PM channel 1 and is enabled when EME in HIPMnCTL register is set to 0. Figure 98 illustrates this scheme.

The host configuration module assigns host interrupts to IRQ numbers (see Section 6.1.13 on page 355). IRQ11 is used as an example interrupt and for the signal naming (the actual interrupt number used is determined by the IRQ routing logic). When hardware-driven IRQ11 is disabled (PMHIE in HICTRL register is cleared), the firmware can control the IRQ11 signal by writing to the signal's respective bit in HIIRQC register. When hardware-driven IRQ11 is enabled (PMHIE is set to 1), interrupts to the host are generated according to the status of the OBF flag.

5.0 Host Controller Interface Modules (Continued)

In Normal Polarity mode (IRQNPOL in HIIRQC register is cleared), the PC87591x supports two types of interrupts: legacy edge or legacy level. When an edge interrupt is selected (IRQM field in HIIRQC register is set to a value other than 0), the interrupt signal default value is high (1). When an interrupt signal must be sent (i.e., OBF flag is set), a negative pulse is generated. The pulse width is determined by the same field, IRQM, that selects the edge interrupt.

When a level interrupt is selected (IRQM in HIIRQC register is cleared), the interrupt signal is usually low (0). It is asserted (1) to indicate that the respective OBF flag has been set. The signal is de-asserted (0) when the output buffer is read (i.e., OBF flag is cleared).

In Negative Polarity mode (IRQNPOL in HIIRQC register is set to 1), IRQ signal behavior is inverted from the behavior described for normal polarity.

The PC87591x firmware can read the value of the IRQ11 signal by performing a read operation of IRQ11B bit in HIIRQC register.

The core can also control the routing of interrupts generated by the PM channel to one of the following:

- IRQ signal, when IRQE bit in HIPMnIE register is set
- SMI output, when SMIE bit in HIPMnIE register is set
- SCI event, using the $\overline{\text{SCIEC}}$ output, when SCIE bit in HIPMnIE register is set.

The core firmware should not enable more than one of these interrupts simultaneously. It should also update ST0 and ST1 bits to indicate the type of host interrupt used.

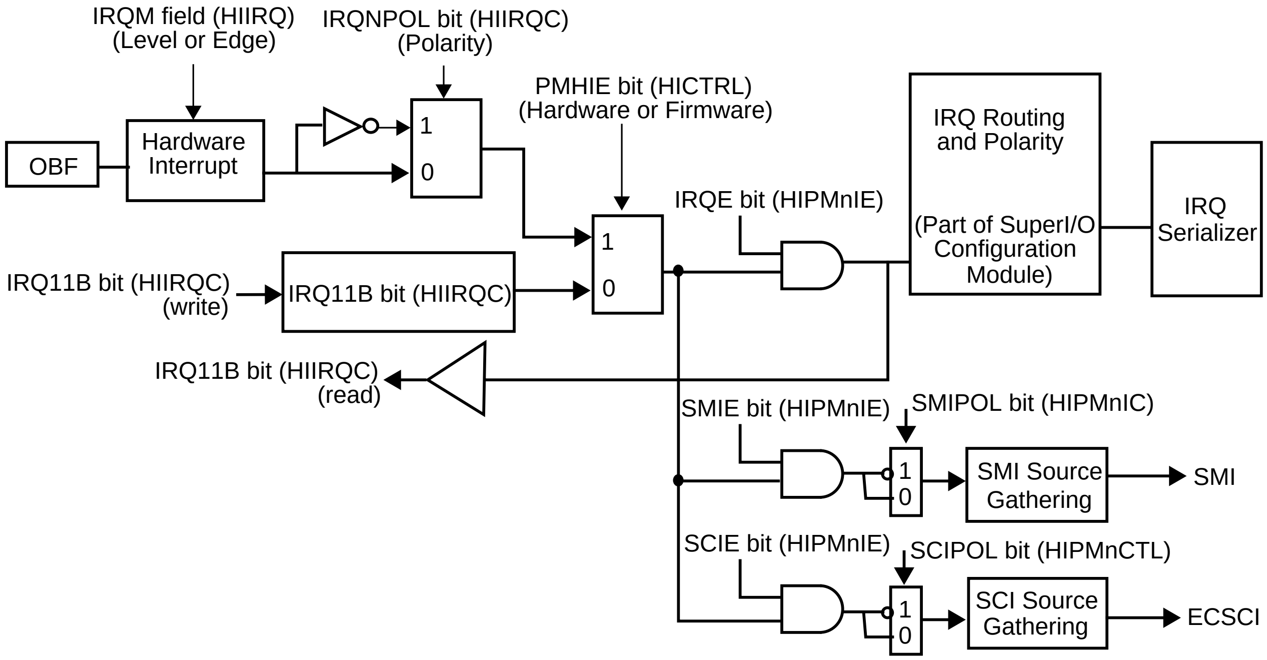


Figure 97. IRQ, SCI and SMI Control in PC87570 Compatible Mode (PM Channel 1 Only)

Enhanced PM Mode

Enhanced PM mode is available for both PM channels. It is enabled when EME in HIPMnCTL register is set to 1. Figure 98 illustrates interrupt generation in this mode.

Either IRQ, SMI or SCI interrupts may be generated under software control or by using hardware. Using hardware reduces software overhead and simplifies procedures.

The mechanism that generates the IRQ is identical to that used in PC87570 Compatible mode. To enable identical control of both channels, the bits that are used for channel 1 are separated from the keyboard/mouse channel's registers; see Figure 98 for bit usage.

IRQE in HIPMnIE register determines if an IRQ is sent from PM Channel n.

Enhanced PM mode supports direct generation of SCI and SMI on core writes to the Data output buffer and generation of SCI on core reading of the Data Input buffer. The core decides whether to generate an interrupt and which type of interrupt to generate by selecting the data register address in use.

When data is written to HIPMnDO register, the OBF flag in HIPMnST register is set and neither SMI nor SCI is generated.

5.0 Host Controller Interface Modules (Continued)

When data is written to HIPMnDOM register, the OBF flag and the internal OBF_SMI flag are set and an SMI interrupt is generated. The OBF_SMI flag is cleared when OBF flag is cleared. The SMI is generated as a pulse whose width is defined by PLMM in HPIMnIC register.

The SMI interrupt is routed to the SMI pin only if both HSMIE and SMIE bits in HIPMnIE register are set. When SMIE is set and HSMIE is cleared, SMIB in HIPMnIC register is used as the PMnSMI signal value. When SMIE is cleared, the PMnSMI signal is inactive (high).

When data is written to HIPMnDOC register, the OBF flag and OBF_SCI internal flag are set and an SCI interrupt is generated. OBF_SCI is cleared when OBF is cleared. The SCI is generated as a pulse whose width is defined by PLMS in HPIMnCTL register.

When data is read from HIPMnDIC register, the IBF flag is cleared, the IBF_SCI Internal flag is set and an SCI interrupt is generated. The IBF_SCI flag is cleared when the IBF is set again. The SCI is generated as a pulse whose width is defined by PLMS in HPIMnCTL register. Reading from HIPnDI register clears the IBF flag but does not generate an SCI interrupt.

Note that IBF_SCI flag may also be set by writing a 1 to SCIIS bit in HIPMnIC register. This is done to start an SCI interrupt on input buffer empty without a read operation from the input buffer.

The SCI interrupt is routed to the SCI pin only if HSCIE and SCIE in HIPMnIE register are set. When SCIE is set and HSCIE is cleared, the value of SCIB in HIPMnIC register is used as the PMnSCI signal value. When SCIE is cleared, PMnSCI is inactive (high).

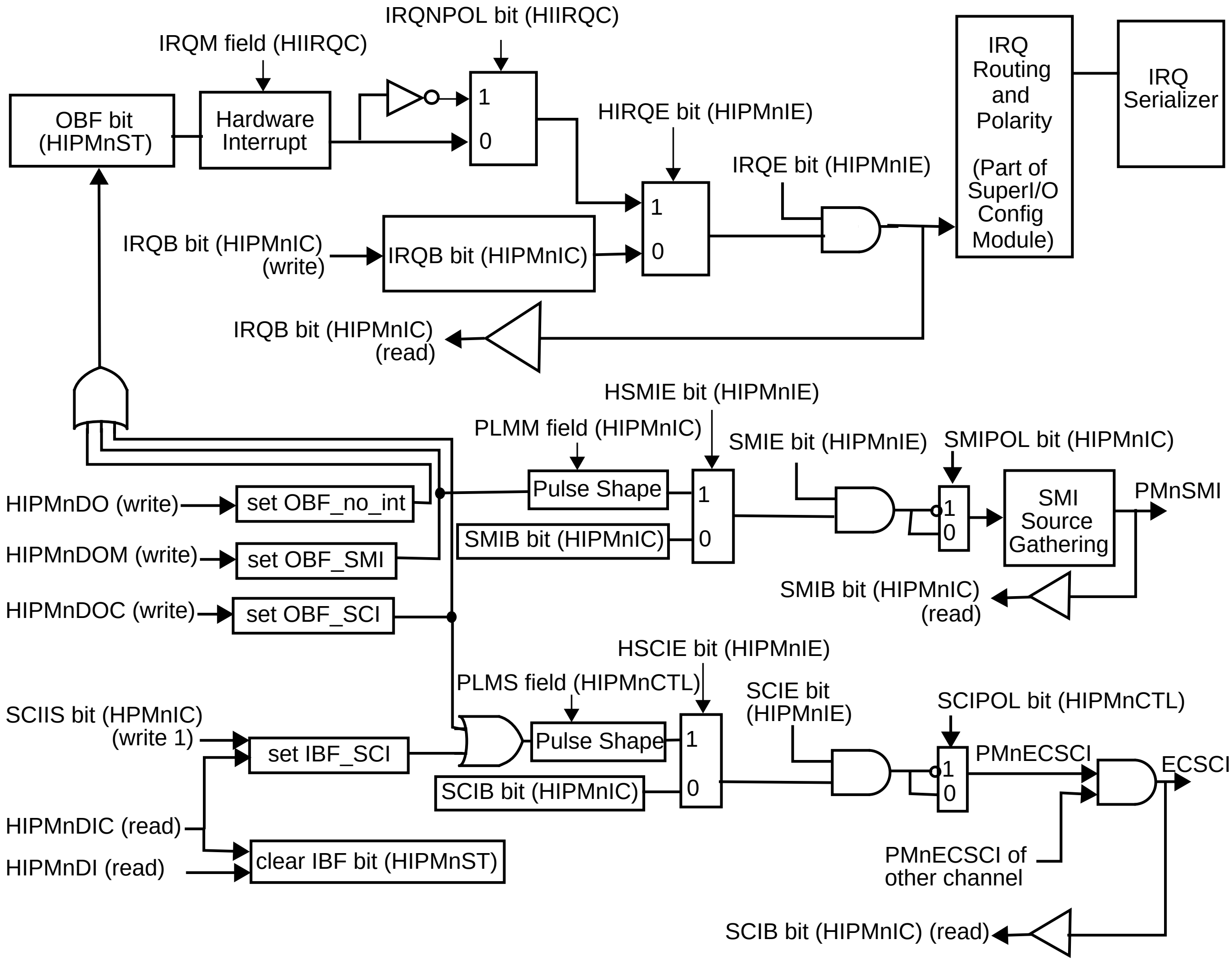


Figure 98. IRQ, SCI and SMI Control, Enhanced PM Mode

Status Read

The status of the PM channel data buffers can be read by both the host and the core. Bits 2 and 4-7 can be written by the core.

The host processor should read the Status register I/O address (legacy 66_{16}, for channel 1) to obtain the contents of the Status register. The core software should read/write the HIPMnST register to access the same information. The format of the Status register is identical for both the host and the core; see “Host Interface PM n Status Register (HIPMnST)” on page 291.

Host Data Write to Host PM Channel

The data buffer has two latches: one serves as an input buffer and the other serves as an output buffer. When writing to the Command (legacy address 66_{16}) or Data (legacy address 62_{16}) registers, the following sequence occurs: data is written to the Data In latch (DBBIN), IBF bit in the Status register is set and bit 3 (A2) in the Status register indicates to the core which of the two addresses was written to. When writing to the data register address, A2 bit of the Status register is cleared (0). When writing to the Command register address, A2 bit in the Status register is set (1).

The core identifies that data is present in the input buffer by either polling IBF bit in the Status register or acknowledging an interrupt when the input buffer interrupt is enabled (IBFCIE bit in HICTRL register is set to 1).

When the input buffer is full, the Status register should be read to identify which addresses were written to by checking A2 bit in HIPMnST register. The core can then read the data from the input buffer (HIPMnDI or HIPMnDIC registers). The IBF status bit is cleared when the data input buffer is read by the core.

Host Data Read from Host Interface Power Management Channel

The core writes to the Output Data latch (DBBOUT) when it needs to send data to the host. The OBF flag in the Status register (HIPMnST) is set to indicate that data is available in DBBOUT. DBBOUT should be written to only when OBF in HIPMnST register is cleared.

The PC87591x supports polling and interrupt communication schemes with the host. IRQ, SMI or SCI interrupts may be used. The core firmware writes data addressed to the PM drivers to the HIPMnDO register. When working in Enhanced PM mode, writes to HIPMnDOC and HIPMnDOM may be used to automatically generate SCI and SMI, respectively. Refer to “Host Interrupt Generation Modes” on page 287 for details of the interrupt generation scheme.

The host processor identifies the presence of data in the output buffer by either polling the Status register or by responding to IRQ, SMI or SCI events. When such data is available, the host can read it using a read operation from the address of the data register (legacy 62_{16} for channel 1). Reading from the data register clears the OBF flag (HIPMnST). In addition, when the host interrupt is in level mode (IRQM in HIIRQC register is set to 000_2) and the hardware interrupt is enabled, the IRQ signal is de-asserted (low).

The core can read OBF in HIPMnST register to identify when the output buffer is empty and ready for a new data transfer. When the output buffer empty interrupt to the core is enabled (PMOCIE bit in HICTRL register is 1 when EME bit in HIPMnCTL register is 0), the interrupt signal to the ICU is set high if the output buffer is empty (OBF is set to 0).

5.2.3 Core PM Registers

For a summary of the abbreviations used for Register Type, see Section 2 on page 34.

5.0 Host Controller Interface Modules (Continued)

Core PM Register Map

Mnemonic	Register Name	Type
HIPMnST[1]	Host Interface PM n Status	Varies per bit
HIPMnDO[1]	Host Interface PM n Data Out Buffer	WO
HIPMnDOC[1]	Host Interface PM n Data Out Buffer with SCI	WO
HIPMnDOM[1]	Host Interface PM n Data Out Buffer with SMI	WO
HIPMnDI[1]	Host Interface PM n Data In Buffer	RO
HIPMnDIC[1]	Host Interface PM n Data In Buffer with SCI	RO
HIPMnCTL[1]	Host Interface PM n Control	R/W
HIPMnIC[1]	Host Interface PM n Interrupt Control	R/W
HIPMnIE[1]	Host Interface PM n Interrupt Enable	R/W

[1] Where n stands for register 1 or 2.

Host Interface PM n Status Register (HIPMnST)

The HIPMnST register contains the status of the host interface PM channel buffers (DBBIN and DBBOUT). It also provides a means for the PC87591x to send status bits to the host. This register is read by a host processor read operation from address 66_{16}. HIPMnST is cleared on reset.

Location: Channel 1 - 00 $FEAC_{16}$
Channel 2 - 00 $FEBE_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**ST3-ST0**				**A2**	**F0**	**IBF**	**OBF**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	RO	**OBF (Output Buffer Full).** The bit is set when the PM channel's DBBOUT is written to by the core (writing to HIPMnDO, HIPMnDOM or HIPMnDOC register). The bit is cleared by a host processor read of the output buffer (62_{16}). Writing to this bit is ignored.
1	RO	**IBF (Input Buffer Full).** The bit is set when the PM channel's DBBIN is written to by the host processor (writing to either address 62_{16} or address 66_{16}). The bit is cleared by a core read of the PM input buffer (HIPMnDI or HIPMnDIC).
2	R/W	**F0 (Flag 0).** General-purpose flag that can be set or cleared by the core firmware.
3	RO	**A2 (A2 Address).** Indicates whether the last write operation of the host to the PMn channel was to the data register or the Command register. Writing to this bit is ignored. 0: Last write was to the data register (pointed to by configuration register index 60_{16} and 61_{16}) 1: Last write was to the command register (pointed to by configuration register index 62_{16} and 63_{16})
7-4	R/W	**ST3-ST0 (Status).** Four general-purpose flags that can be used for signaling between the host and core. When used as an embedded controller interface channel for ACPI, a predefined meaning is assigned to ST0, ST1 and ST2. The standard meaning is BURST, SCI event and SMI event, respectively.

5.0 Host Controller Interface Modules (Continued)

Host Interface PM n Data Out Buffer (HIPMnDO)

The HIPMnDO register allows the core firmware to write to the PM port DBBOUT register while setting the PM port OBF bit in the Status register. If enabled, an IRQ11 (and/or SCI and/or SMI, in PC87570 Compatible mode) interrupt is sent at that time. If the core interrupt on PM port output buffer empty is enabled, writing to HIPMnDO de-asserts it (low). If DMA on output buffer empty is enabled, the DMA request is de-asserted on the write to HIPMnDO.

Location: Channel 1 - 00 FEAE$_{16}$
Channel 2 - 00 FEC0$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**PM Channel DBBOUT Data**							

Bit	Description
7-0	**PM Channel DBBOUT Data.**

Host Interface PM n Data Out Buffer with SCI (HIPMnDOC)

The HIPMnDOC register has the same function as the HIPMnDO register. In addition, it generates an SCI interrupt when OBF is set and hardware SCI generation is enabled.

Location: Channel 1 - 00 FEB2$_{16}$
Channel 2 - 00 FEC4$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**PM Channel DBBOUT Data**							

Bit	Description
7-0	**PM Channel DBBOUT Data.**

Host Interface PM n Data Out Buffer with SMI (HIPMnDOM)

The HIPMnDOM register has the same function as the HIPMnDO register. In addition, it generates an SMI interrupt when OBF is set and hardware SMI generation is enabled.

Location: Channel 1 - 00 FEB4$_{16}$
Channel 2 - 00 FEC6$_{16}$

Type: WO

Bit	7	6	5	4	3	2	1	0
Name	**PM Channel DBBOUT Data**							

Bit	Description
7-0	**PM Channel DBBOUT Data.**

Host Interface PM n Data In Buffer (HIPMnDI)

The HIPMnDI register allows the core firmware to read the PM port DBBIN register while clearing the PM port IBF bit in the Status register. If the core interrupt on IBF for the PM channel is enabled, reading from HIPMnDI de-asserts it (low).

Location: Channel 1 - 00 FEB0$_{16}$
Channel 2 - 00 FED2$_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**PM Channel DBBIN Data**							

Bit	Description
7-0	**PM Channel DBBIN Data.**

Host Interface PM n Data In Buffer with SCI (HIPMnDIC)

The HIPMnDIC has the same function as the HIPMnDI register. In addition, it generates an SCI interrupt when IBF is cleared and when hardware SCI generation is enabled.

Location: Channel 1 - 00 FEB6$_{16}$
Channel 2 - 00 FEC8$_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**PM Channel DBBIN Data**							

Bit	Description
7-0	**PM Channel DBBIN Data.**

5.0 Host Controller Interface Modules (Continued)

Host Interface PM n Control Register (HIPMnCTL)

The HIPMnCTL register controls the operation mode and configuration of the PM channel. It includes the Enhanced mode enable bit and control bits for Enhanced mode operation. HIPMnCTL is 40_{16} on reset.

Location: Channel 1 - 00 $FEB8_{16}$
Channel 2 - 00 $FECA_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**EME**	**SCIPOL**	**PLMS**			**Reserved**	**OBEIE**	**IBFIE**
Reset	see text	1	0	0	0	0	0	0

Bit	Description
0	**IBFIE (Input Buffer Full Interrupt Enabler).** 0: IBF interrupt to the core is disabled 1: Enables an interrupt to the core when IBF in HIPMnST register is set
1	**OBEIE (Output Buffer Empty Interrupt Enable).** 0: OBF interrupt to the core is disabled 1: Enables an interrupt to the core when OBF in HIPMnST register is set
2	**Reserved.**
5-3	**PLMS (Pulse Level Mode SCI).** Sets the hardware-controlled SCI signal mode to be Level or Pulse and sets the pulse width. When PLMS = 000_2, the SCI signal functions in Level mode. In this mode, the SCI pulse shaper output value is low, and a high level is set to issue an interrupt (the respective OBF is set). When PLMS ≠ 0, the host interrupts are in Pulse mode. In this mode, the SCI pulse shaper output value is low, and it toggles high to issue an interrupt (i.e., when the respective output buffer register is written). The pulse widths are: **Bits** **5 4 3 Pulse Width** 0 0 0: Level interrupt 0 0 1: 1-Cycle Pulse 0 1 0: 2-Cycle Pulse 0 1 1: 4-Cycle Pulse 1 0 0: 8-Cycle Pulse 1 0 1: 16-Cycle Pulse Other: Reserved
6	**SCIPOL (SCI Negative Polarity).** 0: SCI output inactive value is low and its active (asserted) value is high 1: Inverted polarity is used. When SCIPOL is set, the SCI signal is the inverse of either what is stored in SCIB or the output of the SCI pulse shaper. This bit affects the SCI signal polarity in both PC87570 Legacy and Enhanced modes.
7	**EME (Enhanced Mode Enable).** 0: PM channel is used in Legacy mode. HIPMnST status bits are controlled by writes to the bit value, and interrupts are controlled by HICTRL and HIIRQC register bits (default for HIPM1CTL). 1: Enables enhanced control of the PM channel. The bits in HICTRL and HIIRQC registers are ignored in this case (default for HIPM2CTL). In HIPM2CTL (i.e., for PM channel 2), EME is a read-only bit that holds the value 1. Writes to this bit are ignored.

Host Interface PM n Interrupt Control Register (HIPMnIC)

The HIPMnIC register and its bits affect operation in Enhanced mode only (i.e., when EME bit in HIPMnCTL register is set). In PC87570 Legacy mode, the bits in this registers are ignored. HIPMnIC controls the PM n interrupt signals mode of operation. HIPMnCTL is 41_{16} on reset.

Location: Channel 1 - 00 FEBA_{16}
Channel 2 - 00 FECC_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**SCIIS**	**SMIPOL**	**PLMM**			**SCIB**	**SMIB**	**IRQB**
Reset	0	1	0	0	0	0	0	1

Bit	Description
0	**IRQB (Host Interrupt Request Control Bit).** When the IRQ signal is configured for direct control by the firmware (HIRQE in HIPMnIE register is 0), IRQB bit is output to the PMnIRQ signal. When read, IRQB bit returns the current value of the PMnIRQ signal. IRQn signal's value can be read regardless of the state of HIRQE in HIPMnIE register.
1	**SMIB (Host SMI Request Control Bit)**. When the SMI signal is configured for direct control by the firmware (HSMIE in HIPMnIE register is 0), SMIB bit is output to the PMnSMI signal (if SMIPOL=0, SMIB is output; if SMIPOL=1, SMIB is inverted before output). When read, SMIB bit returns the current value of the SMI pin. The SMI signal's value can be read regardless of the state of HSMIE in HIPMnIE register.
2	**SCIB (Host SCI Request Control Bit).** When the SCI signal is configured for direct control by the firmware (HSCIE in HIPMnIE register is 0), SMIB bit is output to the PMnSCI signal (if SCIPOL=0, SCIB is output; if SCIPOL=1, SCIB is inverted before output). When read, SCIB bit returns the current value of the SCI pin. The ECSCI signal value can be read regardless of the state of HSCIE bit in HIPMnIE register.
5-3	**PLMM (Pulse Level Mode SMI).** Sets the hardware-controlled SMI signal mode to Level or Pulse and sets the pulse width. When PLMM = 000_2, the SCI signal functions in Level mode. In this mode, the SMI pulse shaper output value is low, and a high level is set to issue an interrupt (i.e., the respective OBF is set). When PLMM ≠ 0, the host interrupts are in Pulse mode. In this mode, the SMI pulse shaper output value is low, and it toggles high to issue an interrupt (i.e., when the respective output buffer register is written). The pulse widths are: **Bits** **5 4 3 Pulse Width** 0 0 0: Level interrupt 0 0 1: 1-Cycle Pulse 0 1 0: 2-Cycle Pulse 0 1 1: 4-Cycle Pulse 1 0 0: 8-Cycle Pulse 1 0 1: 16-Cycle Pulse Other: Reserved
6	**SMIPOL (SMI Negative Polarity).** 0: SMI output inactive value is low and its active (asserted) value is high 1: Inverted polarity is used. When SMIPOL is set, the SMI signal is either the inverse of what is stored in SMIB or the output of the SMI pulse shaper. This bit affects the SMI signal polarity in both PC87570 Legacy and Enhanced modes
7	**SCIIS (SIC on IBF Start).** A write of 1 to this bit starts an SCI interrupt on IBF cleared. A write of 0 to SCIIS is ignored. When read, this bit always return 0.

Host Interface PM n Interrupt Enable Register (HIPMnIE)

The HIPMnIE register controls the PM n interrupt signals that enable SMI, SCI and IRQ interrupts. HIPMnIE is cleared on reset.

Location: Channel 1 - 00 FEBC$_{16}$
Channel 2 - 00 FECE$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**HSMIE**	**HSCIE**	**HIRQE**	**SMIE**	**SCIE**	**IRQE**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**IRQE (IRQ Enable).** 0: PMnIRQ signal assumes its default value (low) and no interrupts are issued 1: Enables PM generation of IRQ events
1	**SCIE (SCI Enable).** 0: PMnSCI signal assumes its default value (high) and no interrupts are issued 1: Enables PM generation of SCI events
2	**SMIE (SMI Enable).** 0: PMnSMI signal assumes its default value (high) and no interrupts are issued 1: Enables the generation of SMI events by this module
3	**HIRQE (Hardware IRQ Enable).** Works only in Enhanced PM mode. 0: IRQB bit of HIPMnIC register controls the value of the IRQ 1: Enables the generation of IRQ events by hardware control based on the status of the OBF flag
4	**HSCIE (Hardware SCI Enable).** Works only in Enhanced PM mode. 0: SCIB bit in HIPMnIC register controls the value of the SCI 1: Enables the generation of SCI events by hardware control based on the status of the OBF and IBF flags
5	**HSMIE (Hardware SMI Enable).** Works only in Enhanced PM mode. 0: SMIB bit in HIPMnIC register controls the value of the SMI 1: Enables the generation of SMI events by hardware control based on the status of the OBF flag
7-6	**Reserved.**

5.3 SHARED MEMORY AND SECURITY

The PC87591x on-chip flash can be shared by the host and the core. It may also be used by the host for BIOS code storage or other purposes. The on-chip flash resides in the core domain. In IRE and OBD environments, it is accessible via the core bus. For host accesses, the flash is mapped to the host memory address space via the host interface, and a bridge is provided between the host bus and the core bus. The bridge functionality includes:

- Memory mapping between host domain address space and core domain address space
- Host bus to core bus transaction bridging
- Locking mechanism between host and core domains to maintain coherence of on-chip flash contents during updates
- Read/write protection on host accesses to the on-chip flash
- Host-accessible control and status registers of on-chip flash
- Signaling interface for host-core communication associated with memory updates

In addition, the PC87591x supports expansion memory controlled by the core domain’s Bus Interface Unit (BIU). The bridge also supports host accesses to the expansion memory (including read/write protection on these accesses). These accesses can occur in IRE, OBD and DEV environments.

In the PC87591S, the Shared Memory and Security module also includes a Random Number Generator (RNG). The output of the RNG is available to the core.

5.3.1 Host Bus to Core Bus Access Translation

A core bus transaction is generated for each of the following types of host bus transactions:

- 8-bit memory read/write
- 8-bit FWH memory read/write
- 8-bit indirect read/write transactions, using I/O read/write to access the shared flash (see Section 5.3.3 on page 300)

Memory and FWH memory read/write transactions drive Long Wait on the Sync field until the transaction is completed on the core bus. Section 5.3.3 on page 300 describes the Sync field for indirect memory read/write transactions. Section 5.3.5 on page 301 describes the behavior for restricted accesses.

The host bus transaction is forwarded to the core bus after the following is done:

- Address is translated
- The translated address and the access type are verified to be both:
  - — In core domain’s base memory or expansion memory spaces
  - — Unprotected
- For writes (erase/program): on-chip flash is unlocked (HLOCK bit in SMCCST register is set)

Note that host bus read transactions are translated to read transactions on the core bus, and host bus write transactions are translated to write transactions on the core bus. Translated reads and writes behave the same as reads and writes by the core. When the erase bit is set and the on-chip flash is enabled and addressed, the effect of a write is the same as a write after the page erase bit is set.

5.3.2 Memory Mapping and Host Address Translation

Section 6.1.11 on page 349 describes in detail the host domain addresses for which the core bus generates transactions. In general, the BIOS memory on the host bus can occupy one of three regions in the memory space (see Table 60 on page 350).

Address translation between the host and the core domains is performed for host memory and FWH memory transactions. The 32-bit address received from the host bus is used to decode the different zones, as described in Section 6.1.11 on page 349. The address is then translated to the core bus address using the following rules:

- Legacy and Extended Legacy BIOS Range
  Handle only when enabled (see Section 6.1.11 on page 349 for the enabling alternatives); otherwise, transactions to this zone are ignored. The address is converted to a shared memory internal address as follows:

  SM_Host_Address[31-0] = {1111 1111 1111 111, Host_Memory_Address[16-0]}
- User Defined Shared Memory Space
  This address range is handled only when enabled (see Section 6.1.11 on page 349 for the enabling alternatives); otherwise, transactions to this zone are ignored. The address translation depends on the window size defined. When the window size is 2^n bytes, the lower ‘n’ bits are taken from the memory address, and the upper 32 – n bits of the LPC address are replaced with 1. The address is converted to an internal address as follows:

  SM_Host_Address[31-0] = {1111 ... ... 1, Host_Memory_Address[n-0]}

Host Controller Interface Modules (Continued)

- 386 Mode-Compatible BIOS Range and User Defined Shared Memory Range
  This address range is handled only when enabled (see Section 6.1.11 on page 349 for the enabling alternatives); otherwise, transactions to this zone are ignored. The address is converted to an internal address as follows:
  SM_Host_Address[31-0] = Host_Memory_Address[31-0]
- Indirect Memory Address
  This address specified in IMA3-0 is used as follows:
  SM_Host_Address[31-0] = {IMA3[7-0], IMA2[7-0], IMA1[7-0], IMA0[7-0]}
- For allowed addresses, the SM_Host_Address is translated to a core address using the following equation (a 21-bit address in the core address space is generated by adding the host memory address to the MBTA):
  CR_Space_Address[20-0] = (21 least significant bits)(SM_Host_Address + MBTA)
  MBTA is the size of the PC87591x on-chip flash memory, as defined in the Shared Memory Main Block Top Address register (SMCTA). This value is defined on reset to indicate the available memory size. In DEV environment, this value may be changed to allow code development for other flash sizes.
- The CR_Space_Address[21-0] is checked against the Host-Controlled Access Protection registers, the Core-Controlled Override Protection registers and general address space access limitations (i.e., space not mapped to the base memory or expansion memory).

Figure 99 illustrates the address translation scheme for shared memory transactions.

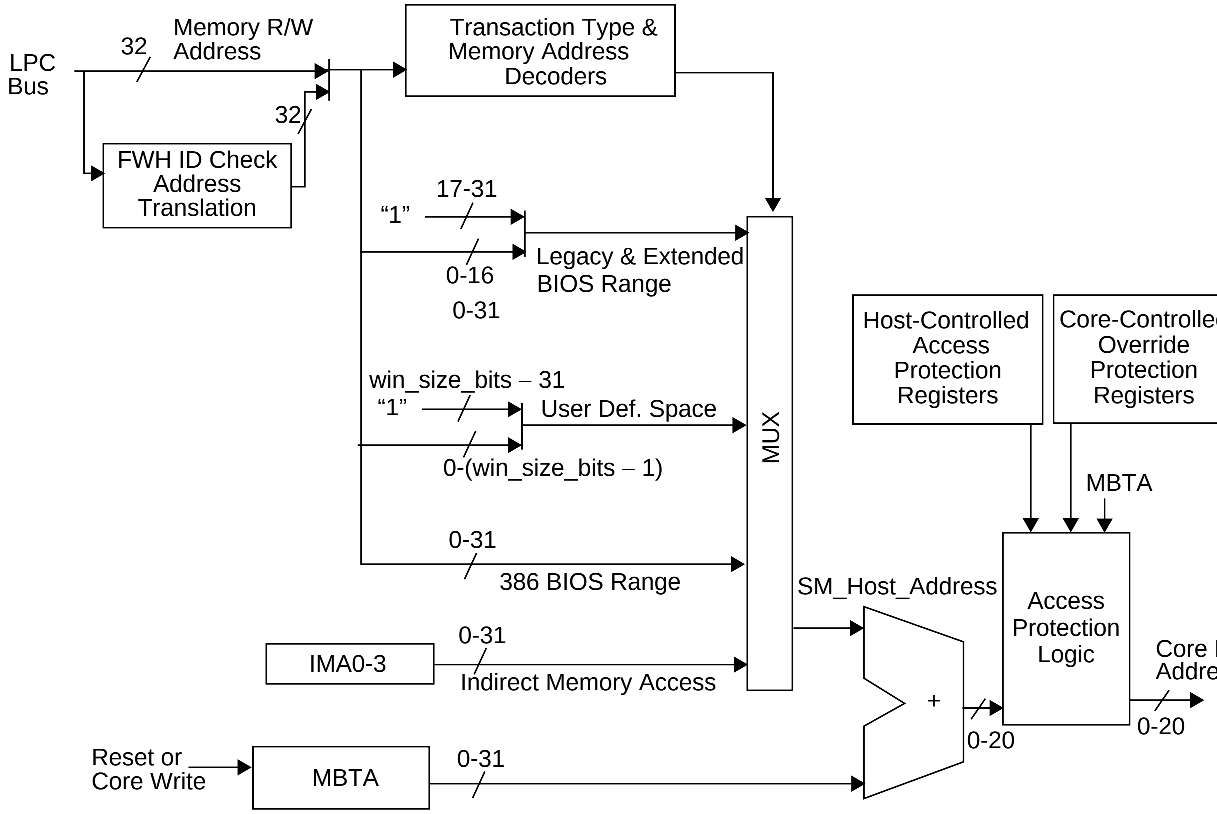


Figure 99. Address Translation Mechanism for 512 Kbyte On-Chip Flash

Access restrictions are based on the contents of the host-controlled and core-controlled access protection registers. The access protection logic may prevent read and/or write access to addresses in the core address space. The core-controlled register setting should always prevent host access to addresses that are not in the core domain's base memory or expansion memory spaces. Note that the resulting memory space is not continuous.

In DEV environment, the value of MBTA may be changed for ease of software development. The memory space between MBTA (1 0000_{16} in PC87591E or 2 0000_{16} in PC87591S) to F $FFFF_{16}$ is not accessible by the host.

The following figures illustrate the mapping function that results from the address translation function for a Shared BIOS scheme (Figure 100) and when the flash is mapped as a non-BIOS block of memory (Figure 101).

.

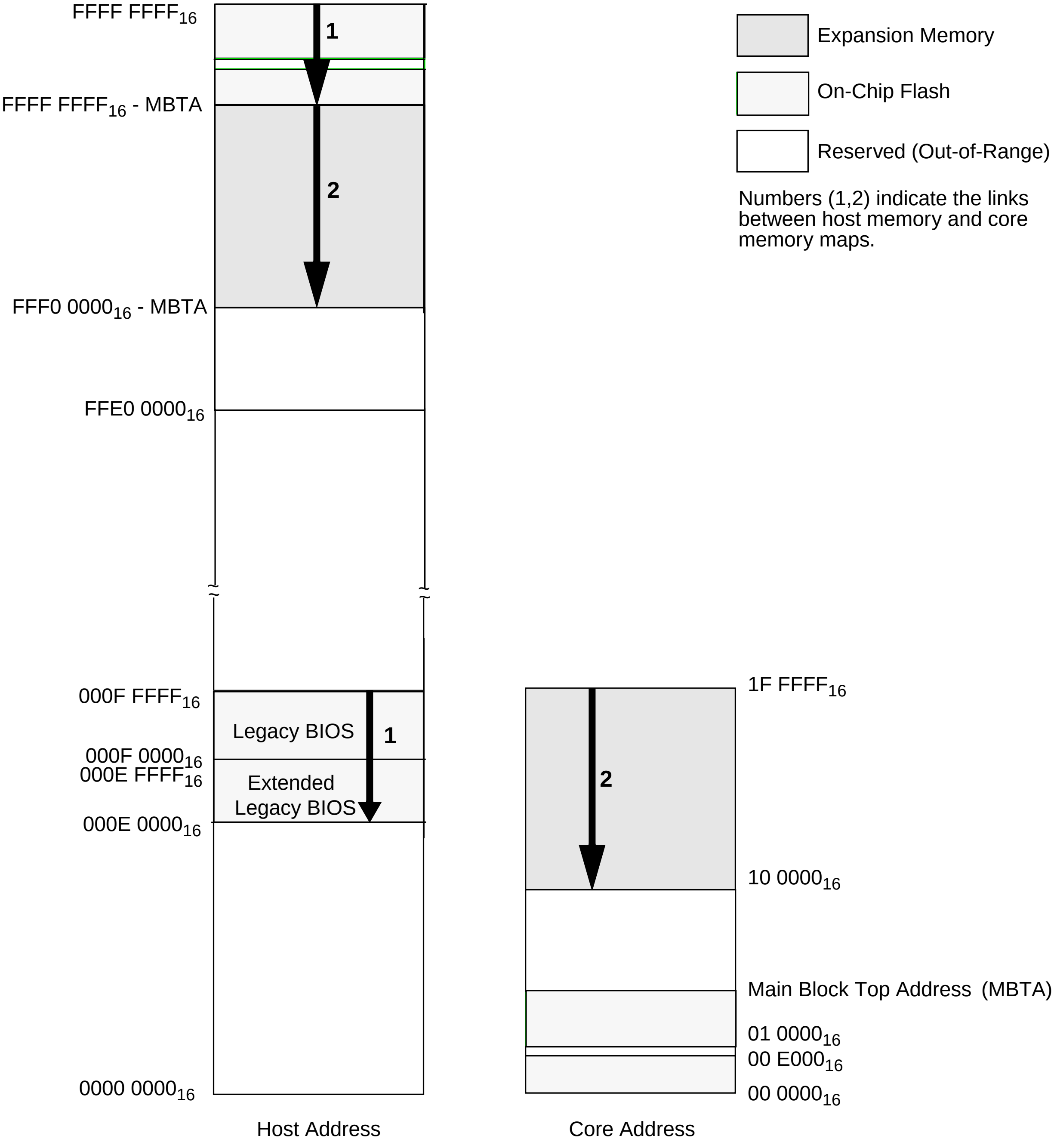


Figure 100. Host to Core Address Translation: 386 Mode-Compatible BIOS Range

Host Controller Interface Modules (Continued)

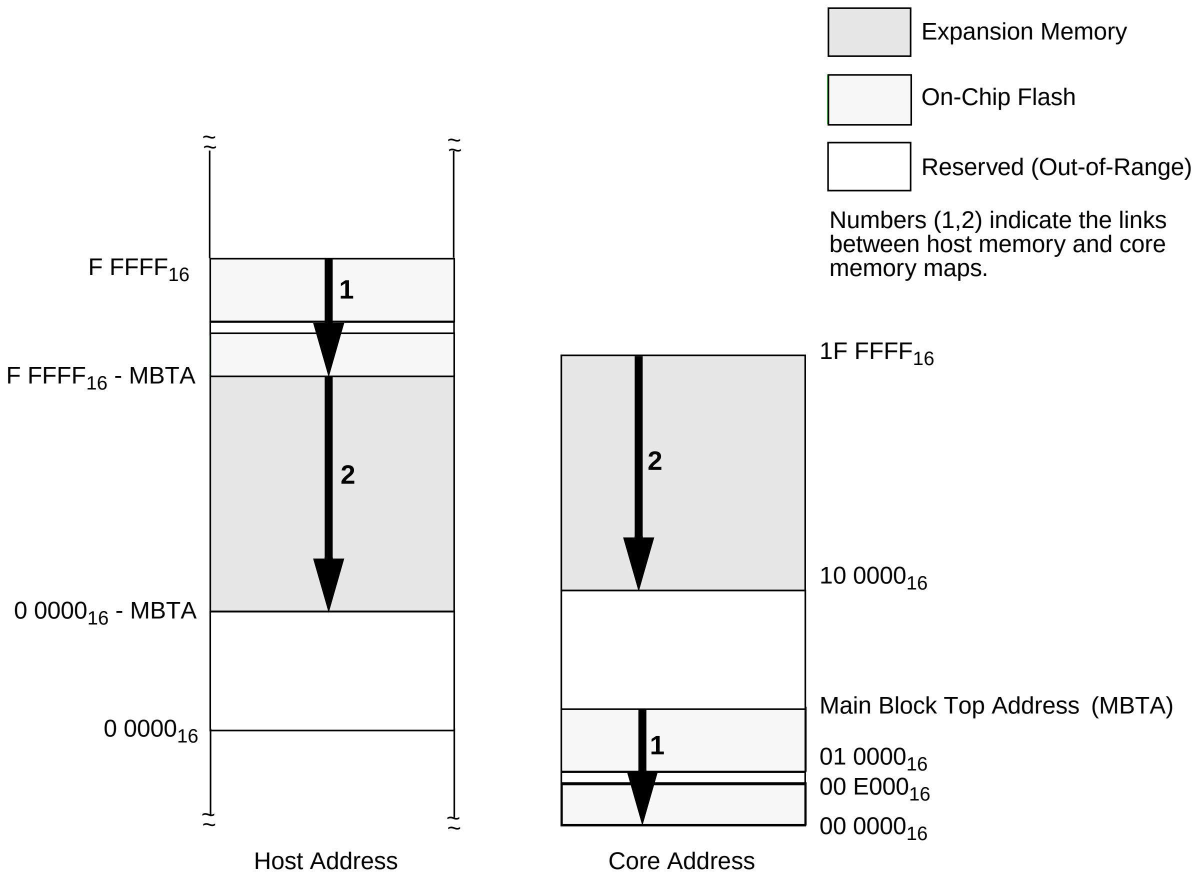


Figure 101. Host to Core Address Translation: Non-BIOS Mode

5.3.3 Indirect Memory Read and Write Transaction

The following I/O mapped registers can be utilized instead of memory mapping to perform a core bus transaction using an LPC I/O transaction:

- Four Indirect Memory Address registers (IMA3-IMA0), representing host address bits 31 to 0
- One Indirect Memory Data register (IMD), representing data bits 7 to 0

An LPC I/O write to the IMD register triggers a core bus memory write cycle using the addresses and data from IMA3-IMA0 and IMD registers, respectively. The LPC I/O write is completed when the core bus transaction is completed.

An LPC I/O read cycle from IMD register triggers a core bus memory read cycle using the addresses from IMA3-IMA0. The data returned from the core bus cycle is used to complete the LPC I/O read cycle from IMD register.

Read/write cycles from/to IMA3-IMA0 registers drive Short Wait on the Sync field. Read/write cycles from/to IMD register drive Long Wait on the Sync field until a transaction is actually performed and completed on the core bus.

Indirect memory read/write transactions are subject to the same memory mapping, locking mechanism and host access protection as memory and FWH memory read/write transactions. For more details, see Sections 5.3.2, 5.3.4 and 5.3.5.

5.3.4 Locking Between Domains

For read operations, hardware handles arbitration between the host and core. For program and erase operations, the PC87591x provides the means for enabling exclusive use for any particular access path over a sequence of operations.

By default, the core has access permission to the on-chip flash, and the HLOCK bit is cleared (0). In this case, if a program or erase operation starts, any host read operation is deferred or aborted, as defined by HERES field in SMCCST register.

When host erase or program operation is required, the Signaling interface (Section 5.3.7 on page 303) or one of the other host/core communication channels should be used to request control of the flash from the core. The core should go into a state in which the flash is not written or erased and then set the HLOCK bit. Note that core reads from the flash may be delayed in this case; therefore, it is recommended for critical code to be copied to RAM. Once the host access to the flash is completed, the host should indicate this to the core, allowing it to clear the HLOCK bit and resume normal operation.

Host Controller Interface Modules (Continued)

The core can clear HLOCK bit at any time, preventing any further erase or program operations by the host.

5.3.5 Host Access Protection

The host read/write protection is software controlled via a set of registers accessible to the host. The protection granularity is per block. Each of the 32 software-controlled protection blocks is 64 Kbytes; the block's read protection and write protection flags may be set independently. A Lock Protect flag may be set to prevent future changes to the read and write protection bits. Once locked, the lock bit and the read/write enable bits may be changed only after Host Domain Hardware reset.

The core can override the host settings and prevent host access to certain areas of the shared memory. The override may be set independently for read and write. In the first 128 Kbytes of address space, each core-controlled block is 8 Kbytes. For the rest of the memory space the blocks are 64 Kbytes each.

The default value of the protection registers is set according to the properties of the block. There are three types of blocks:

- Core Boot Block:Read, Erase and Program protected from the host.
- Host Boot Block:Open for Read by the host; Erase and Program protected from the host.
- Other Blocks:Open for Read, Erase and Program by the host.

Core on-chip peripherals and RAM are never accessible to the host (for both read and write). The range from MBTA to 0 $FFFF_{16}$ should be protected from host access using the core-controlled protection registers.

The size of any of the above blocks is defined as part of the flash protection information. This information is stored in the flash information block at address FE_{16}. This information is updated while the flash is programed or during a Special Erase, through either the JTAG or Parallel Interface.See Section 4.16.6 on page 226 for information about protection information fields.

Core Boot Block

This block is not accessible by the host and (for either read nor write). The core boot block starts at address 0 0000_{16} and ends as defined in the protection information field (8 Kbytes to 128 Kbytes in steps of 8 Kbytes). The core boot block access protection settings may not be changed, and the respective protection bits are read only.

Host Boot Block

By default, this block may be read by the host. Host writes to this block are always disabled. The host boot block size is 64 Kbytes and is available when the Host Boot Block bit in the protection word is cleared. The Host boot block is always the last 64 Kbytes of the base memory (i.e., MBTA – 64 Kbytes to MBTA – 1 Kbyte). In case of an overlap between the host boot block and the core boot block, access protection settings of the core boot block are used. The host boot block access protection settings may not be changed, and the respective protection bits are read only.

Other Blocks

By default, all other blocks are read and write protected. The core may enable host read and/or write access to these blocks.

Setting the Host Access Protection Flags

There are two sets of host access protection flags, as shown in Figure 102:

- Host-controlled host access protection flags
- Core-controlled host access protection flags

Host-Controlled Host Access Protection Flags. For each of the 32 protection blocks there is a set of three bits (flags): Read Protect, Write Protect and Lock Protect. The 32 sets of flags are accessible via two registers, Shared Memory Host Access Protect Register 1 and 2 (SMHAP1-2), using an indexing scheme.

The Host Block index may be calculated using the following equation:

Host_Block_Index = CR_Space_Address[20-0] / 64K or
Host_Block_Index = (21 least significant bits)(SM_Host_Address + MBTA) / 64K

See Section 5.3.2 on page 297 for the definitions of the host address translation.

- To change a flag setting, write the new flag setting, together with the required index field (i.e., Host Access Protection Index) and a cleared Index Write bit, to the appropriate register (SMHAP1 or SMHAP2).
- To read the values of the flags:
  1. Read the value of the register and save the index field.
  2. Write the index of the register's flag (i.e., write the index with a 1 in the Index Write bit).
  3. Read the settings of the register's flag.
  4. Restore the index field by writing back the value of the index field stored in step 1.

Core-Controlled Access Protection Flags. For core-controlled host access protection there is a read protect and write protect bit for each block The core block number are parallel to the host blocks for blocks 2-31. The core-controlled host access protection has a finer granularity for the first two host blocks, which are split into 16 core blocks, indicated as LA0 - LA15.

Host Controller Interface Modules (Continued)

The block number may be calculated using the following equation:

Core_Block_Number = (CR_Space_Address < 128K) ?
CR_Space_Address[20-0] / 64K : CR_Space_Address[20-0] / 8K

The Shared Memory Core Override Read Protect 0-2 registers and Shared Memory Core Override Write Protect 0-2 registers provide core access to the read and write protect bits, respectively. Bits in these registers may be set and cleared by writing to the registers.

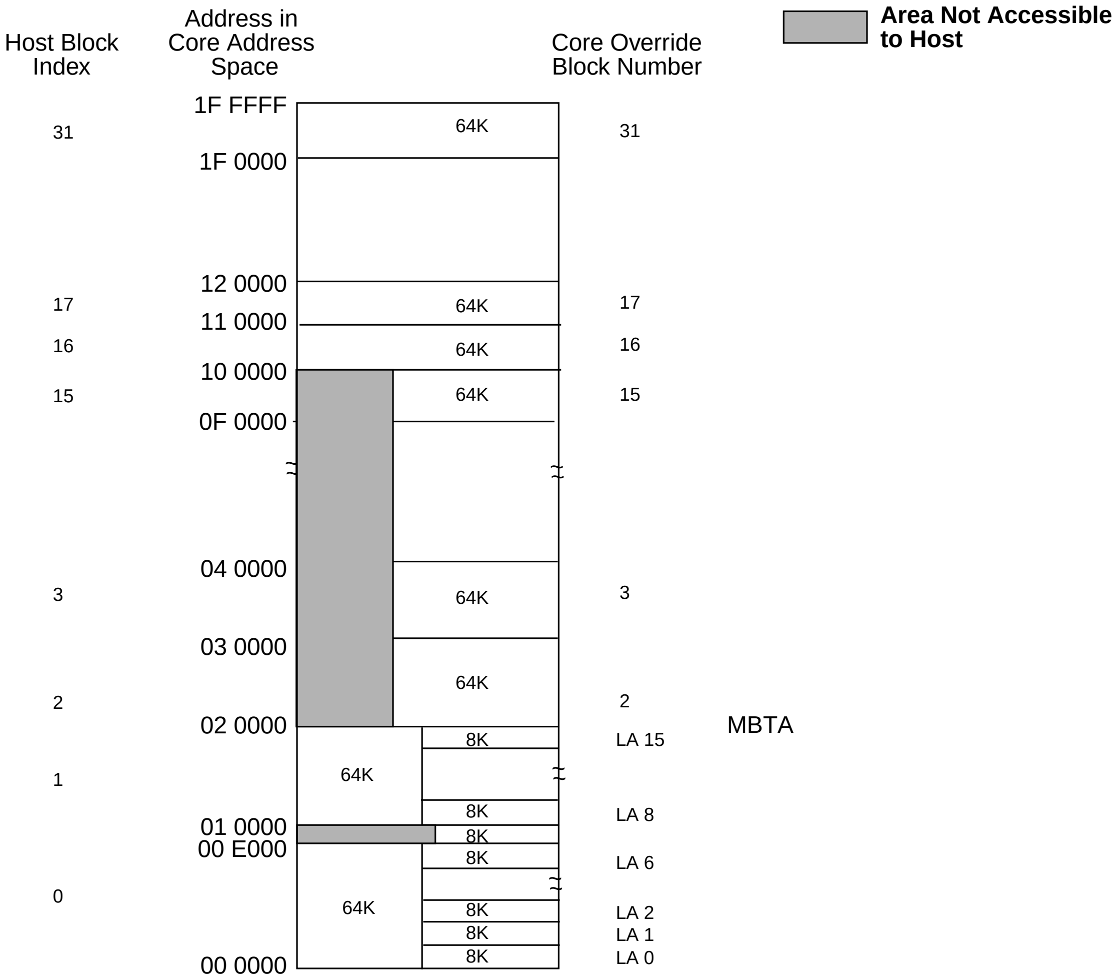


Figure 102. Protection Blocks for Protection Bits and Override Protection Bits

Response to a Restricted Access

A restricted access is an access that complies with at least one of the following conditions:

- The access type (i.e., Read or Write) of the translated host address (i.e., core address) is protected.
- A reset to the core domain due to a Warm reset event or a flash empty indication is set (Core Boot Block field of the protection word is 1111_2).

The PC87591x responds to a restricted access by generating an interrupt (if enabled by HERRIEN bit in SMCCST register) to the core. Two status bits (HWERR and HRERR) indicate whether the restricted access is a read or write access. The response on the host bus is according to the HERES field.

For restricted write accesses: Data written is ignored; when the HERES field is 10_2, the read or write transaction is completed with an error SYNC; otherwise, it is completed with a ready SYNC.

For restricted read accesses: When the HERES field is 00_2, the read or write transaction drives Long Wait (endlessly, unless the access becomes unrestricted or the HERES field is changed). When the field is 01_2, the PC87591x completes the transaction with a ready SYNC and data of 00_{16}; when the field is 10_2, it completes the transaction with an error SYNC and data of 00_{16}.

5.3.6 Random Number Generator (PC87591S)

The PC87591S includes a Random Number Generator unit. Once enabled, this unit can provide up to 800 random bits per second. The random bits may be read one byte at a time from RNGD register when DVALID bit in RNGCS register is set. Note that a double buffer scheme is used to improve the performance of bit gathering. The randomization of the data is verified by compliance with FIPS140-1 specifications.

5.3.7 Signaling Interface

For memory update operations, the host may need to exchange signals (indicating the state) with the core. The Signaling interface supports this scheme. The scheme is designed mainly for polling-based operations, but it allows the host to interrupt the core.

The signaling hardware includes an 8-bit register that may be read by both host and core. The host may modify bits 0 through 3 of this register; the core may modify bits 4 through 7. A host write to the register sets the Host Semaphore Write (HSEMW) bit in the Shared Memory Core Control and Status register (SMCCST); if the interrupt enable bit (HSEMIE in SMCCST register) is set, the shared memory interrupt to the core is set.

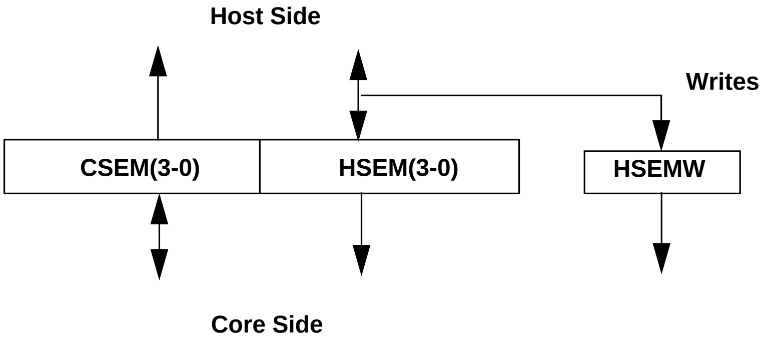


Figure 103. Signaling Interface

5.3.8 Shared Memory Host Registers

The following set of registers is accessible only by the host. The registers are maintained by V_{DD}.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Shared Memory Host Register Map

Offset	Mnemonic	Register Name	Type
00_{16}	SMIMA0	Shared Memory Indirect Memory Address 0	R/W
01_{16}	SMIMA1	Shared Memory Indirect Memory Address 1	R/W
02_{16}	SMIMA2	Shared Memory Indirect Memory Address 2	R/W
03_{16}	SMIMA3	Shared Memory Indirect Memory Address 3	R/W
04_{16}	SMIMD	Shared Memory Indirect Memory Data	R/W
05_{16}	SMHC	Shared Memory Host Control	R/W
06_{16}	SMHST	Shared Memory Host Status	Varies per bit
07_{16}	SMHAP1	Shared Memory Host Access Protect 1	Varies per bit
08_{16}	SMHAP2	Shared Memory Host Access Protect 2	Varies per bit
$0C_{16}$	SMHSEM	Shared Memory Host Semaphore	Varies per bit

Host Controller Interface Modules (Continued)

Shared Memory Indirect Memory Address Register 0 (SMIMA0)

This register defines the addresses 7-0 for a read or write transaction to the memory.

Location: Offset 00_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Indirect Memory Address 7-0**							

Bit	Description
7-0	**Indirect Memory Address 7-0.**

Shared Memory Indirect Memory Address Register 1 (SMIMA1)

This register defines addresses 15-8 for a read or write transaction to the memory.

Location: Offset 01_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Indirect Memory Address 15-8**							

Bit	Description
7-0	**Indirect Memory Address 15-8.**

Shared Memory Indirect Memory Address Register 2 (SMIMA2)

This register defines addresses 23-16 for a read or write transaction to the memory.

Location: Offset 02_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Indirect Memory Address 23-16**							

Bit	Description
7-0	**Indirect Memory Address 23-16.**

Shared Memory Indirect Memory Address Register 3 (SMIMA3)

This register defines addresses 31-24 for a read or write transaction to the memory.

Location: Offset 03_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Indirect Memory Address 31-24**							

Bit	Description
7-0	**Indirect Memory Address 31-24.**

Host Controller Interface Modules (Continued)

Shared Memory Indirect Memory Data Register (SMIMD)

This register defines data bits 7-0 for a read or write transaction to the memory.

Location: Offset 04_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Indirect Memory Data 7-0**							

Bit	Description
7-0	**Indirect Memory Data 7-0.**

Shared Memory Host Control Register (SMHC)

This register controls the embedded flash operation modes for host-initiated transactions. The register is cleared (00_{16}) on Host Domain Hardware reset.

Location: Offset 05_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**PMER**	**Reserved**				
Reset	0	0	0	0	0	0	0	0

Bit	Description
4-0	**Reserved.**
5	**PMER (Page Erase).** When the bit is set (1), a valid host write to the embedded flash erases the entire page pointed to by the write address. This bit can be altered only when HBUSY bit is cleared (0); otherwise results are undefined.
7-6	**Reserved.**

Shared Memory Host Status Register (SMHST)

This register provides the embedded flash status for host initiated transactions. The register is cleared (00_{16}) on Host Domain Hardware reset.

Location: Offset 06_{16}

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**HDERR**	**FMLFULL**	**HBUSY**	**HPERR**	**HEERR**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	R/W1C	**HEERR (Host Erase Error).** The bit is set (1) when an error occurs during a host-initiated erase. The bit is cleared by writing 1 to it. This bit can be cleared only when HBUSY bit is cleared (0).
1	R/W1C	**HPERR (Host Program Error).** The bit is set (1) when an error occurs during a host-initiated program. The bit is cleared by writing 1 to it. This bit can be cleared only when HBUSY bit is cleared (0).
2	RO	**HBUSY (Host Busy).** Read only. The bit is set (1) when the embedded flash is busy programing or erasing the flash due to a host-initiated operation.
3	RO	**FMLFULL** (**Double Buffer Full).** Read only. The bit is set (1) when the double buffer is full.

Host Controller Interface Modules (Continued)

Bit	Type	Description
4	R/W1C	**HDERR (Host Data Loss Error).** The bit is set (1) when a write to a full buffer is detected. The bit is cleared by writing 1 to it. This bit can be cleared only when HBUSY bit is cleared (0).
7-5		**Reserved.**

Shared Memory Host Access Protect Register 1 and 2 (SMHAP1-2)

This register holds the read/write protection and lock control from the host side to the shared memory. The memory is partitioned into 64 Kbyte blocks. SMHAP1 controls the first 16 blocks (addresses 0-1 Mbyte). SMHAP2 controls the second group of 16 blocks (addresses 1-2 Mbyte). The block mapping is in the core address space. See “Setting the Host Access Protection Flags” on page 301 for the calculation method of the block address in the host address space. On Host Domain Hardware reset, all write-protect flags are set and all lock-protect and read-protect flags are cleared.

Location: Offset 07_{16} and 08_{16}

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Host Access Protection Index**				**Index Write**	**Host Lock Protection**	**Host Write Protection**	**Host Read Protection**
Reset	0	0	0	0	0	0	1	0

Bit	Type	Description
0	R/W	**Host Read Protection.** The block number is as held in the index field (bits 7-4). Note that the Core Override protection may disable reads even when reads are allowed by this register. 0: Host Reads are allowed for this block 1: Host Reads are inhibited for this block
1	R/W	**Host Write Protection.** The block number is as held in the index field (bits 7-4). Note that the Core Override protection may disable writes even when writes are allowed by this register. 0: Program and erase are allowed for this block 1: Program and erase are inhibited for this block
2	R/W	**Host Lock Protection.** The block number is as held in the index field (bits 7-4). When set, the bit prevents changing the values of the Host Read Protect, Host Write Protect and Host Lock Protection bits for this block. Once set, this bit is cleared by Host Domain Hardware reset only. 0: Changes to protection bits (0-2) for this block are enabled 1: Protection bits (0-2) for this block are locked, and the bits’ values may not be changed
3	WO	**Index Write.** Indicates that this is an index write transaction; therefore, bits 0-2 of this register are ignored. When read, always returns 0. 0: Write transaction affects all fields of this register (writes to bits 0-2 use the newly written index) 1: Write transaction for purpose of index update; bits 0-2 should not be updated by this write.
7-4	R/W	**Host Access Protection Index.** Holds the index number of the host block accessed by the other fields in this register. All blocks are 64 Kbytes. The block index is calculated in the core address space. For details of the address conversion, see Section 5.3.2 on page 297. Index = Block_First_Address / 64K In SMHAP1: $0000\text{-}1111_{16}$ for indexes 0-15, respectively In SMHAP2: $0000\text{-}1111_{16}$ for indexes 16-31, respectively

Shared Memory Host Semaphore Register (SMHSEM)

This register provides eight semaphore bits between the core and host. Four of the bits may be set by the host and four may be set by the core. The register is cleared (00_{16}) on Host Domain Hardware reset.

Location: Offset $0C_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**CSEM3**	**CSEM2**	**CSEM1**	**CSEM0**	**HSEM3**	**HSEM2**	**HSEM1**	**HSEM0**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
3-0	R/W	**HSEM3-0.** Four bits that may be updated by the host and read by both the host and the core.
7-4	RO	**CSEM3-0.** Four bits that may be updated by the core and read by both the host and the core.

5.3.9 Shared Memory Core Registers

The following set of registers is accessible only by the core. These registers are maintained by V_{CC}.

For a summary of the abbreviations used for Register Type, see Section 2 on page 34.

Shared Memory Core Register Map

Mnemonic	Register Name	Type
SMCCST	Shared Memory Core Control and Status	R/W
SMCTA	Shared Memory Core Top Address	RO in IRE and OBD environments; R/W in DEV environment
SMHSEM	Shared Memory Host Semaphores	Varies per bit
SMCORP0-2	Shared Memory Core Override Read Protect 0-2	R/W or RO
SMCOWP0-2	Shared Memory Core Override Write Protect 0-2	R/W or RO
RNGCS	Random Number Generator Control and Status	Varies per bit
RNGD	Random Number Generator Data	RO

Shared Memory Core Control and Status Register (SMCCST)

This register provides control and status of read/write from/to a restricted address. The register is cleared (00_{16}) on reset.

Location: 00 $F900_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**HSEMIE**	**HSEMW**	**HLOCK**	**HERES**		**HERRIEN**	**HWERR**	**HRERR**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**HRERR (Host Read Error).** The bit is set (1) when the host attempts to read from a read-protected block or out-of-range address. An out-of-range address is an address that the LPC configuration module defines as mapped to the PC87591x, but it is actually translated to a reserved address in the core address space. Writing 1 to this bit clears it to 0. Writing 0 has no effect.
1	**HWERR (Host Write Error).** The bit is set (1) when the host attempts to write to a read-protected block or out-of-range address. An out-of-range address is an address that the LPC configuration module defines as mapped to the PC87591x, but it is actually translated to a reserved address in the core address space. Writing 1 to this bit clears it to 0. Writing 0 has no effect.
2	**HERRIEN (Host Error Interrupt Enable).** When set (1) and either the HRERR or HWERR bit is set (1), a core interrupt is generated; otherwise, the core interrupt is inactive.

Host Controller Interface Modules (Continued)

Bit	Description
4-3	**HERES (Host Error Response).** Controls response type on read/write from/to a protected block or out-of-range address. An out-of-range address is an address that the LPC configuration module defines as mapped to the PC87591x, but it is actually translated to a reserved address in the core address space. **Bits** **4 3 Description** 0 0: Drive Long Wait for read; ignore write 0 1: Read back 00_{16}; ignore write 1 0: Drive error SYNC for both read and write 1 1: Reserved
5	**HLOCK (Host Lock).** 0: The bridge does not generate write transactions on the core bus 1: The bridge can generate write transactions on the core bus This bit should be set only when FMBUSY bit is cleared (see ”Flash Memory Status Register (FLSR)” in Section 4.16.7 on page 228).
6	**HSEMW (Host Semaphore Write).** The bit is set (1) when the host writes to HSEM register. Writing 1 to this bit position clears it to 0. Writing 0 has no effect.
7	**HSEMIE (Host Semaphore Interrupt Enable).** When the bit is set (1), the interrupt to the core is set (level high) if HSEMW is set.

Shared Memory Core Top Address Register (SMCTA)

This register provides information about the size of the on-chip main block. The register is loaded with its default value on V_{CC} Power-Up reset only.

Location: 00 $F902_{16}$

Type: RO in IRE and OBD environments
R/W in DEV environment

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**MBSD**				
Reset	0	0	0	2 (see note in field description)				

Bit	Description
4-0	**MBSD (Main Block Size Definition).** Defines the size of the main block in 64 Kbyte units. Thus the MBTA value is MBSD * 1 0000_{16}. Note that MBTA is actually the first address beyond the main block. The reset value of this field is affected by the Force MBTA Zero bit in the protection word (see Section 4.16.6 on page 226). When the Force MBTA Zero bit is set, the reset value of this field is 0_{16}; when the bit is cleared, the reset value is as shown in the bit table, above. This field is loaded on V_{CC} Power-Up reset with the on-chip flash size. In DEV environment, the MBSD may be loaded with a new value.
7-5	**Reserved.**

Shared Memory Host Semaphore Register (SMHSEM)

This register provides eight semaphore bits between the core and the host. Four of the bits may be set by the host; four may be set by the core. The register is cleared (00_{16}) on reset.

Location: 00 $F904_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**CSEM3**	**CSEM2**	**CSEM1**	**CSEM0**	**HSEM3**	**HSEM2**	**HSEM1**	**HSEM1**
Reset	0	0	0	0	0	0	0	0

Host Controller Interface Modules (Continued)

Bit	Type	Description
3-0	RO	**HSEM3-0.** Four bits that may be updated by the host and read by both the host and the core.
7-4	R/W	**CSEM3-0.** Four bits that may be updated by the core and read by both the host and the core.

Shared Memory Core Override Read Protect Registers 0-2 (SMCORP0-2)

SMCORP0-2 are 16-bit registers that provide core override on the host read protection bits. For the host to be able to read a memory location, both the Host Read Protection bit (controlled through the Shared Memory Host Access Protect Register 1 or 2) and the associated bit in SMCORP0-2 should be cleared. Each bit in this register is associated with a memory block, as described in the bits description. Bits in these registers may be RO or RW depending on their position and the size of the core and host boot blocks. SMCORP0-2 registers are loaded with reset values either on reset or when the value of the SMCTA register is changed; the reset values depend on the size of the core and host boot blocks, as defined in the protection word.

Location: 00 F910$_{16}$, 00 F912$_{16}$, 00 F914$_{16}$

Type: R/W or RO as described in the description below

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ORPLA15-0**															
Reset	See bit description below								1	See bit description below						

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ORP15-2**														**Reserved**	
Reset	See bit description below														0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ORP 31-16**															
Reset	See bit description below															

Bit	Description
15-0	**ORPLA15-0 (Override Read Protect Low Addresses 15 through 0).** **ORP15-2 (Override Read Protect 15 through 2).** **ORP31-16 (Override Read Protect 15 through 0).** Each bit affects the host's ability to read from one block. On the low addresses (covered by ORPLAi), the block size is 8 Kbytes. For the other blocks it is 64 Kbytes. The block address is calculated as follows: Low address blocks, ORPLAi: from i*8K to (i+1)*8K–1 Other blocks, ORPLj: from j*64K to (j+1)*64K See Figure 102 on page 302 for a description of the block mapping. Bit 7 (ORPLA7) in SMCORP0 register is read only. 0: Do not override the host read protect setting for the block 1: Host read for this block is disabled regardless of the setting of the respective bit in the host register The reset values of these registers are as follows: Core boot blocks that cover address 00 0000$_{16}$ to CR_Boot_Block_Size: 1 Host boot blocks not in above group and from (MBTA[1] – Host_Boot_Block_Size) to MBTA: 0 All other blocks: 1 The following access limitations apply to the register bits: Core boot blocks that cover address 00 0000$_{16}$ to CR_Boot_Block_Size: RO Host boot blocks not in above group and ranging from (MBTA[1] – Host_Boot_Block_Size) to MBTA:RO All other blocks: RW

[1] Use the MBTA as specified in SMCTA register, without the MBTA override in the protection word.

Host Controller Interface Modules (Continued)

Shared Memory Core Override Write Protect Registers 0-2 (SMCOWP0-2)

SMCOWP0-2 are 16-bit registers that provide override on the host write protection bits. For the host to be able to write a memory location, both the Host Write Protection bit (controlled through the Shared Memory Host Access Protect Register 1 or 2) and the associated bit in SMCOWP0-2 should be cleared. Each bit in this register is associated with a memory block, as described in the bit description. Bits in these registers may be RO or RW depending on their position and the size of the core and host boot blocks. These registers are cleared on reset and change in value of SMCTA.

Location: 00 $F920_{16}$, 00 $F922_{16}$, 00 $F924_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**OWPLA15-0**															
Reset	$FFFF_{16}$															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**OWP15-2**														**Reserved**	
Reset	$FFFF_{16}$															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**OWP 31-16**															
Reset	$FFFF_{16}$															

Bit	Description
15-0	**OWPLA15-0(Override Write Protect Low Addresses 15 through 0).** **OWP15-2(Override Write Protect 15 through 2).** **OWP31-16(Override Write Protect 15 through 0).** Each bit affects the host’s ability to write to one block. On the low addresses (covered by OWPLAi), the block size is 8 Kbytes. For the other blocks it is 64 Kbytes. The block address is calculated as follows: Low Address Blocks, OWPLAi: from i*8K to (i+1)*8K–1 Other Blocks, OWPLj: from j*64K to (j+1)*64K See Figure 102 on page 302 for a description of the block mapping. Bit 7 (OWPLA7) in SMCOWP0 register is read only. 0: Do not override the host Write Protect setting for the block 1: Host writes for this block are disabled regardless of the setting of the respective bit in the host register The following access limitations apply to the register’s bits: Core boot blocks that cover address 00 0000_{16} to CR_Boot_Block_Size: RO Host boot blocks not in above group and ranging from (MBTA[1] – Host_Boot_Block_Size) to MBTA:RO All other blocks: RW

[1] Use the MBTA as specified in SMCTA register, without the MBTA override in the protection word.

Random Number Generator Control/Status Register, RNGCS (PC87591S)

This register controls the Random Number Generator (RNG) unit. It also provides status of its operation and the validity of the numbers at its output. The register is cleared on reset.

Location: 00 F930$_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**CLKP**			**DVALID**	**RNGE**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	R/W	**RNGE (Random Number Generator Enable).** Enables the operation of the RNG. This bit should be cleared to reduce power consumption when the RNG is not required. Some wake-up time may apply before the first number is generated after RNG activation. 0: RNG disabled (default) 1: RNG enabled
1	RO	**DVALID (Data Valid).** Indicates that valid data may be read from RNGD register. 0: Data not valid; RNGD always reads 00$_{16}$ (default) 1: Data valid; a random number is read
4-2	R/W	**CLKP (Clock Pre-Scaler).** Defines the pre-scale ratio between the RNG internal clocks and the core clock. To change the CLP contents: (a) Clear RNGE; (b) Change CLKP (c); Set RNGE to re-enable the RNG. CLKP may be set to a value above the actual clock frequency of the core, but it should not be set to a lower value than the CLK frequency. **Bits** **4 3 2 Description** 0 0 0: 15-20 MHz (default) 0 0 1: 12-15 MHz 0 1 0: 9-12 MHz 0 1 1: 7-9 MHz 1 0 0: 6-7 MHz 1 0 1: 5-6 MHz 1 1 0: 4-5 MHz 1 1 1: 3-4 MHz
7-5		**Reserved.**

Host Controller Interface Modules (Continued)

Random Number Generator Data Register, RNGD (PC87591S)

This register provides the random number value of the RNG.

Location: 00 F932$_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**RNGD7-0**							

Bit	Description
7-0	**RNGD0-7 (Random Number Generator Data Bits).** Provides random data output from the RNG module. When DVALID in RNGCS register is cleared, RNGD register always reads 00$_{16}$. When DVALID in RNGCS register is set, the data read is a random number. Any random number can be read only once, after which it is marked as invalid and replaced by a new number.

5.3.10 Usage Hints

- After clearing (0) HLOCK bit, the core firmware should wait until FMBUSY bit is clear (0) before it accesses the on-chip flash (see ”Flash Memory Status Register (FLSR)” in Section 4.16.7 on page 228).
- Enable Access to the shared memory: Before any access may occur, the shared memory access must be enabled using the SIO Configuration registers (see Section 6.1.11 on page 349). To enable shared memory as a boot device, the SHBM strap should be set appropriately (see Section 2.2.11 on page 50).
- Host boot block may be implemented as part of the on-chip flash or in the external memory. In the latter case, access to the Host boot block must be enabled by the core after the memory access configuration is completed.
- For host shared memory read:
  1. Have host enable the memory for reading (if disabled), using SMHAP1 and SMHAP2 registers.
  2. Have core enable the memory for reading (in most cases, disabled by default), using SMCORP0-2 registers.
  3. Read from the designated address location.
- For host shared memory write or erase:
  1. Have host enable the desired location for write, using SMHAP1 and SMHAP2 registers (note that the memory may be disabled for reads if the hardware validation is in use).
  2. Have core enable the desired location for a write, using SMCOWP0-2 registers.
  3. Enable the host write operations by setting HLOCK bit in SMCCST register.
  4. To perform a page erase operation, the host should set PMER bit in SMHC register and write any data to an address in the page. The page is erased when HBUSY bit in SMHST register is cleared, and an erase error, if it occurs, is flagged by HEERR bit in SMHST register.
  5. To program a byte in the on-chip flash, the host should make sure PMER bit in SMHC register is clear and write the required data to the designated address location. The programing is completed when HBUSY bit in SMHST register is cleared. At this time, an erase error, if it occurs, is flagged by HPERR in SMHST register.
  6. Programing the expansion memory is done using read/write operations to addresses associated with that memory. The transactions are passed to the memory through the core bus and BIU modules. The software should perform the appropriate sequence of read and write operations as required by the flash device in use.
- At different stages of the flash programing by the host, communication between the core and host is required. Various mechanisms may be used for this, one of which is the Shared Memory Semaphore mechanism. This mechanism is tuned for host-initiated operations that use polling on the registers. The core may receive an interrupt or use polling to identify a semaphore change. An example of bit allocation is:

  Bit 0 - Host requests control of flash
  Bit 4 - Core grants control to host

  The sequence is:
  1. Host sets bit 0 to request control of bus.
  2. Core identifies that bit 0 is set and does the required operations, including setting HLOCK bit in SMCCST register to enable host access.
  3. Core sets bit 4, indicating to the host that memory access is granted.
  4. Host performs write/erase to the memory, as required.
  5. Host clears bit 0, indicating completion of the process.
  6. Core clears HLOCK and protects the memory.
  7. Core indicates completion of process by clearing bit 4.

5.4 CORE ACCESS TO HOST-CONTROLLED MODULES

The PC87591x enables the core to access the Host-Controlled module registers (e.g., host configuration module, RTC and MSWC), using the SuperI/O Internal Bus (SIB) controller.

Host-Controlled Modules Register Arbitration. Since the host processor software and the PC87591x firmware cannot access a Host-Controlled module simultaneously, they must communicate to prevent conflicts in Host-Controlled module register usage.

Access to the Host-Controlled modules is controlled via a lock bit for each module. When the relevant lock bit is cleared, access to the Host-Controlled modules registers by the host processor is enabled. When the relevant lock bit is set, access to the Host-Controlled module registers by the host processor is blocked (i.e., write operations are ignored and read operations return 00_{16}). Any attempt by the host to access the locked register is flagged by setting the respective bit in SIOLV register.

SIB Arbitration. The host and core should access the Host-Controlled modules only after preventing host access to the module (using lock bits, as explained in the previous paragraph). The SIB controller arbitrates SIB usage between the host and core. If a core transaction starts after an LPC transaction has started, it waits for the completion of the LPC transaction. If a core transaction starts before an LPC transaction starts, the core transaction finishes before handling the LPC transaction.

The PC87591x firmware may access the Host-Controlled modules only while the core domain is in Active mode, the Host Domain power plane is on and the host domain clock is enabled with a minimum operation frequency of TBD.

Core Read Operation. To perform a read operation by the core from a Host-Controlled module register:

1. Set CSAE bit in SIBCTRL register, if not already set.
2. Verify that both CSRD and CSWR bits in SIBCTRL register are cleared.
3. Select the device to be accessed by setting its respective bit in CRSMAE register, if not already set. All other bits in the register must be cleared.
4. Specify the offset of the register in the device in IHIOA register, if not already specified.
5. Write 1 to CSRD bit in SIBCTRL register.
6. Read the CSRD bit in SIBCTRL until it returns 0.
7. Read the data from IHD register.

Core Write Operation. To perform a write operation by the core from a Host-Controlled module register:

1. Set CSAE bit in SIBCTRL register, if not already set.
2. Verify that both CSRD and CSWR bits in SIBCTRL register are cleared.
3. Select the device to be accessed by setting its respective bit in CRSMAE register, if not already set. All other bits in the register must be cleared.
4. Specify the offset of the register in the device in IHIOA register, if not already specified.
5. Write the data to IHD register; this starts the write operation to the device.
6. Read the CSWR bit in SIBCTRL until it returns 0; this indicates the completion of the write transaction.

The following sequence is provided for minimal conflict between host and core in the use of Host-Controlled peripherals.

1. After arbitrating the use of the specific Host-Controlled modules with the host, set the corresponding lock bit (see LKSIOHA register).
2. Read and save all Host-Controlled module registers required for proper operation of the host. Beware of destructive reads.
3. After the Host-Controlled module access is complete, restore the Host-Controlled module registers saved in step 2.
4. Clear the corresponding lock bit to allow the host to access the Host-Controlled module.

When accessing the RTC, also:

1. To access locked memory locations in the RTC, set (1) RTCMR bit in SIBCTRL register to clear the RTC lock bits.
2. Access the RTC's CMOS-RAM and its registers. To prevent conflicts with the host software, the firmware should not read any of the RTC read-volatile registers.
3. After the RTC access is complete, if the RTC locking was removed, re-lock the RTC memory and restore the RTC memory bank selection and the address pointer.

5.4.1 Core Access to Host-Controlled Module Registers

The following set of registers is accessible only by the core. The registers are powered by V_{CC}.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Core Access to Host-Controlled Modules Register Map

Mnemonic	Register Name	Type
IHIOA	Indirect Host I/O Address	R/W
IHD	Indirect Host Data	R/W
LKSIOHA	Lock SuperI/O Host Access	R/W
SIOLV	SuperI/O Access Lock Violation	R/W1C
CRSMAE	Core to SIB Modules Access Enable	R/W
SIBCTRL	SIB Control	Varies per bit

Indirect Host I/O Address Register (IHIOA)

This register defines the host I/O address for read or write transactions from/to the Host-Controlled modules. The I/O address is an offset from the least significant bits of the device address. The accessed device is selected using the Core to SIB Modules Access Enable Register (CRSMAE); see page 316.

Location: 00 $FCE0_{16}$

Type: R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**Reserved**								**Indirect Host I/O Offset**							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Indirect Host I/O Offset.** Only offsets within the device range are allowed. Other offsets may have unpredictable results.
15-8	**Reserved.**

Indirect Host Data Register (IHD)

This register holds host data for read or write transactions from/to the Host-Controlled modules.

Location: 00 $FCE2_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Indirect Host Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Indirect Host Data.**

Host Controller Interface Modules (Continued)

Lock SuperI/O Host Access Register (LKSIOHA)

This register controls locking of host access to the Host-Controlled modules. All bits of this register, except bit 1, are cleared on reset. The bit 1 reset value is defined by RTC Lock Default bit in “Protection Word” on page 226.

Location: 00 FCE4$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**LKRTCHA**	**LKCFG**
Reset	0	0	0	0	0	0	see text	0

Bit	15	14	13	12	11	10	9	8
Name	**Reserved**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**LKCFG (Lock Configuration Registers Host Access).** 0: Host processor access to the Configuration registers is enabled 1: Host processor access to the Configuration registers is blocked
1	**LKRTCHA (Lock Real-Time Clock (RTC) Host Access).** 0: Host processor access to the RTC registers is enabled 1: Host processor access to the RTC registers is blocked
15-2	**Reserved.**

SuperI/O Access Lock Violation Register (SIOLV)

This register provides an error indication when a host lock violation occurs on Host-Controlled modules access.

Location: 00 FCE6$_{16}$

Type: R/W1C

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**RTCLV**	**CFGLV**
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	**Reserved**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**CFGLV (Configuration Register Lock Violation).** The bit is set (1) when the host processor attempts to access the configuration registers while LKFDCHA bit in LKSIOHA register is set.
1	**RTCLV (Real-Time Clock (RTC) Lock Violation).** The bit is set (1) when the host processor attempts to access the RTC while LKRTCHA bit in LKSIOHA register is set.
15-2	**Reserved.**

Host Controller Interface Modules (Continued)

Core to SIB Modules Access Enable Register (CRSMAE)

This register enables core access to the Host-Controlled modules. Only one of the bits in this register may be set at a time.

Location: 00 FCE8$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**RTCAE**	**CFGAE**
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Name	**Reserved**							**MSWCAE**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**CFGAE (Configuration Register Core Access Enable).** 0: Core access to the Configuration registers is disabled 1: Core access to the Configuration registers is enabled
1	**RTCAE (Real-Time Clock (RTC) Core** Access **Enable).** The RTC has two chip-select signals defined in its configuration space, each with two registers. A1 of the offset is used to differentiate between the two (when A1 is 0, the pair pointed to by index 60, 61 is accessed; when A1 is 1, the pair pointed to by index 62, 63 is accessed). 0: Core access to the RTC registers is disabled 1: Core access to the RTC registers is enabled
7-2	**Reserved.**
8	**MSWCAE (Mobile System Wake-Up Control (MSWC) Access Enable).** 0: Core access to the MSWC registers is disabled 1: Core access to the MSWC registers is enabled
15-9	**Reserved.**

SIB Control Register (SIBCTRL)

This register allows the core to control the SIB controller operation.

Location: 00 FCEA$_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**				**RTCMR**	**CSWR**	**CSRD**	**CSAE**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	R/W	**CSAE (Core to SIB Access Enabled).** 0: Core access to the SIB bus is disabled (default) 1: Core access to the SIB bus is enabled. The device to be accessed is selected by the SIB Device Select register.
1	R/W1S	**CSRD (Core Read from SIB).** Writing 1 to this bit starts a read from the SIB; the read is based on the address and enabled device specified in CRSMAE register. A write of 0 to this bit is ignored. This bit is cleared when the read operation is completed, indicating that the data is ready in IHD register.
2	RO	**CSWR (Core Write to SIB).** The bit is set by a write operation to IHD register. It is cleared when the write to the SIB is completed.
3	R/W1S	**RTCMR (Real-Time Clock (RTC) Master Reset).** Writing 1 to this bit generates a reset pulse to the RTC module. This bit is cleared by the hardware once the reset pulse is completed. Writing 0 to this bit is ignored.
7-4		**Reserved.**

5.5 MOBILE SYSTEM WAKE-UP CONTROL (MSWC)

The MSWC detects and handles wake-up events from various sources in the host-controlled modules. The MSWC generates interrupts to the host via $\overline{\text{SMI}}$ or $\overline{\text{PWUREQ}}$, and/or alerts the core, which enables the core to control the wake-up sequence. Since the MSWC is powered by V_{CC}, it can operate in low power consumption states. Some MSWC operations depend on the presence of a clock; these functions are not available when the core clock is turned off.

Figure 104 shows the block diagram of the MSWC.

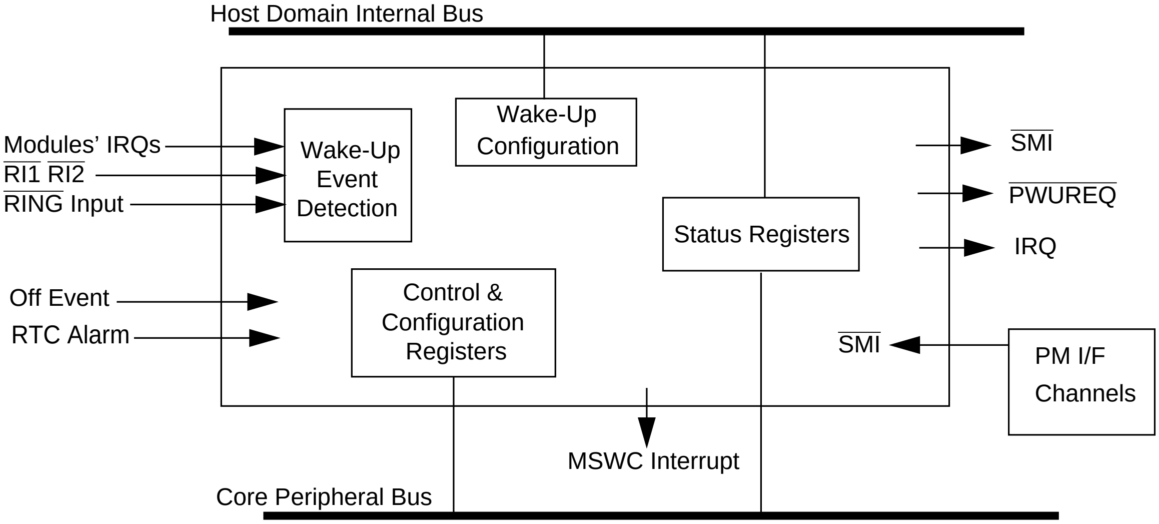


Figure 104. MSWC Block Diagram

5.5.1 Features

The MSWC recognizes the following maskable system events:

- Modem ring ($\overline{\text{RI1}}$ and $\overline{\text{RI2}}$ pins)
- Telephone ring ($\overline{\text{RING}}$ input pin)
- Wake-up on module IRQs for RTC, KBD and Mouse
- Software events
  - Software triggered wake-up event
  - ACPI power state change indications
  - Software off command

The MSWC notifies the host and/or core when any of the above events occur by asserting one or more of the following output pins:

- Power-Up Request ($\overline{\text{PWUREQ}}$)
- System Management Interrupt ($\overline{\text{SMI}}$)
- Interrupt to the host (IRQ)
- MSWC interrupt to the core

5.5.2 Wake-Up Event Detection and Status Bits

The MSWC monitors various system signals for a wake-up event. When an event is detected, a status bit is set to record it. Each event goes to the Wake-Up Mode Control Logic, which determines its effect (see Figure 105 for an illustration of this mechanism). A set of dedicated registers is used to determine the wake-up criteria, including the RING detection mode.

Host Controller Interface Modules (Continued)

The following wake-up input events are detected by the MSWC:

- Software events
- IRQ from SuperI/O modules
- Modem Ring ($\overline{\text{RI1}}$ and $\overline{\text{RI2}}$)
- Telephone Ring ($\overline{\text{RING}}$ input)
- ACPI state change
- Legacy off event
- RTC Alarm

When an input event is detected, the corresponding status bit in both host and core status registers is set to 1, regardless of any Routing Enable bit setting. If both the status bit and a Routing Enable bit corresponding to a specific event are set to 1 (no matter in what order), the output pin corresponding to that Routing Enable bit is asserted.

A status bit is cleared by writing 1 to it. Writing 0 to a status bit does not change its value. Clearing the routing enable bit of an event prevents it from issuing the corresponding system notification (output event) but does not affect the status bit. Figure 105 shows the routing scheme of detected wake-up events to the various means of system notification (i.e., output events).

Both the core and the host have status registers; thus both core and host software can monitor the various event status bits. This enables handling of events via wake-up logic that is implemented as part of the Embedded Controller firmware, and passing the wake-up notifications through the Power Management host-interface protocol. The core uses a mask register (WK_SMIENn) to define which of the status bits it should respond to.

It is recommended that each of the wake-up sources be handled by one handler routine on the host (i.e., SMI, SCI or IRQ triggered) or the core.

Software Event

A software event may be used to trigger a wake-up event via software control. A software event to the host is active when the Software Event Status bit in WK_STS0 register is set. A software event to the core is active when the Software Event Status bit in MSHES0 register is set.

The software event may be activated (i.e., the status bits in WK_STS0 and MSHESO registers are set) by writing 1 to the Software Event Status bit in WK_STS0 register when the bit is cleared. Once this bit is set to 1, writing 1 to the bit clears it.

WK_STS0 is accessible by the host only when V_{DD} is present. The software event can be activated by the host only when V_{DD} is present. The software event may be activated by the core by the core access to the Host-Controlled Module bridge even when V_{DD} is off.

When HSECM bit in MSWCTL1 register is cleared, the Software Event Status bit in MSHES0 is set on a change of the respective bit in WK_STS0 register from 0 to 1. When HSECM bit is set, the Software Event Status bit in MSHES0 is set on a write of a 1 to the respective bit in WK_STS0. The Software Event Status bit in MSHES0 is cleared by writing 1 to it.

Module IRQ Wake-Up Event

A module IRQ wake-up event is defined as the leading edge of the IRQ assertion of the RTC.

To enable the IRQ of a specific logical device to trigger a wake-up event, the associated enable bit must be set to 1. This is bit 4 of the Interrupt Number and Wake-Up on IRQ Enable register, located at index 70_{16} in the configuration space of the logical device (see Table 52 on page 339). When this bit is set, any IRQ assertion of the corresponding logical device activates the module IRQ wake-up event. Therefore, the module IRQ wake-up event is a combination of all IRQ signals of the logical devices for which wake-up on IRQ is enabled.

When the event is detected as active, its associated status bit (bit 7 of WK0_STS register) is set to 1. If the associated enable bit (bit 7 of WK_EN0 register) is also set to 1, the $\overline{\text{PWUREQ}}$ output is asserted and remains asserted until the status bit is cleared.

Since V_{DD} powers IRQ generation of the logical devices, a module IRQ event can be activated only when V_{DD} is present (see Section 6.1 on page 335 for a list of logical devices).

Modem Ring

High to low transitions on $\overline{\text{RI1}}$ (or $\overline{\text{RI2}}$) indicate the detection of a ring in an external modem connected to Serial Port 1 (or Serial Port 2) and can be used as wake-up events.

Telephone Ring

A telephone ring is detected by the MSWC by processing the raw signal coming directly from the telephone line into the $\overline{\text{RING}}$ input pin. Detection of a pulse train, with a frequency higher than 16 Hz lasting at least 0.19 sec, is used as a wake-up event.

Host Controller Interface Modules (Continued)

The RING pulse-train detection is achieved by monitoring the falling edges on RING in time slots of 62.5 msec (a 16 Hz cycle). A positive detection occurs if falling edges of RING are detected in three consecutive time slots, following a time slot in which no RING falling edge is detected. This detection method guarantees the detection of a RING pulse train with frequencies higher than 16 Hz. It filters out (does not detect) pulses of less than 10 Hz and may detect pulses between 10 Hz and 16 Hz.

ACPI State Change and Legacy Off Events

The host may operate in either Legacy or ACPI mode. The operation mode is specified by the Power Button Mode bit in SuperI/O Configuration D register (SIOCFD). When EICFGPBM bit in MSIEN2 register is set, a change to the Power Button Mode bit generates an interrupt to the core. The core may read the value of the Power Button Mode bit, using CFGPBM bit in MSWCTL2 register, to determine how to interpret the other power state request bits.

The Power Supply Off bit in SIOCFD register may be used in Legacy mode to indicate a request to turn power off. A write of 1 to this bit sets CFGPSO bit in MSWCTL2 register; then, if EICFGPSO bit in MSIEN2 register is set, an interrupt to the core is generated, indicating the event.

A set of System State Change Request bits (S1-S5) are provided in WK_STATE register. The host uses these bits for ACPI-compliant state change requests. A write of 1 to any of these bits indicates a state change request to the core through the respective bit in MSWCTL2 register. When all bits in WK_STATE are written with 0, a request of S0 is indicated, and ACPIS0 bit in MSWCTL2 register is set. When any S0-S5 bit in MSWCTL2 is set and the respective mask bit in MSIEN2 register is set, an interrupt to the core is generated whenever a change to any of the state bits is detected.

All interrupt requests may be cleared by writing 1 to the corresponding status bit or by masking the event (by clearing the corresponding Interrupt Enable bit).

RTC Alarm

The RTC module may generate an ALARM signal (see Section 6.2.8 on page 359). The RTC alarm can serve as a wake-up request to wake up the system; the request is routed to the core, which then wakes up the system. To enable an alarm wake-up, the following settings should be made:

- Set the Alarm conditions in the RTC module. By masking the various interrupts, software may select a wake-up either to the host directly, using the RTC's IRQ, or to the core through the Alarm signal.
- Enable the Wake-Up on Alarm status interrupt masking (optional, for Interrupt Enabled mode) by setting EIRTCAL bit in MSIEN2 register.
- Verify that the RTCAL bit in MSWCTL3 bit is cleared (no pending Alarm request).
- Enable the Wake-Up on MSWC event in the MIWU and ICU modules.
- Verify that the ALARM bit in the RTC is cleared.

After an ALARM event is detected in the RTC, the RTC ALARM status bit is set (bit 5 in CRC register, page 369); in response, RTCAL bit in MSWCTL3 register is set.

When handling an ALARM event, make sure that no events are lost by clearing RTCAL bit before clearing the ALARM status in the RTC.

5.5.3 Wake-Up Output Events

The MSWC generates four types of output events:

- IRQ - an interrupt routed as configured in the MSWC PnP configuration registers.
- PWUREQ - an event that is typically routed to an input in the chipset that triggers an SCI event.
- SMI - an event typically connected to an input in the chipset that triggers an SMI event.
- MSWCI - an interrupt to the MIWU module in the core domain. This enables the core firmware to handle the wake-up events.

Figure 105 illustrates the enabling mechanism and the event generation scheme for the various output events. Output events to the host are generated for input events that have their status bit set (WK_STSn.i is 1). Output events to the core, through the MIWU, are generated for input events that have their core status bit set (MSHESn.ii is 1).

Each of the three Host Wake-Up Event Routing Control registers (WK_ENn, WK_SMIENn and WK_IRQENn) holds a Routing Enable bit for each event; this allows selective routing of these events to PWUREQ, SMI and/or the assigned MSWC interrupt request (IRQ) channel, respectively.

After an output event is asserted, it is active until all set status bits are cleared or masked. The current status of the event may be read at the ACPI status registers in the chipset's ACPI controller or by reading "Wake-Up Event Status Register 0 (WK_STS0)" on page 323 and "Wake-Up Signals Value Register (WK_SIGV)" on page 325.

As shown in Figure 106, for SMI output events, the MSWC combines the event request coming from the Host Interface's Power Management channels 1 and 2 with MSWC internal SMI events; for PWUREQ output events, the MSWC combines the PWUREQ events generated in the MSWC module with wake-up requests associated with the IRQ configuration registers.

Host Controller Interface Modules (Continued)

The $\overline{\text{SMI}}$ may be output from the PC87591x using the dedicated $\overline{\text{SMI}}$ signal or by routing $\overline{\text{SMI}}$ to an interrupt request channel via the device's configuration registers.

The Wake-Up Event Routing Control register, MSHEIEn, which is controlled by the core, holds an enable bit for each of the events, which allows selective routing of these events to the core wake-up interrupt (MSWCI) to the MIWU. The core event is controlled using a separate set of status signals to prevent race conditions when clearing events.

The MSWCI interrupt is a level high interrupt that gathers requests from MSHESn, MSWCTL2 and MSWCTL3 registers. Once an output event is asserted, it keeps its active state until all set status bits are cleared or masked. This interrupt signal is connected to the MSWC wake-up input of the MIWU. This enables handling state change requests even in Idle mode. The MSWC output for this input is connected to the core through the MIWU module, enabling a power state change on interrupt.

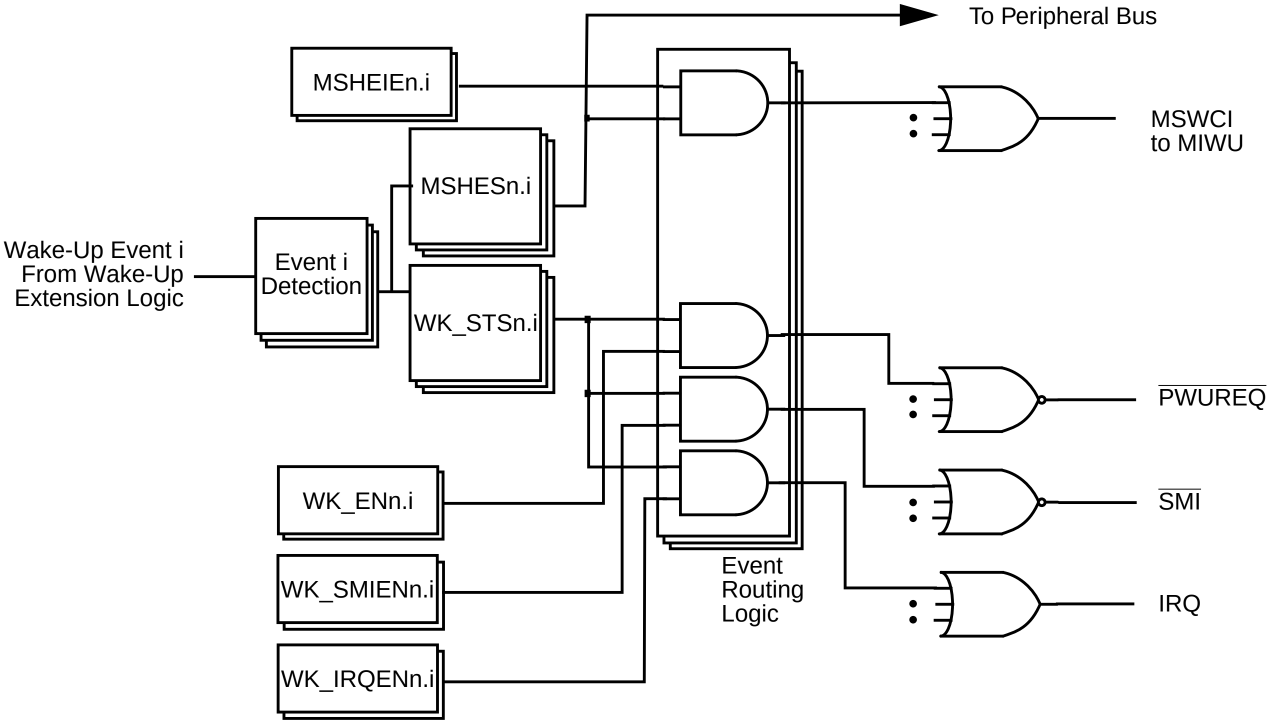
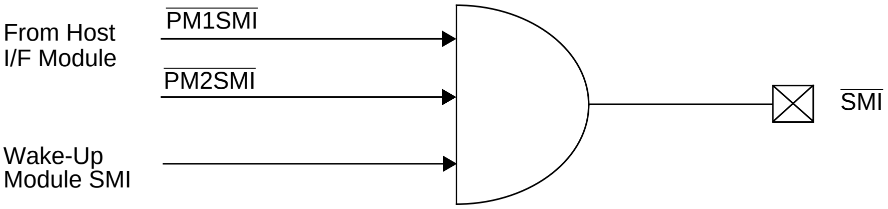


Figure 105. Wake-Up Event Routing Scheme



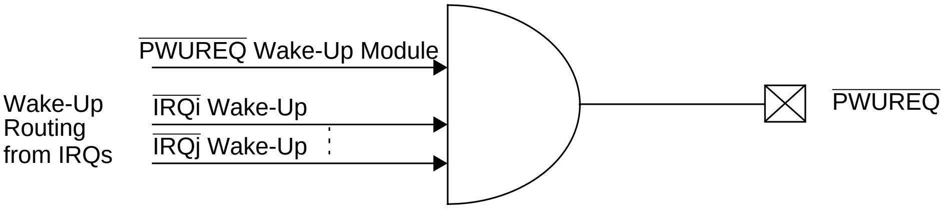


Figure 106. $\overline{\text{SMI}}$ and $\overline{\text{PWUREQ}}$ Source Gathering Scheme

Host Controller Interface Modules (Continued)

5.5.4 Other MSWC Controlled Elements

In addition to its Power Management functions, the MSWC controls the handling of the following system control elements:

- Host Configuration Address Selection
- Host Keyboard Reset Fast Reset Output ($\overline{\text{KBRST}}$)
- GA20 Pin Functionality
- Host Power on indication

Host Configuration Address Selection

The standard strap configuration enables the selection of one of two SuperI/O configuration register addresses. When the PC87591x is enabled in Programmable Configuration Address mode (see Table 47 on page 335), the core may set the address of the SuperI/O configuration index/data registers.

HCFGBAL and HCFGBAH are byte-wide read/write registers. HCFGBAL holds the least significant byte of a host motherboard PnP initial configuration address; HCFGBAH holds the most significant byte. The contents of HCFGBAH and HCFGBAL change only during V_{CC} Power-Up reset.

To update the base address of the SuperI/O configuration index/data registers, do the following:

1. Clear VHCFGA bit in MSWCTL1 register by writing 1 to it.
2. Write the lower byte of the address to HCFGBAL (MSB must be cleared).
3. Write the higher byte of the address to HCFGBAH.
4. Set HCFGLK bit to prevent an accidental change of the address written to HCFGBAL and HCFGBAH.

The base address is preserved by V_{CC}, and VHCFGA is set as long as a valid address is maintained. If there is no valid configuration base address, the LPC interface does not respond to configuration requests.

Host Keyboard Fast Reset

The Host Keyboard Reset output ($\overline{\text{KBRST}}$) is an output of the PC87591x that serves as one of the sources for Host Soft reset commands (i.e., $\overline{\text{INIT}}$ input in the x86 processors). Figure 107 shows the $\overline{\text{KBRST}}$ generation scheme. The host is reset when the $\overline{\text{KBRST}}$ output is low. A reset command is issued by the PC87591x by software or hardware, as follows:

- Software: The core firmware can issue a reset command to the host by writing 1 to HRSTOB in MSWCTL1 register. The reset to the host ends by writing 0 to this bit.
- Hardware: The host is reset during V_{CC} Power-Up reset if HRAPU bit in MSWCTL3 register is set and an LPC transaction is started. This is used to prevent accesses to the PC87591x from being ignored due to the duration of the Power-Up reset.

.

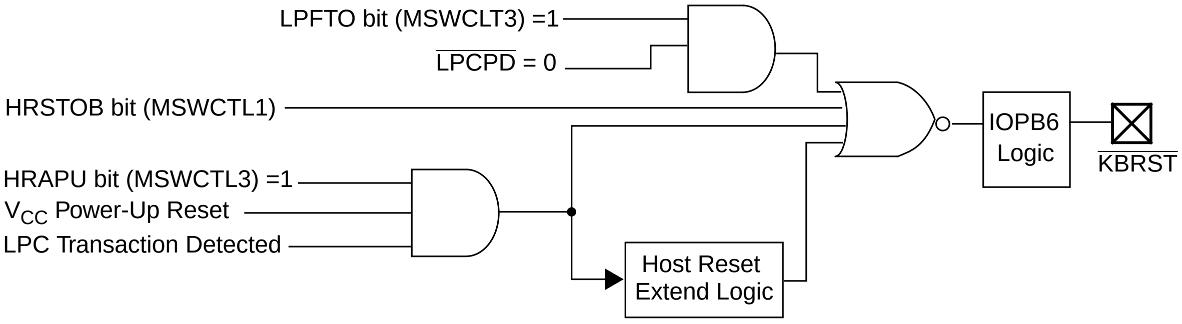


Figure 107. $\overline{\text{KBRST}}$ Generation Scheme

GA20 Pin Functionality

The GA20 (Gate Address A20) function is part of the PC architecture. In PC87591x, the GA20 function is implemented by a GPIO signal that is configured as output. Port PB5 is recommended to be used as GA20 since its default state after reset is output driving high. The firmware running on the core may change the GA20 signal state by modifying bit 5 in PBDOUT register. There is no special hardware or multiplexing on PB5; since there is no multiplexing, bit 5 of PBALT register is always 0 and any writes to it are disregarded. PB5 may be used as a GPIO; however, note that wake-up for PB5 differs from the other signals in port B.

Host Controller Interface Modules (Continued)

5.5.5 MSWC Host Registers

The MSWC registers are organized in four banks, all of which are battery-backed. The offsets are related to a base address that is determined by the MSWC Base Address register in the device configuration registers. The lower 19 offsets are common to the four banks; the upper offsets (13_{16}-$1F_{16}$) are divided as follows:

- Bank 0 is reserved.
- Bank 1 is reserved.
- Bank 2 holds the Event Routing Configuration and Wake-Up Extension Control registers.

The active bank is selected through the Configuration Bank Select field (bits 1-0) in the Wake-Up Configuration register (WK_CFG).

As a programing aid, the registers are described in this chapter according to the following functional groupings:

- General status
- Enable
- Configuration
- Routing

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

MSWC Host Register Map

The following tables list the MSWC host registers. For the MSWC core register map, see Section 5.5.6 on page 328.

Table 44. Banks 0, 1, 2 and 3 - The Common Control and Status Register Map

Offset	Mnemonic	Register Name	Type
00_{16}	WK_STS0	Wake-Up Event Status 0	R/W1C
02_{16}	WK_EN0	Wake-Up Enable 0	R/W
04_{16}	WK_CFG	Wake-Up Configuration	R/W
06_{16}	WK_SIGV	Wake-Up Signal Value	RO
07_{16}	WK_STATE	Wake-Up ACPI State	WO
Other	Reserved		

Table 45. Bank 2 - Event Routing Configuration Register Map

Offset	Mnemonic	Register Name	Type
13_{16}	WK_SMIEN0	Wake-Up SMI Enable 0	R/W
15_{16}	WK_IRQEN0	Wake-Up Interrupt Request Enable 0	R/W
Other	Reserved		

Wake-Up Event Status Register 0 (WK_STS0)

This register is set to 00_{16} on V_{PP} power-up, V_{CC} power-up or Host Domain Software reset. It indicates which wake-up events, associated with the register, have occurred. Writing 1 to a bit clears it to 0. Writing 0 has no effect. Bit 6 behaves in a special way, as described in the table below.

Location: Offset 00_{16}

Type: R/W1C

Bit	7	6	5	4	3	2	1	0
Name	**Module IRQ Event Status**	**Software Event Status**	**Reserved**		**$\overline{\text{RING}}$ Event Status**	**Reserved**	**$\overline{\text{RI2}}$ Event Status**	**$\overline{\text{RI1}}$ Event Status**
Reset	0	0	0	0	0	0	0	0

Host Controller Interface Modules (Continued)

Bit	Description
0	**$\overline{\text{RI1}}$ Event Status.** 0: Event not detected (default) 1: Event detected
1	**$\overline{\text{RI2}}$ Event Status.** 0: Event not detected (default) 1: Event detected
2	**Reserved.**
3	**$\overline{\text{RING}}$ Event Status.** $\overline{\text{RING}}$ event detection, according to the RING detection mode enabled. 0: Event not detected (default) 1: Event detected
5-4	**Reserved.**
6	**Software Event Status.** This bit may work in two modes, as defined by HSECM bit in MSWCTL1 register (see "MSWC Control Status Register 1 (MSWCTL1)" on page 329). When HSECM is 0, writing 1 to Software Event Status bit inverts its value. When HSECM is 1, writing 1 to Software Event Status bit sets it; the bit is cleared by a write of 1 to bit 6 in MSHES0 register. 0: Event not active (default) 1: Event active
7	**Module IRQ Event Status.** This sticky bit shows the status of the module IRQ event detection. 0: Event not active (default) 1: Event active

Wake-Up Events Enable Register (WK_EN0)

This register is set to 00_{16} on V_{PP} power-up or Host Domain Software reset. When enabled wake-up events are detected, the $\overline{\text{PWUREQ}}$ signal is activated.

Location: Offset 02_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Module IRQ Event Enable**	**Software Event Enable**	**Reserved**		**$\overline{\text{RING}}$ Event Enable**	**Reserved**	**$\overline{\text{RI2}}$ Event Enable**	**$\overline{\text{RI1}}$ Event Enable**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**$\overline{\text{RI1}}$ Event Enable.** 0: Disabled (default) 1: Enabled
1	**$\overline{\text{RI2}}$ Event Enable.** 0: Disabled (default) 1: Enabled
2	**Reserved.**
3	**$\overline{\text{RING}}$ Event Enable.** 0: Disabled (default) 1: Enabled

Host Controller Interface Modules (Continued)

Bit	Description
5-4	**Reserved.**
6	**Software Event Enable.** 0: Disabled (default) 1: Enabled
7	**Module IRQ Event Enable.** 0: Disabled (default) 1: Enabled

Wake-Up Configuration Register (WK_CFG)

This register is set to 00_{16} on V_{PP} power-up or Host Domain Software reset. It enables access to Event Routing Control registers (bank selected).

Location: Offset 04_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**Configuration Bank Select**	
Reset	0	0	0	0	0	0	0	0
Required	0	0						

Bit	Description
1-0	**Configuration Bank Select.** **Bits** **1 0 Bank Register** 0 0: 0 Reserved 0 1: 1 Reserved 1 0: 2 Event Routing, Wake-Up Extension 1 1: 3 Reserved
7-2	**Reserved.**

Wake-Up Signals Value Register (WK_SIGV)

This is a read-only register that returns the value of $\overline{\text{SMI}}$ and $\overline{\text{PWUREQ}}$ signal output and input to this module. This register helps to identify the source of the wake-up request when multiple sources are enabled.

Location: Offset 06_{16}

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**PWUREQ of Wake-Up Value**	**PWUREQ Output Value**	**PM2 SMI Output Value**	**PM1 SMI Output Value**	**SMI Wake-Up Output Value**	**SMI Output Value**

Host Controller Interface Modules (Continued)

Bit	Description
0	**SMI Output Value.** 0: $\overline{\text{SMI}}$ output is low (asserted) 1: $\overline{\text{SMI}}$ output is high (de-asserted)
1	**SMI Wake-Up Output Value.** 0: $\overline{\text{SMI}}$ output of the wake-up module is low (asserted) 1: $\overline{\text{SMI}}$ output of the wake-up module is high (de-asserted)
2	**PM1 SMI Output Value.** 0: $\overline{\text{SMI}}$ output of the Power Management channel 1 is low (asserted) 1: $\overline{\text{SMI}}$ output of the Power Management channel 1 is high (de-asserted)
3	**PM2 SMI Output Value.** 0: $\overline{\text{SMI}}$ output of the Power Management channel 2 is low (asserted) 1: $\overline{\text{SMI}}$ output of the Power Management channel 2 is high (de-asserted)
4	**PWUREQ Wake-Up Output Value.** 0: $\overline{\text{PWUREQ}}$ output is low (asserted) 1: $\overline{\text{PWUREQ}}$ output is high (de-asserted)
5	**PWUREQ Wake-Up Value.** 0: $\overline{\text{PWUREQ}}$ output of the wake-up module is low (asserted) 1: $\overline{\text{PWUREQ}}$ output of the wake-up module is high (de-asserted)
7-6	**Reserved.**

Wake-Up ACPI State Register (WK_STATE)

This is a read/write register. It always returns 00_{16} when read.

Location: Offset 07_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**S5**	**S4**	**S3**	**S2**	**S1**	**Reserved**

Bit	Description
0	**Reserved.**
1	**S1 (Request to Change to S1 State).** A write of 1 to this bit indicates to the core that the host requests to change to S1 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0. 0: Not an S1 state request 1: S1 state setting request
2	**S2 (Request to Change to S2 State).** A write of 1 to this bit indicates to the core that the host requests to change to S2 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests This bit always reads back 0. 0: Not an S2 state request 1: S2 state setting request

3	**S3 (Request to Change to S3 State).** A write of 1 to this bit indicates to the core that the host requests to change to S3 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0. 0: Not an S3 state request 1: S3 state setting request
4	**S4 (Request to Change to S4 State).** A write of 1 to this bit indicates to the core that the host requests to change to S4 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0. 0: Not an S4 state request 1: S4 state setting request
5	**S5 (Request to Change to S5 State).** A write of 1 to this bit indicates to the core that the host requests to change to S5 state. The S state and transition are interpreted, as specified in the ACPI standard for S state change requests. This bit always reads back 0. 0: Not an S5 state request 1: S5 state setting request
7-6	**Reserved.**

Wake-Up Event Routing to $\overline{\text{SMI}}$ Enable Register 0 (WK_SMIEN0)

This register is set to 00_{16} on V_{PP} power-up or Host Domain software reset. It controls the routing of detected wake-up events to the $\overline{\text{SMI}}$ signal. Detected wake-up events that are enabled activate the $\overline{\text{SMI}}$ signal regardless of the value of WK_EN0 register.

Location: Bank 2, Offset 13_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**Software Event to $\overline{\text{SMI}}$ Enable**	**Reserved**		**$\overline{\text{RING}}$ Event to $\overline{\text{SMI}}$ Enable**	**Reserved**	**$\overline{\text{RI2}}$ Event to $\overline{\text{SMI}}$ Enable**	**$\overline{\text{RI1}}$ Event to $\overline{\text{SMI}}$ Enable**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**$\overline{\text{RI1}}$ Event to $\overline{\text{SMI}}$ Enable.** 0: Disabled (default) 1: Enabled
1	**$\overline{\text{RI2}}$ Event to $\overline{\text{SMI}}$ Enable.** 0: Disabled (default) 1: Enabled
2	**Reserved.**
3	**$\overline{\text{RING}}$ Event to $\overline{\text{SMI}}$ Enable.** 0: Disabled (default) 1: Enabled
5-4	**Reserved.**
6	**Software Event to SMI Enable.** 0: Disabled (default) 1: Enabled
7	**Reserved.**

Wake-Up Event Routing to IRQ Enable Register 0 (WK_IRQEN0)

This register is set to 00_{16} on V_{PP} power-up or Host Domain Software reset. It controls the routing of detected wake-up events to the assigned MSWC interrupt request (IRQ) channel. Detected wake-up events that are enabled activate the assigned IRQ channel regardless of the value of WK_EN0 register.

Location: Bank 2, Offset 15_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**Software Event to IRQ Enable**	**Reserved**		**$\overline{\text{RING}}$ Event to IRQ Enable**	**Reserved**	**$\overline{\text{RI2}}$ Event to IRQ Enable**	**$\overline{\text{RI1}}$ Event to IRQ Enable**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**$\overline{\text{RI1}}$ Event to IRQ Enable.** 0: Disabled (default) 1: Enabled
1	**$\overline{\text{RI2}}$ Event to IRQ Enable.** 0: Disabled (default) 1: Enabled
2	**Reserved.**
3	**$\overline{\text{RING}}$ Event to IRQ Enable.** 0: Disabled (default) 1: Enabled
5-4	**Reserved.**
6	**Software Event to IRQ Enable.** 0: Disabled (default) 1: Enabled
7	**Reserved.**

5.5.6 MSWC Core Registers

For a summary of the abbreviations used for Register Type, see Section 2 on page 34.

Table 46. MSWC Core Register Map

Mnemonic	Register Name	Type
MSWCTL1	MSWC Control Status Register 1	Varies per bit
MSWCTL2	MSWC Control Status Register 2	Varies per bit
MSWCTL3	MSWC Control Status Register 3	R/W
HCFGBAL	Host Configuration Base Address Low	R/W
HCFGBAH	Host Configuration Base Address High	R/W
MSIEN2	MSWC Interrupt Enable Register 2	R/W
MSHES0	MSWC Host Event Status Register 0	R/W1C
MSHEIE0	MSWC Host Event Interrupt Enable Register	R/W

Host Controller Interface Modules (Continued)

MSWC Control Status Register 1 (MSWCTL1)

This is a byte-wide read/write register that controls the settings associated with host wake-up and activity. The contents of this register are preserved by V_{CC}. Bit 0 is cleared by Warm reset; other bits are reset only on V_{CC} Power-Up reset.

Location: 00 $FCC0_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**HSECM**	**HCFGLK**	**VHCFGA**	**LPCRSTA**	**HPWRON**	**HRSTOB**
Reset	0	0	0	0	0	0	-	-

Bit	Type	Description
0	R/W	**HRSTOB (Host Reset Out Bit).** Enables the PC87591x to generate a Host Soft reset via firmware, using the $\overline{\text{KBSTO}}$ pin. The pin is held low (reset is active) for as long this bit 1. 0: $\overline{\text{KBRST}}$ is not forced active (default) 1: Force $\overline{\text{KBRST}}$ active
1	RO	**HPWRON (Host Power On).** The V_{DD} power detection logic indicates that V_{DD} is on. 0: V_{DD} is off (below V_{DDON}) 1: V_{DD} is on (above V_{DDON})
2	RO	**LPCRSTA (LPC Reset Active).** The $\overline{\text{RESET1}}$ input is active (low). 0: $\overline{\text{RESET1}}$ is not active (high) 1: $\overline{\text{RESET1}}$ is active (low)
3	R/W1C	**VHCFGA (Valid Host Configuration Address).** This bit is set by a write to HCFGBAH register, as detailed in the update sequence in “Host Configuration Address Selection” on page 322. The firmware can clear the bit by writing 1 to it. This register is used as the address of the Configuration registers when the PC87591x is set to operate with the internal base address. Writing 0 to this bit is ignored. This bit can be locked and made read-only by setting HCFGLK (bit 4). 0: Host Configuration Registers base address is not valid and access to this registers by the host is not enabled (default) 1: Host Configuration Registers base address is specified in HCFGBAH and HCFGBAL registers
4	R/W1S	**HCFGLK (Host Configuration Address Lock).** This bit is cleared during V_{CC} power-up, WATCHDOG reset or Debugger Interface reset, but is unchanged during other reset events. When 1 is written to this bit, it becomes read only (i.e., it cannot be cleared by the firmware) and locks VHCFGA bit, HCFGBAH register and HCFGBAL register, preventing accidental alteration to them. 0: Allows update of the Host Configuration Registers base address (default) 1: Locks the Host Configuration Registers base address
5	R/W1C	**HSECM (Host Software Event Clear Mode).** Controls the clear mode of Host Software Event Status bit in WK_STS0. This bit is cleared at V_{CC} power-up and $\overline{\text{RESET1}}$ events. 0: Host Software Event Status bit in WK_STS0 (bit 6) toggles on host writes of 1. MSHES0 bit 6 is set when WK_STS0 bit 6 changes from 0 to 1 (default). 1: Host Software Event Status bit in WK_STS0 (bit 6) and MSHES0 bit 6 are both cleared by writes of 1 to MSHES0 register
7-6		**Reserved.**

Host Controller Interface Modules (Continued)

MSWC Control Status Register 2 (MSWCTL2)

This is a byte-wide read/write register that controls the settings associated with host wake-up and activity. Bits in this register are cleared by V_{CC} Power-Up and $\overline{\text{RESET1}}$ resets.

Location: 00 $FCC2_{16}$

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**CFGPSO**	**CFGPBM**	**ACPIS5**	**ACPIS4**	**ACPIS3**	**ACPIS2**	**ACPIS1**	**ACPIS0**
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
0	R/W1C	**ACPIS0 (ACPI request for S0).** This bit may be used by ACPI software to directly request a change of power state. This bit is set when the host software writes a value of 0 to bits S1 through S5 in WK_STATE register. This bit is cleared by writing 1 to it. A write of 0 is ignored. 0: No pending request for S0 change (default) 1: A request for S0 change was detected
5-1	R/W1C	**ACPIS1-5 (ACPI request for S1 through S5).** These bits may be used by ACPI software to directly request a change of power state. These bits are set by a host software write of 1 to the respective bit in WK_STATE register. The bit is cleared by writing 1 to it. A write of 0 is ignored. When any ACPIS0 through ACPIS5 bit is set, an APC interrupt to the core, using a MIWU input, is asserted.
6	RO	**CFGPBM (SuperI/O Configuration Register D Power Button Mode).** This bit reflects the current status of the Power Button Mode bit in SIOCFD register. This bit may be used by the host software to specify to the core he method used for power off signaling. See “SuperI/O Configuration D Register (SIOCFD)” on page 347 A write of 1 clears the interrupt signal caused by a change in this bit value. A write of 0 to this bit is ignored.
7	RO	**CFGPSO (SuperI/O Configuration Register D Power Supply Off).** This bit is set whenever a 1 is written to the Power Supply Off bit in SIOCFD register. This bit may be used by the host software to specify to the core that the power supply should be turned off, in a non-ACPI system. See “SuperI/O Configuration D Register (SIOCFD)” on page 347. A write of 1 clears this bit and the interrupt signal caused by a change in this bit value. A write of 0 to this bit is ignored.

MSWC Control Status Register 3 (MSWCTL3)

This is a byte-wide read/write register that controls the settings associated with host wake-up and activity. The contents of this register is preserved by V_{PP} and it is reset only on V_{PP} Power-Up reset.

Location: 00 $FCC4_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**					**RTCAL**	**LPFTO**	**HRAPU**
Reset	0	0	0	0	0	0	0	1

Host Controller Interface Modules (Continued)

Bit	Description
0	**HRAPU (Host Reset when Accessed During V_{CC} Power-Up Reset).** Indicates that a reset should be sent to the host if LPC activity was detected while the V_{CC} Power-Up reset was not completed. This intends to re-start any LPC transaction that may have addressed the PC87591x but could not be handled correctly. When HCFGLK bit is set, writes to this bits are ignored. 0: Do not generate a reset on LPC transactions while the PC87591x is in Power-Up reset 1: Assert $\overline{\text{KBRST}}$ output on LPC transactions while the PC87591x is executing the V_{CC} Power-Up reset sequence (default)
1	**LPFTO (LPC Power Fail Turn Off $\overline{\text{KBRST}}$ and GA20).** Indicates the handling of $\overline{\text{KBRST}}$ and GA20 outputs when $\overline{\text{LPCPD}}$ is active. 0: Ignore $\overline{\text{LPCPD}}$ in handling these signals (default) 1: Force the two signals low while $\overline{\text{LPCPD}}$ is active or V_{DD} is low
2	**RTCAL (RTC Alarm).** Indicates that an RTC Alarm event occurred. This bit is set on the rising edge of the RTC Alarm output. It is cleared by writing 1 to it. Note that the ALARM event detection is edge triggered by the RTCAL bit; thus for a new event to be detected, first RTCAL bit and then Alarm Status bit in the RTC must be cleared. 0: No RTC Alarm is flagged (default) 1: RTC Alarm rising edge was detected
7-3	**Reserved.**

Host Configuration Base Address Low (HCFGBAL)

This is a byte-wide read/write register that holds the lower byte of the Host Configuration Registers base address. Bit 0 of this register is always forced to 0 to guarantee address alignment. This register is cleared on V_{CC} Power-Up reset.

Location: 00 FCC8$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Host Configuration Registers Base Address Low**							
Reset	0	0	0	0	0	0	0	0

Host Configuration Base Address High (HCFGBAH)

This is a byte-wide read/write register that holds the higher byte of the Host Configuration Registers base address. This register is cleared on V_{CC} Power-Up reset.

Location: 00 FCCA$_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Host Configuration Registers Base Address High**							
Reset	0	0	0	0	0	0	0	0

MSWC Interrupt Enable Register 2 (MSIEN2)

This is a byte-wide read/write register that holds enable bits for interrupt generation to the core through the MIWU (level high) for the respective bits in MSWCTL2 and MSWCTL3 registers. The interrupt may be cleared by clearing the status bit or masking the interrupt. On Warm reset, this register is cleared (00$_{16}$).

Location: 00 FCCC$_{16}$

Type: R/W

Host Controller Interface Modules (Continued)

Bit	7	6	5	4	3	2	1	0
Name	**EICFGPSO**	**EICFGPBM**	**EIACPIS5**	**EIACPIS4**	**EIACPIS3**	**EIACPIS2**	**EIACPIS1**	**EIRTCAL**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**EIRTCAL (Enable Interrupt on RTC Alarm).** Mask generation of interrupt to the core on setting of the RTC Alarm bit in MSWCTL1 register. 0: Interrupt disabled (default) 1: Generate a level high interrupt when the RTC Alarm bit is set
5-1	**EIACPIS5-1 (Enable Interrupt ACPI request for S5 through S1).** Mask generation of interrupt to the core on changes to ACPISi (i=5-1) bit in MSWCTL2 register. An interrupt enable for ACPIS0 is enabled when any of these bits is set. 0: Interrupt disabled (default) 1: Generate a level high interrupt on any change to the ACPISi bit
6	**EICFGPBM (Enable Interrupt SuperI/O Configuration Register D Power Button Mode).** Mask generation of interrupt to the core on changes to CFGPBM bit in MSWCTL2 register. 0: Interrupt disabled (default) 1: Generate a level high interrupt on any change to the CFGPBM bit
7	**EICFGPSO (Enable Interrupt SuperI/O Configuration Register D Power Supply Off).** Mask generation of interrupt to the core on changes to CFGPSO bit in MSWCTL2 register. 0: Interrupt disabled (default) 1: Generate a level high interrupt on any change to the CFGPBM bit

MSWC Host Event Status Register 0 (MSHES0)

This register holds information similar to that in WK_STS0 register. The same event that causes a WK_STS0 bit to be set sets the respective bit in MSHES0 register. Clearing bits is done for each of the status registers separately. This register is reset to 00_{16} on V_{CC} power-up or Host Domain Software reset. Writing 1 to a bit clears it to 0. Writing 0 has no effect. Bit 6 of this register behaves in a special way on set and clear, as described below.

Location: Offset 00 $FCCE_{16}$

Type: R/W1C

Bit	7	6	5	4	3	2	1	0
Name	**Module IRQ Event Status**	**Software Event Status**	**Reserved**		**$\overline{\text{RING}}$ Event Status**	**Reserved**	**$\overline{\text{RI2}}$ Event Status**	**$\overline{\text{RI1}}$ Event Status**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**$\overline{\text{RI1}}$ Event Status.** 0: Event not detected (default) 1: Event detected
1	**$\overline{\text{RI2}}$ Event Status.** 0: Event not detected (default) 1: Event detected
2	**Reserved.**

Host Controller Interface Modules (Continued)

Bit	Description
3	**$\overline{\text{RING}}$ Event Status.** $\overline{\text{RING}}$ event detection, according to the RING detection mode enabled. 0: Event not detected (default) 1: Event detected
5-4	**Reserved.**
6	**Software Event Status.** This bit indicates a host software event. It may operate in two modes depending on HSECM bit in MSWCTL1 register. When HSECM is cleared, this bit is set when bit 6 of WK_STS0 changes from 0 to 1. When HSECM is set, this bit is set on a write of 1 to WK_STS0 bit 1. This bit is cleared by writing 1 to it. 0: Event not active (default) 1: Event active
7	**Module IRQ Event Status.** This sticky bit shows the status of the module IRQ event detection. 0: Event not active (default) 1: Event active

MSWC Host Event Interrupt Enable Register (MSHEIE0)

This register is cleared to 00_{16} on Warm reset. It enables a core interrupt through the MIWU (level high) for the respective bit in the mobile MSHES0 register. The interrupt may be cleared by clearing the status bit or masking the interrupt.

Location: Offset 00 $FCD0_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Module IRQ Event Enable**	**Software Event Enable**	**Reserved**		**RING Event Enable**	**Reserved**	**$\overline{\text{RI2}}$ Event Enable**	**$\overline{\text{RI1}}$ Event Enable**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**$\overline{\text{RI1}}$ Event Enable.** 0: Disabled (default) 1: Enabled
1	**$\overline{\text{RI2}}$ Event Enable.** 0: Disabled (default) 1: Enabled
2	**Reserved.**
3	**RING Event Enable.** 0: Disabled (default) 1: Enabled
5-4	**Reserved.**
6	**Software Event Enable.** 0: Disabled (default) 1: Enabled
7	**Module IRQ Event Enable.** 0: Disabled (default) 1: Enabled

5.5.7 Usage Hints

PWUREQ Output Connection

The PWUREQ concentrates a set of wake-up and other power management events in the PC87591x. In typical use, this signal is connected to one of the chipset inputs that drive SCI event to the host.

RESET2 Events

When RESET2 is used to reset the host domain, some of the MSWC functions may need to be set to their default values when a falling edge of the RESET2 is detected. An interrupt routine triggered by this event may be used for this task. The following functions should be reset:

- GA20
- KBRST
- Host Registers: WK_EN0, WK_SMIEN0 and WK_IRWEN0;
  Use the “Core Access to Host-Controlled Modules” for this operation (see Section 5.4 on page 313).

6.0 Host-Controlled Modules and Host Interface

6.1 DEVICE ARCHITECTURE AND CONFIGURATION

The PC87591x Host-Controlled Functions comprises a collection of generic and proprietary functional blocks. Each functional block is described in a separate chapter in this document. However, some parameters in the implementation of the functional blocks may vary per function and/or device. This chapter describes the PC87591x structure and provides all logical device-specific information, including special implementation of generic blocks, system interface and device configuration.

The PC87591x Host-Controlled Functions consist of seven logical devices (involving six modules), the host interface and a central set of configuration registers, all built around a central internal bus. The internal bus is similar to an 8-bit ISA bus protocol. Figure 108 illustrates the blocks and their interconnection.

The LPC Bus Interface serves as a bridge between the external LPC interface and the internal bus. It supports the following operations, as defined in Intel’s *LPC Interface Specification, Revision 1.0*:

- 8-bit I/O read
- 8-bit I/O write
- 8-bit Memory read
- 8-bit Memory write
- 8-bit FWH read
- 8-bit FWH write

The Configuration and Control register set supports ACPI-compliant PnP configuration. The configuration registers are structured as a subset of the Plug and Play Standard registers, defined in Appendix A of the *Plug and Play ISA Specification, Revision 1.0a* by Intel and Microsoft, and are similar to those used in National SuperI/O devices. All system resources assigned to the functional blocks (I/O address space, and IRQ lines) are configured in and managed by this register set. In addition, some function-specific parameters are configurable through the configuration registers and distributed to the functional blocks through special control signals.

6.1.1 Configuration Structure and Access

The configuration structure comprises a set of banked registers that are accessed via a pair of specialized registers.

The Index-Data Register Pair

Access to the Host-Controlled Functions configuration registers is via an Index-Data register pair, using two system I/O byte locations. The base address of this register pair is determined during V_{CC} Power-Up reset, according to the state of the hardware strapping option on the BADDR1-0 pins. Table 47 shows the selected base addresses as a function of BADDR1-0 (see Section 2.2.11 on page 50).

Table 47. BADDR1-0 Strapping Options

BADDR1-0	I/O Address	
	Index Register	Data Register
0 0	$2E_{16}$	$2F_{16}$
0 1	$4E_{16}$	$4F_{16}$
1 0[1]	(HCFGBAH,HCFGBAL)	(HCFGBAH,HCFGBAL)+1
1 1	Reserved	

[1] See “Host Configuration Address Selection” on page 322 for more details about this option.

The Index register is an 8-bit read/write register located at the selected base address (Base+0). It is used as a pointer to the configuration register file and holds the index of the configuration register that is currently accessible via the Data register. Reading the Index register returns the last value written to it (or a default of 00_{16} after Host Domain reset).

The Data register is an 8-bit register located at the selected base address (Base+1) used as a data path to any configuration register. Accessing the Data register actually accesses the configuration register that the Index register is currently pointed to.

6.0 Host-Controlled Modules and Host Interface (Continued)

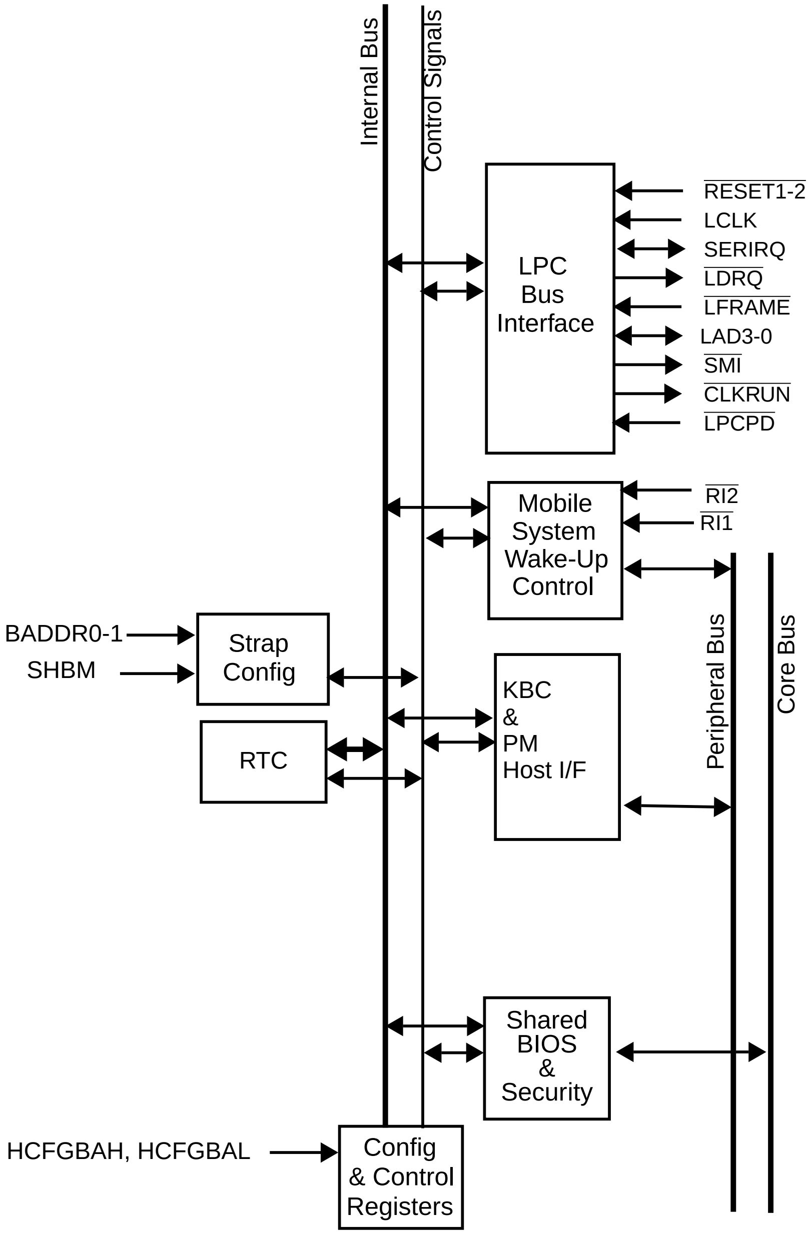


Figure 108. Host-Controlled Domain Detailed Block Diagram

Banked Logical Device Registers Structure

Each functional block is associated with a Logical Device Number (LDN). The configuration registers are grouped into banks, where each bank holds the standard configuration registers of the corresponding logical device. Table 48 shows the LDN values of the PC87591x functional blocks. Any value not listed is reserved.

Figure 109 shows the structure of the standard configuration register file. The Host-Controlled Functions control and configuration registers are not banked and are accessed by the Index-Data register pair only, as described above. However, the device control and device configuration registers are duplicated over seven banks for the seven logical devices. Therefore, two-dimensional indexing is used to access a specific register in a specific bank: the LDN register selects the bank (or logical device) and the Index register selects the register within the bank. Accessing the Data register while the Index register holds a value of 30_{16} or higher physically accesses the logical device configuration registers currently pointed to by the Index register, within the logical device currently selected by the LDN register.

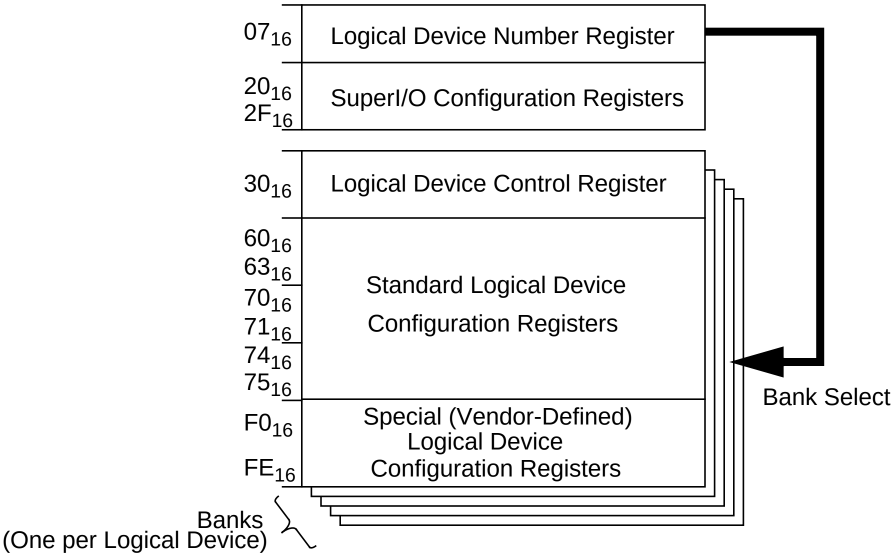

07_{16}
Logical Device Number Register
20_{16}
$2F_{16}$
SuperI/O Configuration Registers
30_{16}
Logical Device Control Register
60_{16}
63_{16}
70_{16}
71_{16}
74_{16}
75_{16}
Standard Logical Device Configuration Registers
$F0_{16}$
FE_{16}
Special (Vendor-Defined) Logical Device Configuration Registers
Bank Select
Banks (One per Logical Device)

Figure 109. Structure of Standard Configuration Register File

Table 48. Logical Device Number (LDN) Assignments

LDN	Functional Block
04_{16}	Mobile System Wake-Up Control (MSWC)
05_{16}	Keyboard and Mouse Controller (KBC) - Mouse Interface
06_{16}	Keyboard and Mouse Controller (KBC) - Keyboard Interface
$0F_{16}$	Shared BIOS and Security
10_{16}	Real Time Clock
11_{16}	Power Management I/F Channel 1
12_{16}	Power Management I/F Channel 2

Write accesses to unimplemented registers (i.e., accessing the Data register while the Index register points to a non-existing register) are ignored; reads return 00_{16} for all addresses except 74_{16} and 75_{16} (DMA configuration registers), which return 04_{16} (indicating that no DMA channel is active). The configuration registers are accessible immediately after Host Domain reset.

Standard Logical Device Configuration Register Definitions

In the registers below, any undefined bit is reserved. Unless otherwise noted, the following definitions also hold true:

- All registers are read/write.
- All reserved bits return 0 on reads except where noted. To prevent unpredictable results, do not modify these bits. Use read-modify-write to prevent the values of reserved bits from being changed during write.
- Write-only registers should not use read-modify-write during updates.

6.0 Host-Controlled Modules and Host Interface (Continued)

Table 49. Standard Control Registers

Index	Register Name	Description
07_{16}	Logical Device Number	This register selects the current logical device. See Table 48 for valid numbers. All other values are reserved.
20_{16} - $2F_{16}$	SuperI/O Configuration	SuperI/O configuration registers and ID registers

Table 50. Logical Device Activate Register

Index	Register Name	Description
30_{16}	Activate	Bits 7-1: Reserved Bit 0: Logical device activation control 0: Disabled 1: Enabled

Table 51. I/O Space Configuration Registers

Index	Register Name	Description
60_{16}	I/O Port Base Address Bits (15-8) Descriptor 0	Indicates selected I/O lower limit address bits 15-8 for I/O Descriptor 0.
61_{16}	I/O Port Base Address Bits (7-0) Descriptor 0	Indicates selected I/O lower limit address bits 7-0 for I/O Descriptor 0.
62_{16}	I/O Port Base Address Bits (15-8) Descriptor 1	Indicates selected I/O group 2 lower limit address bits 15-8 for I/O Descriptor 1.
63_{16}	I/O Port Base Address Bits (7-0) Descriptor 1	Indicates selected I/O group 2 lower limit address bits 7-0 for I/O Descriptor 1.

6.0 Host-Controlled Modules and Host Interface (Continued)

Table 52. Interrupt Configuration Registers

Index	Register Name	Description
70_{16}	Interrupt Number and Wake-Up on IRQ Enable	Indicates selected interrupt number. Bits 7-5: Reserved. Bit 4: Enables wake-up on the IRQ of the logical device. When enabled, IRQ assertion triggers a wake-up event. 0: Disabled (default) 1: Enabled Bits 3-0: select the interrupt number. A value of 1 selects IRQ1, a value of 2 selects IRQ2, etc. (up to IRQ15). A value of 0 disables this interrupt.
71_{16}	Interrupt Request Type Select	Indicates the type and level of the interrupt request number selected in the previous register. Bits 7-2: Reserved. Bit 1: Polarity of interrupt request selected in previous register 0: Low polarity 1: High polarity Bit 0: Type of interrupt request selected in previous register 0: Edge 1: Level

Table 53. DMA Configuration Registers

Index	Register Name	Description
74_{16}	DMA Channel Select 0	Indicates selected DMA channel for DMA 0 of the logical device (0: The first DMA channel if more than one DMA channel is used). Bits 7-3: Reserved. Bits 2-0: Select the DMA channel for DMA 0. The valid choices are 0-3, where: ● A value of 0 selects DMA channel 0, 1 selects channel 1, etc. ● A value of 4 indicates that no DMA channel is active. ● The values 5-7 are reserved.
75_{16}	DMA Channel Select 1	Indicates selected DMA channel for DMA 1 of the logical device (1: The second DMA channel if more than one DMA channel is used). Bits 7-3: Reserved. Bits 2-0: Select the DMA channel for DMA 1. The valid choices are 0-3, where: ● A value of 0 selects DMA channel 0, 1 selects channel 1, etc. ● A value of 4 indicates that no DMA channel is active. ● The values 5-7 are reserved.

Table 54. Special Logical Device Configuration Registers

Index	Register Name	Description
$F0_{16}$-FE_{16}	Logical Device Configuration	Special (vendor-defined) configuration options

6.1.2 Standard Configuration Registers

	Index	Register Name
SuperI/O Control and Configuration Registers	07_{16}	Logical Device Number
	20_{16}	SuperI/O ID
	21_{16}	SuperI/O Configuration 1
	25_{16}	SuperI/O Configuration 5
	26_{16}	SuperI/O Configuration 6
	27_{16}	SuperI/O Revision ID
	28_{16}	SuperI/O Configuration 8
	29_{16}	SuperI/O Configuration 9
	$2D_{16}$	SuperI/O Configuration D
	$2B_{16}$ - $2E_{16}$	Reserved exclusively for National use
Logical Device Control and Configuration Registers - one per Logical Device (some are optional)	30_{16}	Logical Device Control (Activate)
	60_{16}	I/O Base Address Descriptor 0 Bits 15-8
	61_{16}	I/O Base Address Descriptor 0 Bits 7-0
	62_{16}	I/O Base Address Descriptor 1 Bits 15-8
	63_{16}	I/O Base Address Descriptor 1 Bits 7-0
	70_{16}	Interrupt Number and Wake-Up on IRQ Enable
	71_{16}	IRQ Type Select
	74_{16}	DMA Channel Select 0
	75_{16}	DMA Channel Select 1
	$F0_{16}$ - $F9_{16}$	Device Specific Logical Device Configuration 1 to 10

Figure 110. Configuration Register Map

SuperI/O Control and Configuration Registers

The SuperI/O configuration registers at indexes 20_{16} and 27_{16} are mainly used for part identification, global power management and the selection of pin multiplexing options. For details, see Section 6.1.8 on page 343.

Logical Device Control and Configuration Registers

A subset of these registers is implemented for each logical device. See the functional block descriptions in Chapter 5 on page 277 and Chapter 6 on page 335.

Control

The only implemented control register for each logical device is the Activate register at index 30_{16}. Bit 0 of Activate register controls the activation of the associated functional block. Activation enables access to the functional block's registers and attaches the functional block's system resources (e.g., address space and interrupts), which are unused as long as the block is not activated. Other effects may apply on a function-specific basis (such as clock enable and active pinout signaling).

6.0 Host-Controlled Modules and Host Interface (Continued)

Standard Configuration

The standard configuration registers manage the PnP resource allocation to the functional blocks. The I/O port base address (descriptor 0) is a pair of registers at index $60\text{-}61_{16}$, holding the first 16-bit base address for the register set of the functional block. An optional 16-bit second base-address (descriptor 1) at index $62\text{-}63_{16}$ is used for logical devices with more than one continuous register set. Interrupt Number and Wake-Up on IRQ Enable (index 70_{16}) and IRQ Type Select (index 71_{16}) registers allocate an IRQ number to the module's interrupt and control the interrupt type and polarity. DMA Channel Select 0 (index 74_{16}) allocates a DMA channel to the block, where applicable. DMA Channel Select 1 (index 75_{16}) allocates a second DMA channel, where applicable.

Special Configuration

The vendor-defined registers, starting at index $F0_{16}$ - $F9_{16}$, control function-specific parameters such as operation modes, power saving modes, clock rate selection and non-standard extensions to generic functions.

6.1.3 Default Configuration Setup

The default configuration setup of the PC87591x Host-Controlled functions is set according to the following reset types (see Section 3.2 on page 65):

- **V_{PP} Power-Up Reset**
  Resets V_{PP}-retained SuperI/O functions, such as the RTC and the MSWC registers, whose values are retained by V_{PP}.
- **V_{CC} Power-Up Reset**
  Resets the MSWC registers whose values are retained by V_{CC} only.
- **Host Domain Hardware Reset**
  - — Resets all SuperI/O logical devices, with the exception of the Mobile System Wake-Up Control (MSWC) and the RTC registers retained by V_{PP} or V_{CC}.
  - — Resets all SuperI/O configuration registers.
- **Host Domain Software Reset**
  - — Resets all SuperI/O logical devices, except the MSWC and the RTC registers retained by V_{PP} or V_{CC}.
  - — Resets most bits in the SuperI/O configuration registers. This reset does not affect register bits that are locked for write access (see "SuperI/O Configuration 6 Register (SIOCF6)" on page 345 and Table 47 on page 335).

If a Host Domain Hardware reset occurs, the PC87591x wakes up with the following default SuperI/O configuration setup:

- — The configuration base address is according to the BADDR strap pin value, as shown in Table 47 on page 335.
- — All logical devices are disabled, with the exception of the MSWC and shared memory, which remain functional but whose registers cannot be accessed.

If a Host Domain reset occurs (either Software or Hardware), the PC87591x wakes up with the following default SuperI/O configuration setup:

- — The legacy devices are assigned with their legacy system resource allocation.
- — The National proprietary functions are not assigned with any default resources, and the default values of their base addresses are all 00_{16}.

6.1.4 Address Decoding

A full 16-bit address decoding is applied when accessing the configuration I/O space as well as the registers of the functional blocks. However, the number of configurable bits in the base address registers varies for each logical device.

The lower 0, 1, 2, 3, 4 or 5 address bits are decoded within the functional block to determine the offset of the accessed register within the logical device's I/O range of 1, 2, 4, 8, 16 or 32 bytes, respectively. The rest of the bits are matched with the base address register to decode the entire I/O range allocated to the logical device. Therefore, the lower bits of the base address register are forced to 0 (read only), and the base address is forced to be 1, 2, 4, 8, 16 or 32 byte-aligned, according to the size of the I/O range.

The base address of the KBC, PM channel 1 and PM channel 2 are limited to the I/O address range of 0000_{16} to $07FX_{16}$ only (bits 11-15 are forced to 0). The addresses of other devices are configurable within the full 16-bit address range (up to $FFFF_{16}$).

In some special cases, other address bits are used for internal decoding (such as bit 2 in the KBC). The KBC has two I/O descriptors with some implied dependency between them. For more details, see the description of the base address register for each logical device.

The Shared Memory and Security module serves as a bridge from the LPC to the on-chip flash and off-chip expansion memory. For module control and security function registers, the 16-bit base address is applied through the configuration address space. To access the registers, the lower four address bits are decoded within the Shared Memory module. The address ranges in the LPC memory space and the FWH memory space, which are bridged to the shared memory, are defined in the

SuperI/O configuration section for the shared memory bridge. The number of address bits used for this decoding varies according to the specified zones and their sizes. See “Memory Range Programing” on page 350 and “Shared Memory Configuration Register” on page 351 for details about the address range specifications.

6.1.5 Interrupt Serializer

The Interrupt Serializer translates internal IRQ sources into serial interrupt request data transmitted over the SERIRQ bus. Figure 111 shows the interrupt serialization mechanism.

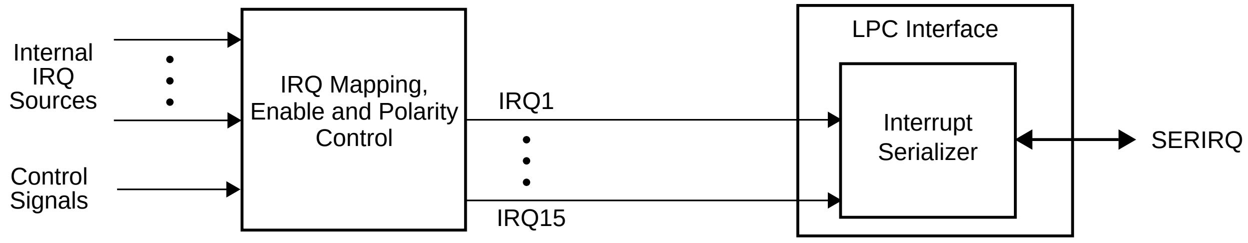


Figure 111. Interrupt Serialization Mechanism

The internal IRQ signals are fed into an IRQ Mapping and Polarity Control block. This block maps them to their associated IRQ slots. The IRQs are then fed into the Interrupt Serializer, where they are translated into serial data and transmitted over the SERIRQ bus.

6.1.6 Protection

The PC87591x provides features to protect the Personal Computer (PC) at software levels. The PC can be locked to protect configuration bits and to prevent alteration of the device hardware configuration and several types of configuration settings.

The use of all protection mechanisms is optional.

6.1.7 LPC Interface

LPC Transactions Supported

The PC87591x LPC interface responds to the following LPC transactions as part of the standard Host Bus interface:

- I/O read cycles
- I/O write cycles

In addition, the Shared Memory module uses the following transactions:

- 8-bit memory read and write
- 8-bit FWH read and write

LPC transactions conform with Intel’s *LPC Interface Specification, Revision 1.0.*

The LPC- FWH read and write protocols are similar to memory read and write cycles. The specifications of these cycles are listed below. The Address, Data, TAR and SYNC cycles are as specified for LPC memory read and write cycles. The START and ID fields are similar to the equivalent cycle in LPC memory read and write transactions but differ in the data placed on the LAD signals (see details in the cycle description).

Note: The PC87591x supports FWH transactions from LPC controllers that accept wait-sync and long wait-sync cycles. With other LPC controllers, use the indirect write mechanism in the Shared Memory module to perform write operations.

FWH Read Cycle

1. START: 1101_{16} (0xD).
2. ID field: FWH ID nibble (compared with bits 7-4 of shared memory; see “Shared Memory Configuration Register” on page 351).
3. Address: Eight address nibbles, MS nibble first; see usage below).
4. TAR (two cycles).
5. SYNC.
6. DATA: Two data nibbles, LS nibble first (D3-D0, D7-D4).
7. TAR (two cycles).

FWH Write Cycle:

1. START: 1110_{16} (0xE).
2. ID field: FWH ID nibble (compared with bits 7-4 of shared memory; see “Shared Memory Configuration Register” on page 351).
3. Address: Eight address nibbles MS nibble first (see usage below).
4. DATA: Two data nibbles, LS nibble first (D3-D0, D7-D4).
5. TAR (two cycles).
6. SYNC.
7. TAR (two cycles).

The ID field is compared with bits 7-4 of shared memory; see “Shared Memory Configuration Register” on page 351. If the two match, the PC87591x continues handling the transaction; if they do not match, the current LPC-FWH transaction is ignored.

LPC-FWH Address translation: The address field in the LPC-FWH transaction is constructed of eight nibbles. The first seven correspond to the first LS seven address nibbles (A27-A0) as follows: The first nibble that appears corresponds to addresses A27-A24, the second to A23-A20, until the seventh incoming nibble, which corresponds to addresses A3-A0. Incoming nibble number eight is ignored. The MS bits of the 32-bit addresses are ‘1111’ (A31 - A28).

Core Interrupt

Whenever there is an LPC or FWH transaction that is responded to by any of the PC87591x logical devices, a positive pulse is generated on the Host Access Wake-Up input of the MIWU module. This interrupt may be used to wake up the core for handling any host activity.

$\overline{\text{CLKRUN}}$ Functionality

The PC87591x supports the $\overline{\text{CLKRUN}}$ I/O signal, the use of which is highly recommended in portable systems. This signal is implemented according to the specification in *PCI Mobile Design Guide, Revision 1.1*, December 18, 1998. The PC87591x supports operation with both a slow and stopped clock in ACPI state S0 (the system is active but is not being accessed). The PC87591x drives the $\overline{\text{CLKRUN}}$ low to force the LPC bus clock into full speed operation when an IRQ is pending internally and waiting to be sent through the serial IRQ.

$\overline{\text{LPCPD}}$ Functionality

The PC87591x supports the $\overline{\text{LPCPD}}$ input. This signal is used when the V_{DD} chip supply is not shared by all residents of the LPC bus. The $\overline{\text{LPCPD}}$ signal conforms with Intel’s *LPC Interface Specification, Revision 1.0*. Note that if the PC87591x power supply exists while $\overline{\text{LPCPD}}$ is active, it is not mandatory to reset the PC87591x when $\overline{\text{LPCPD}}$ is de-asserted.

6.1.8 SuperI/O Configuration Registers

This section describes the SuperI/O configuration and ID registers (those registers with first level indexes in the range of 20_{16} - $2E_{16}$). See Table 55 for a summary and directory of these registers.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Table 55. SuperI/O Configuration Registers

Index	Mnemonic	Register Name	Power Well	Type
20_{16}	SID	SuperI/O ID	V_{DD}	RO
21_{16}	SIOCF1	SuperI/O Configuration 1	V_{DD}	Varies per bit
22_{16} - 24_{16}	Reserved exclusively for National use			
25_{16}	SIOCF5	SuperI/O Configuration 5	V_{DD}	R/W
26_{16}	SIOCF6	SuperI/O Configuration 6	V_{DD}	R/W
27_{16}	SRID	SuperI/O Revision ID	V_{DD}	RO
28_{16}	SIOCF8	SuperI/O Configuration 8	V_{DD}	R/W
29_{16}	SIOCF9	SuperI/O Configuration 9	V_{DD}	R/W

6.0 Host-Controlled Modules and Host Interface (Continued)

Table 55. SuperI/O Configuration Registers (Continued)

Index	Mnemonic	Register Name	Power Well	Type
$2A_{16}$ - $2C_{16}$	Reserved exclusively for National use			
$2D_{16}$	SIOCFD	SuperI/O Configuration D	V_{PP}	R/W
$2E_{16}$	Reserved exclusively for National use			

SuperI/O ID Register (SID)

This register contains the identity number of the chip. The PC87591x is identified by the value EC_{16}.

Location: Index 20_{16}

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Family ID**							
Reset	EC_{16}							

Bit	Description
7-0	**Family ID.** These bits identify a family of devices with similar functionality but with different options implemented.

SuperI/O Configuration 1 Register (SIOCF1)

Location: Index 21_{16}

Type: Varies per bit

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**		**Number of DMA Wait States**		**Number of I/O Wait States**		**Software Reset**	**SuperI/O Devices Enable**
Reset	0	0	0	1	0	0	0	1

Bit	Type	Description
0	R/W	**SuperI/O Devices Enable.** Controls the function enable of all the PC87591x SuperI/O logical devices, except shared memory and Mobile System Wake-Up Control (MSWC). This bit enables the simultaneous disabling of these modules using a write to a single bit. 0: All SuperI/O logical devices in the PC87591x are disabled, except MSWC and shared memory 1: Each SuperI/O logical device is enabled according to its Activate register (Index 30_{16}) (default)
1	WO	**Software Reset.** Read always returns 0. 0: Ignored (default) 1: Triggers the Host Domain Software Reset event, which resets the logical devices (see Section 6.1.3 on page 341)
3-2	R/W	**Number of I/O Wait States.** **Bits** **3 2 Number** 0 0: 0 (default) 0 1: 2 1 0: 6 1 1: 12

Bit	Type	Description
5-4	R/W	**Number of DMA Wait States.** **Bits** **5 4 Number** 0 0: Reserved 0 1: 2 (default) 1 0: 6 1 1: 12
7-6		**Reserved.**

SuperI/O Configuration 5 Register (SIOCF5)

Location: Index 25_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**			**$\overline{\text{SMI}}$ to IRQ2 Enable**	**Reserved**			
Reset	0	0	0	0	0	0	0	0

Bit	Type	Description
3-0		**Reserved.**
4	R/W	**$\overline{\text{SMI}}$ to IRQ2 Enable.** Enables using slot number 2 in the serial IRQ protocol as an SMI interrupt in parallel to or instead of using the dedicated pin. 0: Disabled (default) 1: Enabled
7-5		**Reserved.**

SuperI/O Configuration 6 Register (SIOCF6)

Write access to this register can be inhibited by setting bit 7 of this register. Activation of each logical device (bits 0-4) is also affected by bit 0 of the logical device Activate register, index 30_{16}, and bit 0 of SIOCF1 register.

Location: Index 26_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**SIOCF6 Software Lock**	**General-Purpose Scratch**		**RTC Disabled**	**Reserved**			
Reset	0	0	0	0	0	0	0	0

Bit	Description
3-0	**Reserved.**

6.0 Host-Controlled Modules and Host Interface (Continued)

Bit	Description
4	**RTC Disabled.** 0: Enabled (default) 1: Disabled
6-5	**General-Purpose Scratch.**
7	**SIOCF6 Software Lock.** When set to 1 by software, it and other bits in this register can be cleared only by Host Domain Hardware reset. 0: Write access to bits 0-6 of this register enabled (default) 1: Bits 6-0 of this register are read only.

SuperI/O Revision ID Register (SRID)

This register contains the ID number of the specific family member (Chip ID) and the chip revision number (Chip Rev). The Chip Rev is incremented on each revision.

Location: Index 27_{16}

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Chip ID**			**Chip Rev**				
Reset	See values in field description			X	X	X	X	X

Bit	Description
4-0	**Chip Rev.** Identifies the device revision.
7-5	**Chip ID.** Identifies a specific device of the PC87591x family. **Bits** **7 6 5 Device** 0 0 0: PC87591E 0 0 1: PC87591S 1 0 0: PC87591 L Other: Reserved

SuperI/O Configuration 8 Register (SIOCF8)

Location: Index 28_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Reserved.**

6.0 Host-Controlled Modules and Host Interface (Continued)

SuperI/O Configuration 9 Register (SIOCF9)

Location: Index 29_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**Reserved**	
Reset	0	0	0	0	0	0	0	1

Bit	Description
7-0	**Reserved.**

SuperI/O Configuration D Register (SIOCFD)

This is a battery-backed register.

Location: Index $2D_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**						**Power Supply Off**	**Power Button Mode**
Reset	0	0	0	0	0	0	0	0

Bit	Description
0	**Power Button Mode.** This is a R/W bit. The value of this bit is available to the core through a register in the MSWC; see Section 5.5 on page 318. 0: Legacy (default at V_{CC} Power-Up reset) 1: ACPI
1	**Power Supply Off.** This is a R/W bit. It always returns 0 when read. When using Legacy mode (bit 0 is set to 0) and setting this bit to 1, this bit indicates to the core that the host requests shutting down the power. The value of this bit is available to the core by reading a register in the MSWC; see Section 5.5 on page 318. 0: No action (default at V_{PP} Power-Up reset) 1: Indicates power shut down to the core in Legacy mode
7-2	**Reserved.**

6.0 Host-Controlled Modules and Host Interface (Continued)

6.1.9 Mobile System Wake-Up Control (MSWC) Configuration

Logical Device 4 (MSWC) Configuration

Table 56 lists the configuration registers that affect the MSWC. See Section 6.1.2 on page 340 for a detailed description of these registers.

Table 56. Mobile System Wake-Up Control (MSWC) Configuration Registers

Index	Configuration Register or Action	Type	Reset
30_{16}	Activate. When bit 0 is cleared, the registers of this logical device are not accessible.[1]	R/W	00_{16}
60_{16}	Base Address MSB register.	R/W	00_{16}
61_{16}	Base Address LSB register. Bits 4-0 (for A4-0) are read only, '00000'.	R/W	00_{16}
70_{16}	Interrupt Number.	R/W	00_{16}
71_{16}	Interrupt Type. Bit 1 is read/write. Other bits are read only.	R/W	03_{16}
74_{16}	Report no DMA assignment.	RO	04_{16}
75_{16}	Report no DMA assignment.	RO	04_{16}

[1] The logical device registers are maintained, and all wake-up detection mechanisms are functional.

6.1.10 Keyboard and Mouse Controller (KBC) Configuration

Logical Devices 5 and 6 (Mouse and Keyboard) Configuration

Tables 57 and 58 list the configuration registers that affect the Mouse and the Keyboard respectively. Only the last register ($F0_{16}$) is described here. See "Standard Logical Device Configuration Register Definitions" on page 337 and Section 6.1.2 on page 340 for descriptions of the other configuration registers.

Usage Hints: It is recommended to set the type of interrupt request as level and its level of interrupt as high.

Table 57. Mouse Configuration Registers

Index	Mouse Configuration Register or Action	Type	Reset
30_{16}	Activate. See also bit 0 of the SIOCF1. When the Mouse of the KBC is inactive, the IRQ selected by the Mouse Interrupt Number and Wake-Up on IRQ Enable register (index 70_{16}) is not asserted. This register has no effect on host KBC commands handling the PS/2 Mouse.	R/W	00_{16}
70_{16}	Mouse Interrupt Number and Wake-Up on IRQ Enable register.	R/W	$0C_{16}$
71_{16}	Mouse Interrupt Type. Bit 1 is read/write; other bits are read only.	R/W	03_{16}
74_{16}	Report no DMA assignment.	RO	04_{16}
75_{16}	Report no DMA assignment.	RO	04_{16}

6.0 Host-Controlled Modules and Host Interface (Continued)

Table 58. Keyboard Configuration Registers

Index	Keyboard Configuration Register or Action	Type	Reset
30_{16}	Activate. See also bit 0 of the SIOCF1.	R/W	00_{16}
60_{16}	Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’.	R/W	00_{16}
61_{16}	Base Address LSB register. Bits 2-0 are read only, ‘000’.	R/W	60_{16}
62_{16}	Command Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’.	R/W	00_{16}
63_{16}	Command Base Address LSB. Bits 2-0 are read only, ‘100’.	R/W	64_{16}
70_{16}	KBD Interrupt Number and Wake-Up on IRQ Enable register.	R/W	01_{16}
71_{16}	KBD Interrupt Type. Bit 1 is read/write; other bits are read only.	R/W	03_{16}
74_{16}	Report no DMA assignment.	RO	04_{16}
75_{16}	Report no DMA assignment.	RO	04_{16}

6.1.11 Shared Memory Configuration

Logical Device 15 ($0F_{16}$) (Shared Memory) Configuration

Table 59 lists the configuration registers that affect the shared memory functional block. The shared memory base address registers point to the shared memory registers described in Section 5.3 on page 297. The memory space to which the shared memory responds is defined by the configuration registers in the following sections. See “Standard Logical Device Configuration Register Definitions” on page 337 and Section 6.1.2 on page 340 for a detailed description of the other configuration registers.

Table 59. Shared Memory Configuration Registers

Index	Configuration Register or Action	Type	Reset
30_{16}	Activate. When bit 0 is cleared, the registers of this logical device are not accessible.	R/W	00_{16}
60_{16}	Base Address MSB register.	R/W	00_{16}
61_{16}	Base Address LSB register. Bits 3-0 (for A3-A0) are read only, ‘0000’.	R/W	00_{16}
70_{16}	No interrupt assignment.	RO	00_{16}
71_{16}	No interrupt assignment.	RO	00_{16}
74_{16}	Report no DMA assignment.	RO	04_{16}
75_{16}	Report no DMA assignment.	RO	04_{16}
$F4_{16}$	Shared Memory Configuration register.	R/W	00_{16} or 09_{16}, depending on the value of the SHBM strap input
$F5_{16}$	Shared Memory Base Address High Byte register.	R/W	00_{16}
$F6_{16}$	Shared Memory Base Address Low Byte register.	R/W	00_{16}
$F7_{16}$	Shared Memory Size Configuration register.	R/W	00_{16}

6.0 Host-Controlled Modules and Host Interface (Continued)

Memory Range Programing

LPC memory transactions and/or LPC-FWH transactions can be forwarded to the PC87591x shared memory. The Shared Memory Configuration register defines the transaction type and address range to which the PC87591x responds. The SHBM strap inputs affect the default settings of the Shared Memory Configuration register to enable boot process from shared memories. Two memory areas may be individually enabled: a user-defined zone and BIOS memory (either BIOS-LPC and/or BIOS-FWH spaces).

To enable BIOS support, set the SHBM strap inputs to select any of the BIOS modes (see Section 2.3 on page 53 for details). The PC87591x responds to LPC memory read and write transactions to/from the BIOS address spaces, shown in Table 60, as long as BIOS LPC Enable (bit 0) of the Shared Memory Configuration register is set.

Table 60. BIOS-LPC Memory Space Definition

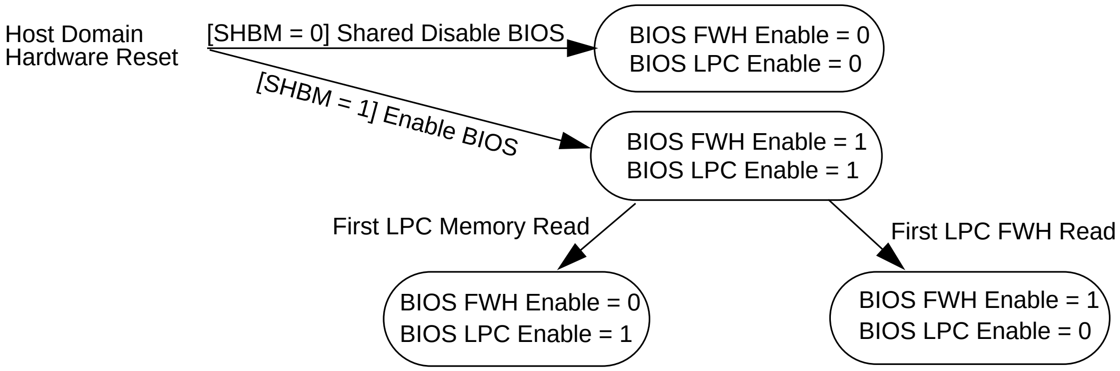
Memory Address Range	Description
000E 0000_{16} - 000E $FFFF_{16}$	Extended BIOS range (Legacy); only when Extended BIOS Enable bit in Shared Memory Range Configuration register is set.
000F 0000_{16} - 000F $FFFF_{16}$	BIOS Range (Legacy)
FFE0 0000_{16} - FFFF $FFFF_{16}$	386 mode BIOS range; this is the upper 2 Mbytes of the memory space.

The PC87591x responds to LPC-FWH read and write transactions to/from the high memory address range (’386’ mode BIOS range), shown in Table 60, as long as BIOS FWH Enable (bit 3) of the Shared Memory Configuration register is set.

Table 61. BIOS-FWH Memory Space Definition

Memory Address Range	Description
FFE0 0000_{16} - FFFF $FFFF_{16}$	386 mode BIOS range; this is the upper 2 Mbytes of the memory space. The PC87591x uses the first 21 address lines and ID field to identify FWH access to the shared memory.

On host domain hardware reset in BIOS mode, the BIOS LPC Enable bit is set and the BIOS FWH Enable bit is set. The PC87591x automatically detects the type of host boot protocol in use, via the first completed BIOS read operation after host domain hardware reset. If the first read is an LPC memory read, the BIOS FWH Enable bit is cleared. If the first read is an LPC-FWH read, the BIOS LPC Enable bit is cleared. Any other LPC or LPC-FWH transactions are ignored. The bits are cleared only by the first read operation, allowing software to enable responding to these address ranges by setting the bit. Figure 112 illustrates this behavior.



Note: Only hardware-controlled transitions are shown;
other transitions are possible via software writes to the bits.

Figure 112. BIOS Mapping Enable Scheme

The user-defined shared memory enables sharing the memory without using it as a shared BIOS memory. The memory base address in the host address space and the size of the shared memory are defined in the Shared Memory Base and Shared Memory Size registers. Address bits above the block size are ignored and are internally replaced with 1s for the purpose of address translation (e.g., for a 2 Mbyte block size, A21 through A31 are replaced with 1s).

6.0 Host-Controlled Modules and Host Interface (Continued)

Shared Memory Configuration Register

This register is reset on host domain hardware reset to 00_{16} or 09_{16}, depending on the value of the SHBM strap input.

Location: Index $F4_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**BIOS FWH ID**				**BIOS FWH Enable**	User-Defined Memory Space Enable	**BIOS Extended Space Enable**	**BIOS LPC Enable**
Reset	0	0	0	0	Strap	0	0	Strap

Bit	Description
0	**BIOS LPC Enable.** Enables the PC87591x to respond to LPC memory accesses to the BIOS-LPC space. The reset value of this register is defined by the SHBM configuration inputs. The value of this bit is updated later, based on the detected host BIOS scheme; see “Memory Range Programing” on page 350 for details. 0: Disabled (default when SHBM disable BIOS configuration) 1: Enabled (default when SHBM enable BIOS configuration)
1	**BIOS Extended Space Enable.** Expands the BIOS address space to which the PC87591x responds to include the Extended BIOS address range. 0: Disabled (default) 1: Enabled
2	**User-Defined Memory Space Enable.** When set, enables the PC87591x to respond to LPC memory read and write accesses in the user-defined memory area range. The base address and size of the user-defined range are specified by the Shared Memory Base Address High and Low Byte registers and the Shared Memory Size Configuration register. 0: Disabled (default) 1: Enabled
3	**BIOS FWH Enable.** When set, enables PC87591x response to LPC-FWH transactions to the BIOS-FWH space. The reset value of this register is defined by the XCNF(2-0) configuration inputs. The value of this bit is later updated based on the detected host BIOS scheme; see “Memory Range Programing” on page 350 for details. 0: Disabled (default when SHBM disable BIOS configuration) 1: Enabled (default when SHBM enable BIOS configuration)
7-4	**BIOS FWH ID.** These four bits correspond to the identification nibble, which is part of a FWH transaction (see Section 6.1.7 for details).

Shared Memory Base Address High Byte Register

This register describes the high byte for the user-defined memory zone mapped to the shared memory (decoded as bits 31 to 24 of the 32-bit address range, bits 15-0 are 0). This register is reset to 00_{16} on Host Domain Hardware reset.

Location: Index $F5_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**User-Defined Memory Zone Address High**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**User-Defined Memory Zone Address High.** Defines the higher eight bits of the user-defined memory block base address. The base address should be aligned on the selected block size.

6.0 Host-Controlled Modules and Host Interface (Continued)

Shared Memory Base Address Low Byte Register

This register describes the low byte for the user-defined memory zone mapped to the shared memory (decoded as bits 23 to 16 of the 32-bit address range, bits 15-0 are 0). This register is reset to 00_{16} on Host Domain Hardware reset.

Location: Index $F6_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**User-Defined Memory Zone Address Low**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**User-Defined Memory Zone Address Low.** Defines the lower eight bits of the user-defined memory block base address. The base address should be aligned on the selected block size.

Shared Memory Size Configuration Register

This register defines the size of the user-defined memory zone mapped to the shared memory. This register is reset to 00_{16} on host domain hardware reset.

Location: Index $F7_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	Reserved				User-Defined Memory Zone Size			
Reset	0	0	0	0	0	0	0	0

Bit	Description
3-0	**User-Defined Memory Zone Size.** Defines the size, in bytes, of the zone window. The size is defined as an exponent of two, using the equation: NumOfBytes = 2^n (where, n = User-Defined Memory Zone size+16). The zone must always be aligned to the window size (i.e., for a 128 Kbyte window, the 17 LSBs of the address should be zero). **Bits** **3 2 1 0 Size (Bytes)** 0 0 0 0 64K (default) . . . 0 1 0 1 2M Other Reserved
7-4	**Reserved.**

6.1.12 Real Time Clock (RTC) Configuration

Logical Device 16 (10_{16}) RTC Configuration

Table 62 lists the configuration registers that affect the RTC. See Section 6.1.1 on page 335 and Section 6.1.2 on page 340 for descriptions of the other configuration registers.

Table 62. RTC Configuration Registers

Index	RTC Configuration Register or Action	Type	Reset
30_{16}	Activate. When bit 0 is cleared, the registers of this logical device are not accessible.	R/W	00_{16}
60_{16}	Base Address MSB register.	R/W	00_{16}
61_{16}	Base Address LSB register. Bit 0 (for A0) is read only, ‘0’.	R/W	70_{16}
62_{16}	Base Address MSB register.	R/W	00_{16}
63_{16}	Base Address LSB register. Bit 0 (for A0) is read only, ‘0’.	R/W	72_{16}
70_{16}	RTC Interrupt Number and Wake-Up on IRQ Enable register.	R/W	08_{16}
71_{16}	RTC Interrupt Type. Bit 1 is read/write; other bits are read only.	R/W	00_{16}
74_{16}	Report no DMA assignment.	RO	04_{16}
75_{16}	Report no DMA assignment.	RO	04_{16}
$F0_{16}$	RAM Lock Register (RLR)	R/W	00_{16}
$F1_{16}$	Date of Month Alarm Register Offset (DOMAO)	R/W	00_{16}
$F2_{16}$	Month Alarm Register Offset (MONAO)	R/W	00_{16}
$F3_{16}$	Century Register Offset (CENO)	R/W	00_{16}

RAM Lock Register (RLR)

When a non-reserved bit is set to 1, it can be cleared only by host domain hardware reset.

Location: Index $F0_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Block Standard RAM**	**Block RAM Write**	**Block Extended RAM Write**	**Block Extended RAM Read**	**Block Extended RAM**	**Reserved**		
Reset	0	0	0	0	0	0	0	0

Bit	Description
2-0	**Reserved.**
3	**Block Extended RAM.** Controls access to the Extended RAM 128 bytes. 0: No effect on Extended RAM access (default) 1: Reads and writes to the Extended RAM are blocked, writes are ignored and reads return FF_{16}
4	**Block Extended RAM Read.** Controls read from bytes 00_{16}-$1F_{16}$ of the Extended RAM. 0: No effect on Extended RAM access (default) 1: Reads to bytes 00_{16}-$1F_{16}$ of the Extended RAM are ignored; reads return FF_{16}
5	**Block Extended RAM Write.** Controls writes to bytes 00_{16}-$1F_{16}$ of the Extended RAM. 0: No effect on the Extended RAM access (default) 1: Writes to bytes 00_{16}-$1F_{16}$ of the Extended RAM are ignored

Bit	Description
6	**Block RAM Write.** 0: No effect on RAM access (default) 1: Writes to RAM (Standard and Extended) are ignored
7	**Block Standard RAM.** 0: No effect on Standard RAM access (default) 1: Reads and writes to locations 38_{16}-$3F_{16}$ of the Standard RAM are blocked, writes are ignored and reads return FF_{16}

Date Of Month Alarm Register Offset (DOMAO)

Location: Index $F1_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**Date of Month Alarm Register Offset Value**						
Reset	0	0	0	0	0	0	0	0

Bit	Description
6-0	**Date of Month Alarm Register Offset Value.**
7	**Reserved.**

Month Alarm Register Offset (MONAO)

Location: Index $F2_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**Month Alarm Register Offset Value**						
Reset	0	0	0	0	0	0	0	0

Bit	Description
6-0	**Month Alarm Register Offset Value.**
7	**Reserved.**

Century Register Offset (CENO)

Location: Index $F3_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Reserved**	**Century Register Offset Value**						
Reset	0	0	0	0	0	0	0	0

Bit	Description
6-0	**Century Register Offset Value.**
7	**Reserved.**

6.0 Host-Controlled Modules and Host Interface (Continued)

6.1.13 Power Management Interface Channel 1 Configuration

Logical Device 17 (11_{16}) Power Management Channel 1

- Table 63 lists the configuration registers that affect Power Management I/F Channel 1

See “Standard Logical Device Configuration Register Definitions” on page 337 and Section 6.1.2 on page 340 for descriptions of the other configuration registers.

Table 63. Power Management Configuration Registers

Index	Power Management Channel 1 Configuration Register or Action	Type	Reset
30_{16}	Activate. See also bit 0 of the SIOCF1.	R/W	00_{16}
60_{16}	Data Register Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’.	R/W	00_{16}
61_{16}	Data Register Base Address LSB register.	R/W	62_{16}
62_{16}	Command/Status Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’.	R/W	00_{16}
63_{16}	Command/Status Base Address LSB.	R/W	66_{16}
70_{16}	PM Interrupt Number and Wake-Up on IRQ Enable register.	R/W	01_{16}
71_{16}	PM Interrupt Type. Bit 1 is read/write; other bits are read only.	R/W	03_{16}
74_{16}	Report no DMA assignment.	RO	04_{16}
75_{16}	Report no DMA assignment.	RO	04_{16}

6.1.14 Power Management Interface Channel 2 Configuration

Logical Device 18 (12_{16}) Power Management Channel 2

Table 64 lists the configuration registers that affect Power Management I/F Channel 2. See Section 6.1.2 on page 340 for descriptions of the other configuration registers.

Table 64. Power Management Configuration Registers

Index	Power Management 2 Configuration Register or Action	Type	Reset
30_{16}	Activate. See also bit 0 of the SIOCF1.	R/W	00_{16}
60_{16}	Data Register Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000‘.	R/W	00_{16}
61_{16}	Data Register Base Address LSB register.	R/W	68_{16}
62_{16}	Command/Status Base Address MSB register. Bits 7-3 (for A15-11) are read only, ‘00000’.	R/W	00_{16}
63_{16}	Command/Status Base Address LSB.	R/W	$6C_{16}$
70_{16}	PM Interrupt Number and Wake-Up on IRQ Enable register.	R/W	01_{16}
71_{16}	PM Interrupt Type. Bit 1 is read/write; other bits are read only.	R/W	03_{16}
74_{16}	Report no DMA assignment.	RO	04_{16}
75_{16}	Report no DMA assignment.	RO	04_{16}

6.2 REAL-TIME CLOCK (RTC)

The RTC provides timekeeping and calendar management capabilities. It uses a 32.768 KHz signal as the basic clock for timekeeping. The RTC also includes 242 bytes of battery-backed RAM for general-purpose use.

The RTC provides the following functions:

- Accurate timekeeping and calendar management
- Alarm at a predetermined time and/or date
- Three programmable interrupt sources
- Valid timekeeping during power-down, by utilizing external battery backup
- 242 bytes of battery-backed RAM
- RAM lock schemes to protect its content
- Internal oscillator circuit (the crystal itself is off-chip) or external clock supply for the 32.768 KHz clock
- A century counter
- PnP support:
  - Relocatable index and data registers
  - Module access enable/disable option
  - Host interrupt enable/disable option
- Additional low-power features such as:
  - Automatic switching from battery to V_{CC}
  - Internal power monitoring on the VRT bit
  - Oscillator disabling to save battery during storage
- Software compatible with the DS1287 and MC146818

6.2.1 Bus Interface

The RTC function is initially mapped to the default SuperI/O locations at indexes 70_{16} to 73_{16} (two Index/Data pairs). These locations may be reassigned in compliance with Plug and Play requirements.

6.2.2 RTC Clock Generation

The RTC uses a 32.768 KHz clock signal as the basic clock for timekeeping. The 32.768 KHz clock is generated by the internal oscillator circuit or by an external oscillator (see Sections 6.2.3 and 6.2.4).

6.2.3 Internal Oscillator

The internal oscillator employs an external crystal connected to the on-chip amplifier. The on-chip amplifier is accessible on the 32KX1 input pin and the 32KX2 output pin. See Figure 113 for the recommended external circuit and Table 65 for a listing of the circuit components. The oscillator may be disabled in certain conditions. See Section 6.2.12 on page 362 for more details.

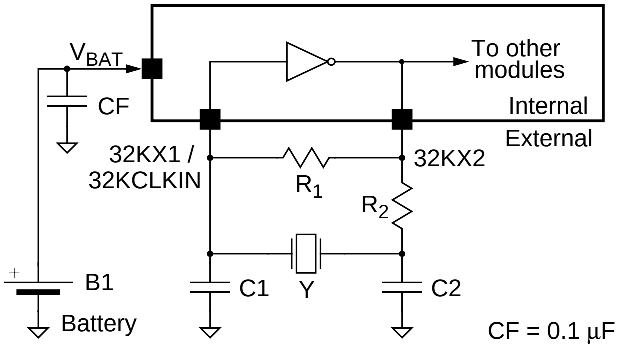


Figure 113. Recommended Oscillator External Circuitry

Host-Controlled Modules and Host Interface (Continued)

Table 65. Crystal Oscillator Circuit Components

Component	Parameters	Values	Tolerance
Crystal	Resonance Frequency	32.768 KHz Parallel Mode	User defined
	Type	N-Cut or XY-bar	
	Serial Resistance	40 KΩ	Max
	Quality Factor, Q	35000	Min
	Shunt Capacitance	2 pF	Max
	Load Capacitance, C_L	9-13 pF	
	Temperature Coefficient	User defined	
Resistor R_1	Resistance	20 MΩ	5%
Resistor R_2	Resistance	121 KΩ	5%
Capacitor C_1	Capacitance	10 pF	10%
Capacitor C_2	Capacitance	12 pF	10%

External Elements

Choose C1 and C2 capacitors (see Figure 113) to match the crystal's load capacitance. The load capacitance C_L "seen" by crystal Y is comprised of C_1 in series with C_2 and in parallel with the parasitic capacitance of the circuit. The parasitic capacitance is caused by the chip package, board layout and socket (if any), and can vary from 0 to 8 pF. The rule of thumb for choosing these capacitors is:

$$C_L = (C_1 * C_2) / (C_1 + C_2) + C_{PARASITIC}$$

Oscillator Start-Up

The oscillator starts to generate 32.768 KHz pulses to the RTC after approximately t_{32KW} from when V_{BAT} is higher than V_{BATMIN} (2.4 V) or from when V_{CC} is higher than V_{CCMIN} (3.0V).

C_1 can be trimmed to achieve precisely 32.768 KHz. To achieve a high time accuracy, use crystal and capacitors with low tolerance and temperature coefficients.

6.2.4 External Oscillator

32.768 KHz can be applied from an external clock source, as shown in Figure 114.

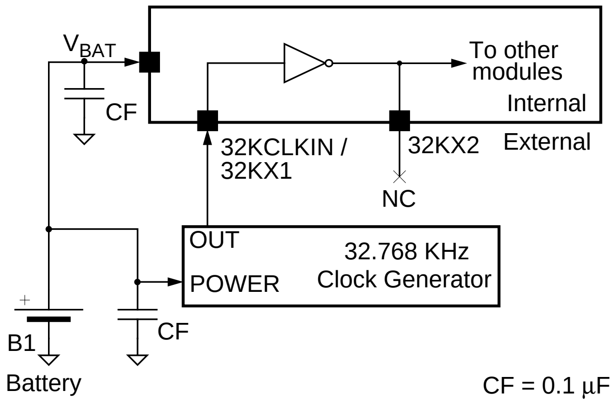


Figure 114. External Oscillator Connections

Connections

Connect the clock to the 32KCLKIN pin, leaving the oscillator output, 32KX2, unconnected.

Host-Controlled Modules and Host Interface (Continued)

Signal Parameters

The signal levels should conform to the voltage level requirements for 32KCLKIN/32KX1, stated in Section 7 on page 372. The signal should have a duty cycle of approximately 50%. To oscillate during power-down, it should be sourced from a battery-backed source. This guarantees that the RTC delivers updated time/calendar information.

6.2.5 Timing Generation

The timing generation function divides the 32.768 KHz clock by 2^{15} to derive a 1 Hz signal, which serves as the input for the seconds counter. This is performed by a divider chain composed of 15 divide-by-two latches, as shown in Figure 115.

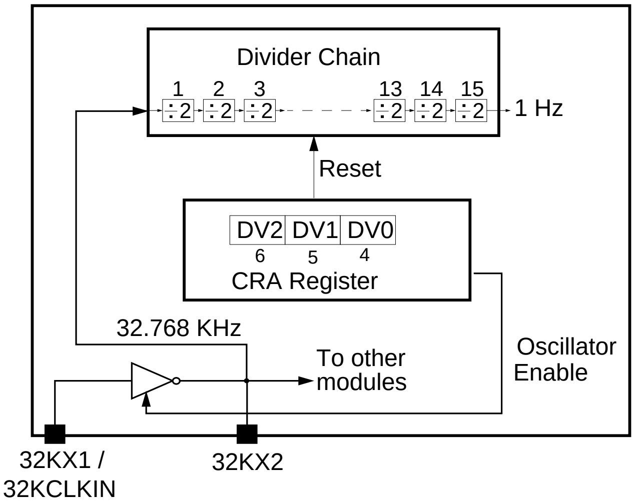


Figure 115. Divider Chain Control

Bits 6-4 (DV2-0) of CRA register control the following functions:

- Normal operation of the divider chain (counting)
- Divider chain reset to 0
- Oscillator activity when only V_{BAT} power is present (backup state).

The divider chain can be activated by setting normal operational mode (bits 6-4 of CRA = 010_2). The first update occurs 500 ms after divider chain activation.

Bits 3-0 of CRA register select one the of 15 taps from the divider chain to be used as a periodic interrupt. The periodic flag becomes active after half of the programed period has elapsed, following divider chain activation.

See “RTC Control Register A (CRA)” on page 367 for more details.

6.2.6 Timekeeping

Data Format

Time is kept in BCD or binary format, as determined by bit 2 (DM) of Control Register B (CRB), and in either 12 or 24-hour format, as determined by bit 1 of this register.

Note: When changing the above formats, re-initialize all the time registers.

Daylight Saving

Daylight saving time exceptions are handled automatically, as described in the RTC Control Register B (CRB) in Section 6.2.15 on page 363.

Leap Years

Leap year exceptions are handled automatically by the internal calendar function. Every four years, February is extended to 29 days. Year 2000 is a leap year.

6.2.7 Updating

The time and calendar registers are updated once per second regardless of bit 7 (SET) of CRB register. Since the time and calendar registers are updated serially, unpredictable results may occur if they are accessed during the update. Therefore, it is essential to ensure that reading or writing to the time storage locations does not coincide with a system update of these locations. There are several methods to avoid this contention.

Method 1

1. Set bit 7 of CRB register to 1. This takes a “snapshot” of the internal time registers and loads them into the user copy registers. The user copy registers are seen when accessing the RTC from outside and are part of the double buffering mechanism. This bit may be kept set for up to 1 second, since the time/calendar chain continues to be updated once per second.
2. Read or write the required registers (since bit 1 is set, the access is to the user copy registers). If a read operation is performed, the information read is correct from the time bit 1 was set. If a write operation is performed, the write is only to the user copy registers.
3. Reset bit 1 to 0. During the transition, the user copy registers update the internal registers, using the double buffering mechanism to ensure that the update is performed between two time updates. This mechanism enables new time parameters to be loaded in the RTC.

Method 2

1. Access the RTC registers after detection of an Update Ended interrupt. This implies that an update has just been completed, and 999 ms remain until the next update.
2. To detect an Update Ended interrupt, do one of the following:
   - Poll bit 4 of CRC register.
   - Use the following interrupt routine:
     a. Set bit 4 of CRB register.
     b. Wait for an interrupt from interrupt pin.
     c. Clear the IRQF flag of CRC register before exiting the interrupt routine.

Method 3

Poll bit 7 of CRA register. The update occurs 244 µs after this bit goes high. Therefore, if a 0 is read, the time registers remain stable for at least 244 µs.

Method 4

Use a periodic interrupt routine to determine if an update cycle is in progress, as follows:

1. Set the periodic interrupt to the desired period.
2. Set bit 6 of CRB register to enable the interrupt from periodic interrupt.
3. Wait for the periodic interrupt to occur. This indicates that the period represented by the following expression remains until another update occurs:

   [(Period of periodic interrupt / 2) + 244 µs]

6.2.8 Alarms

The timekeeping function can be set to generate an alarm when the current time reaches a stored alarm time. After each RTC time update (every 1 second), the seconds, minutes, hours, date of month and month counters are compared with their corresponding registers in the alarm settings. If equal, bit 5 of CRC register is set. Bit 5 of CRC is sent to the MSWC as an alarm signal. If the Alarm Interrupt Enable bit was previously set (bit 5 of CRB register), interrupt request pin is also active.

Any alarm register may be set to “Unconditional Match” by setting bits 7-6 to ‘11’. This combination, not used by any BCD or binary time codes, results in a periodic alarm. The rate of this periodic alarm is determined by the registers that were set to “Unconditional Match”.

For example, if all but the seconds and minutes alarm registers are set to “Unconditional Match”, an interrupt is generated every hour at the specified minute and second. If all but the seconds, minutes and hours alarm registers are set to “Unconditional Match”, an interrupt is generated every day at the specified hour, minute and second.

Host-Controlled Modules and Host Interface (Continued)

6.2.9 Power Supply

The PC87591x is supplied from four supply voltages, as shown in Figure 116 (see Section 3.1 on page 62 for more detailed description):

- System power supply voltage, V_{DD}
- System analog power supply voltage, AV_{CC}
- System standby power supply voltage, V_{CC}
- Backup voltage, from low-capacity Lithium battery

A standby voltage (V_{CC}) from the external AC/DC power supply powers the RTC under normal conditions.

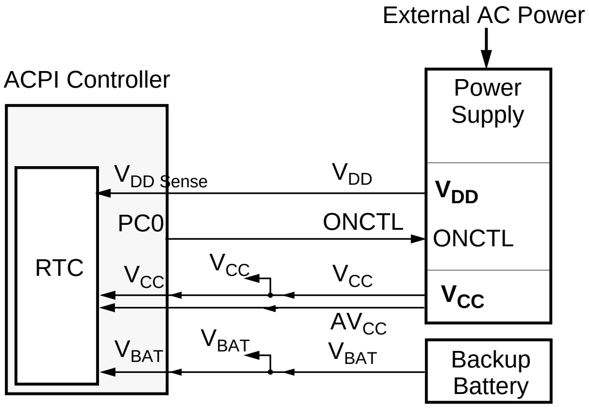
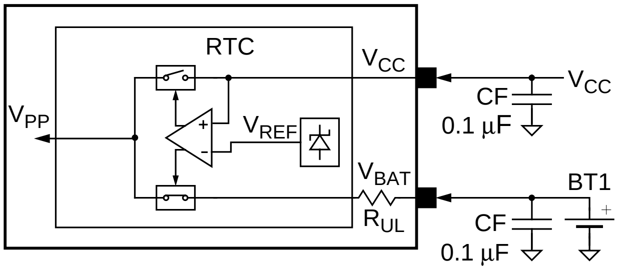


Figure 116. Power Supply Connections

Figure 117 shows a typical battery configuration. No external diode is required to meet the UL standard, because of the internal switch and internal serial resistor R_{UL}.



1. Place a 0.1 µF capacitor on each V_{CC} power supply pin as close as possible to the pin, and also on V_{BAT}.
2. Place a 10-47 µF capacitor on the common power supply net as close as possible to the device.

Figure 117. Typical Battery Configuration

RTC power is supplied from either V_{CC} or V_{BAT} voltage source, according to the sources' levels. An internal voltage comparator delivers the control signals to a pair of switches. When V_{CC} is active, the RTC power is drawn from it. Battery backup voltage (V_{BAT}) maintains the correct time and saves the CMOS memory when the V_{CC} voltage is absent due to power failure or disconnection of the main battery. Figure 117 illustrates the mechanism whereby V_{CC} or V_{BAT} is selected.

To assure that the module uses power from V_{CC} and not from V_{BAT}, the V_{CC} voltage should be maintained above its minimum (V_{CC2PP}), as detailed in Section 7 on page 372. Figure 118 illustrates the switching between V_{CC} (thick green line) and V_{BAT} (thin red line) and vice-versa for generating V_{PP} (thin blue line).

The actual voltage point where the PC87591x switches from V_{BAT} to V_{CC} is lower than the minimum workable battery voltage but high enough to guarantee the correct functionality of the oscillator and the CMOS RAM.

The Valid RAM and Timer bit (VRT) in CRD register indicates the state of non-interrupted power supplied to the PC87591x RTC module. VRT is cleared once V_{PP} power is lost, i.e., when V_{CC} and V_{BAT} are both below $V_{LOWBAT.}$ The host code should read this bit as part of the startup routine, before using the timer or the CMOS-RAM contents. In case a value of 0 is read, the RAM and timer must be initialized before use. The VRT bit is set once read.

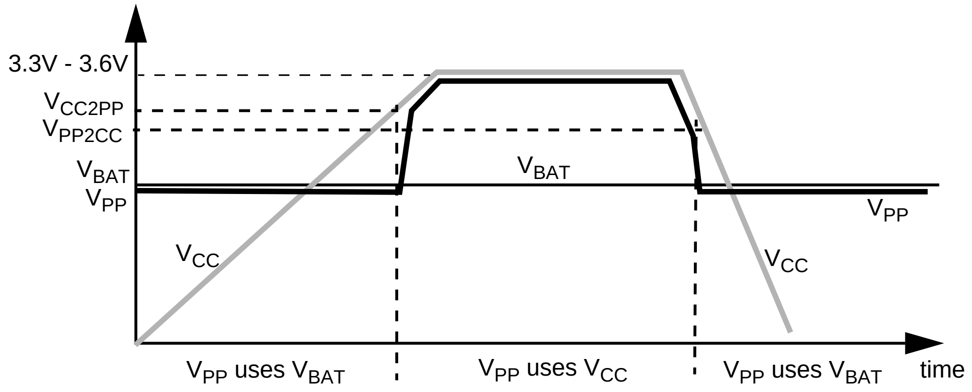


Figure 118. V_{PP} Generation Using V_{CC} or V_{BAT}

Figure 119 shows typical battery current consumption during battery-backed operation; Figure 120 shows typical battery current consumption during normal operation.

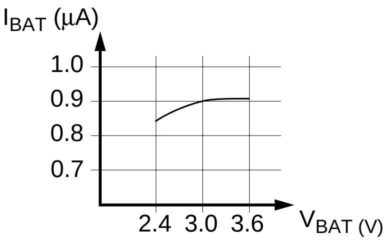
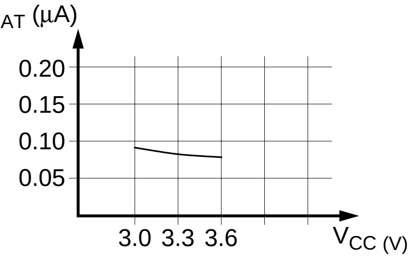


Figure 119. Typical Battery Current During Battery-Backed Power Mode



Note: Battery voltage in this test is 3.0V.

Figure 120. Typical Battery Current During Normal Operation Mode

6.2.10 System Bus Lockout

During power on or power off, spurious bus transactions from the host may occur. To protect the RTC internal registers from corruption, all inputs are automatically locked out. The lockout condition is asserted when V_{CC} is lower than V_{CCON}.

6.2.11 Power-Up Detection

When system power is restored after a power failure or power off state (V_{CC}=0), the lockout condition continues for a delay of 62 ms (minimum) to 125 ms (maximum) after the RTC switches from battery to system power.

The lockout condition is switched off immediately in the following situations:

- If the Divider Chain Control bits, DV0-2, (bits 6-4 in CRA register) specify a normal operation mode (010_2), all input signals are enabled immediately on detection of system voltage above V_{CCON}.
- When battery voltage is below V_{BATDCT} and host domain hardware reset is active, all input signals are enabled immediately on detection of system voltage above V_{CCON}. This also initializes registers at offsets 00_{16} through $0D_{16}$.
- If bit 7 (VRT) of CRD register is 0, all input signals are enabled immediately on detection of system voltage above V_{CCON}.

6.2.12 Oscillator Activity

The RTC oscillator is active if:

- V_{CC} power supply is higher than V_{CCON}, independent of the battery voltage, V_{BAT}.
- V_{BAT} power supply is higher than V_{BATMIN} whether or not V_{CC} is present.

The RTC oscillator is disabled in the following cases:

- During power-down (V_{BAT} only), if the battery voltage drops below V_{BATMIN}, the PC87591x may enter Battery Fail state. In this case, the oscillator may stop oscillating and memory contents may be corrupted or lost.
- Software wrote 00X to DV2-0 bits of CRA register and V_{CC} is removed. This disables the oscillator and decreases the power consumption from the battery connected to the V_{BAT} pin. When disabling the oscillator, the CMOS RAM is not affected as long as the battery is present at a correct voltage level.

If the RTC oscillator becomes inactive, the following features are dysfunctional/disabled:

- Timekeeping
- Periodic interrupt
- Alarm

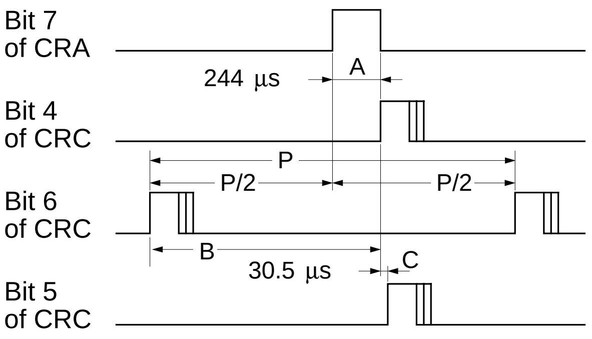
6.2.13 Interrupt Handling

The RTC has a single Interrupt Request line, which handles the following three interrupt conditions:

- Periodic interrupt
- Alarm interrupt
- Update end interrupt

The interrupts are generated if the respective enable bits in CRB register are set prior to an interrupt event occurrence. Reading the CRC register clears all interrupt flags. Therefore, when multiple interrupts are enabled, the interrupt service routine should first read and store the CRC register and then deal with all pending interrupts by referring to this stored status.

If an interrupt is not serviced before a second occurrence of the same interrupt condition, the second interrupt event is lost. Figure 121 illustrates the interrupt timing in the RTC.



Flags (and IRQ) are reset at the conclusion of CRC read or by reset.

A = Update In Progress bit high before update occurs = 244 µs
B = Periodic interrupt to update
= Period (periodic int) / 2 + 244 µs
C = Update to Alarm Interrupt = 30.5 µs
P = Period is programed by RS3-0 of CRA

Figure 121. Interrupt/Status Timing

6.2.14 Battery-Backed RAMs and Registers

The RTC has two battery-backed RAMs and 17 registers, used by the logical units themselves. Battery-backup power enables information retention during system power down.

The RAMs are:

- Standard RAM
- Extended RAM

The memory maps and register content of the RAMs are shown in Section 6.2.18 on page 371 and illustrated in Figure 122.

The first 14 bytes and three programmable bytes of the Standard RAM are overlaid by time, alarm data and control registers. The remaining 111 bytes are general-purpose memory.

Registers with reserved bits should be written using the “Read-Modify-Write” method.

All register locations within the device are accessed by the RTC Index and Data registers (at base address and base address+1, as defined by RTC configuration registers at index 60_{16} and 61_{16}). The Index register points to the register location being accessed, and the Data register contains the data to be transferred to or from the location. An additional 128 bytes of battery-backed RAM (also called Extended RAM) may be accessed via a second pair of Index and Data registers (at base address and base address+1, as defined by RTC configuration registers in index 62_{16} and 63_{16}).

Access to the two RAMs may be locked. For details, see ”RAM Lock Register (RLR)” in Section 6.1.12 on page 353.

The index of three of the RTC registers is programmable using registers in the RTC logical device bank (part of the SuperI/O configuration registers). If enabled, these registers override three of the Standard RAM locations; see Section 6.1.12 on page 353.

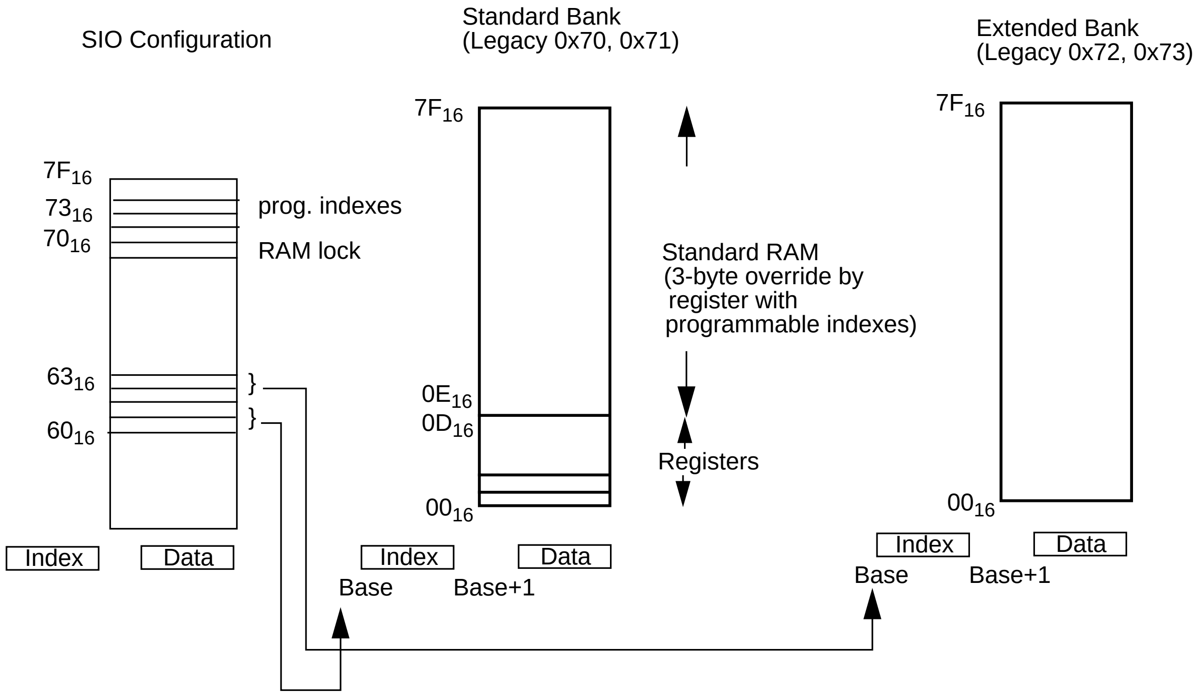


Figure 122. RTC Module Registers Mapping

6.2.15 RTC Registers

The RTC registers can be accessed at any time during normal operation mode; i.e., when V_{CC} is within the recommended operation range. This access is disabled during battery-backed operation. Write operation to these registers is also disabled if bit 7 of CRD register is 0.

Note: Before attempting to perform any start-up procedures, read the explanation of bit 7 (VRT) of CRD register.

See Section 6.2.18 on page 371 for a detailed description of the memory map for the RTC registers.

This section describes the RTC Timing and Control registers, which control basic RTC functionality.

For a summary of the abbreviations used for Register Type, see “Register Abbreviations and Access” on page 34.

Host-Controlled Modules and Host Interface (Continued)

RTC Register Map

Index	Mnemonic	Name	Type	Reset
00_{16}	SEC	Seconds Register	R/W	V_{PP} PUR
01_{16}	SECA	Seconds Alarm Register	R/W	V_{PP} PUR
02_{16}	MIN	Minutes Register	R/W	V_{PP} PUR
03_{16}	MINA	Minutes Alarm Register	R/W	V_{PP} PUR
04_{16}	HOR	Hours Register	R/W	V_{PP} PUR
05_{16}	HORA	Hours Alarm Register	R/W	V_{PP} PUR
06_{16}	DOW	Day Of Week Register	R/W	V_{PP} PUR
07_{16}	DOM	Date Of Month Register	R/W	V_{PP} PUR
08_{16}	MON	Month Register	R/W	V_{PP} PUR
09_{16}	YER	Year Register	R/W	V_{PP} PUR
$0A_{16}$	CRA	RTC Control Register A	R/W	Bit specific
$0B_{16}$	CRB	RTC Control Register B	R/W	Bit specific
$0C_{16}$	CRC	RTC Control Register C	R/O	Bit specific
$0D_{16}$	CRD	RTC Control Register D	R/O	V_{PP} PUR
Programmable[1]	DOMA	Date of Month Alarm Register	R/W	V_{PP} PUR
Programmable[1]	MONA	Month Alarm Register	R/W	V_{PP} PUR
Programmable[1]	CEN	Century Register	R/W	V_{PP} PUR

[1] Overlaid on RAM bytes in range $0E_{16}$-$7F_{16}$.

Seconds Register (SEC)

Location: Index 00_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Seconds Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Seconds Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format.

Host-Controlled Modules and Host Interface (Continued)

Seconds Alarm Register (SECA)

Location: Index 01_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Seconds Alarm Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Seconds Alarm Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected.

Minutes Register (MIN)

Location: Index 02_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Minutes Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Minutes Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format.

Minutes Alarm Register (MINA)

Location: Index 03_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Minutes Alarm Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Minutes Alarm Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected.

Hours Register (HOR)

Location: Index 04_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Hours Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Hours Data.** For 12-hour mode, values may be 01 to 12 (a.m.) and 81 to 92 (p.m.) in BCD format or 01 to 0C (a.m.) and 81 to 8C (p.m.) in binary format. For 24-hour mode, values may be 00 to 23 in BCD format or 00 to 17 in binary format.

Hours Alarm Register (HORA)

Location: Index 05_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Hours Alarm Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Hours Alarm Data.** For 12-hour mode, values may be 01 to 12 (a.m.) and 81 to 92 (p.m.) in BCD format or 01 to 0C (a.m.) and 81 to 8C (p.m.) in binary format. For 24-hour mode, values may be 00 to 23 in BCD format or 00 to 17 in binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected.

Day Of Week Register (DOW)

Location: Index 06_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Day Of Week Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Day Of Week Data.** Values may be 01 to 07 in BCD format or 01 to 07 in binary format.

Date Of Month Register (DOM)

Location: Index 07_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Date Of Month Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Date Of Month Data.** Values may be 01 to 31 in BCD format or 01 to 1F in binary format.

Month Register (MON)

Location: Index 08_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Month Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Month Data.** Values may be 01 to 12 in BCD format or 01 to 0C in binary format.

Host-Controlled Modules and Host Interface (Continued)

Year Register (YER)

Location: Index FF_{16}

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Year Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Year Data.** Values may be 00 to 99 in BCD format or 00 to 63 in binary format.

RTC Control Register A (CRA)

This register controls test selection, among other functions and cannot be written before reading bit 7 of the CRD register.

Location: Index $0A_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Update in Progress**	**Divider Chain Control 2-0**			**Periodic Interrupt Rate Select 3-0**			
Reset	0	0	1	0	0	0	0	0

Bit	Description
3-0	**Periodic Interrupt Rate Select.** These R/W bits select one of 15 output taps from the clock divider chain to control the rate of the periodic interrupt (see Table 67 on page 368 and Figure 115 on page 358). They are cleared to 000_2 as long as bit 7 of CRD register is 0.
6-4	**Divider Chain Control.** These R/W bits control the configuration of the divider chain for timing generation and register bank selection (see Table 66). They are cleared to 000_2 as long as bit 7 of CRD register is 0.
7	**Update in Progress.** This RO bit is not affected by reset; it is 0 when bit 7 of CRB register is 1. 0: Timing registers not updated within 244 µs 1: Timing registers updated within 244 µs

Table 66. Divider Chain Control and Test Selection

DV2	DV1	DV0	Configuration
CRA6	CRA5	CRA4	
0	0	X	Oscillator Disabled
0	1	0	Normal Operation
0	1	1	Test
1	0	X	
1	1	X	Divider Chain Reset

Host-Controlled Modules and Host Interface (Continued)

Table 67. Periodic Interrupt Rate Encoding

Rate Select 3 2 1 0	Periodic Interrupt Rate (ms)	Divider Chain Output
0 0 0 0	No interrupts	-
0 0 0 1	3.906250	7
0 0 1 0	7.812500	8
0 0 1 1	0.122070	2
0 1 0 0	0.244141	3
0 1 0 1	0.488281	4
0 1 1 0	0.976562	5
0 1 1 1	1.953125	6
1 0 0 0	3.906250	7
1 0 0 1	7.812500	8
1 0 1 0	15.625000	9
1 0 1 1	31.250000	10
1 1 0 0	62.500000	11
1 1 0 1	125.000000	12
1 1 1 0	250.000000	13
1 1 1 1	500.000000	14

RTC Control Register B (CRB)

Location: Index $0B_{16}$

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Set Mode**	**Periodic Interrupt Enable**	**Alarm Interrupt Enable**	**Update Ended Interrupt Enable**	**Reserved**	**Data Mode**	**Hour Mode**	**Daylight Savings**
Reset	0	0	0	0	0	0	0	0

.

Bit	Description
0	**Daylight Savings.** This bit is reset at V_{PP} Power-Up reset only. 0: Disabled 1: Enabled In the spring, time advances from 1:59:59 AM to 3:00:00 AM on the first Sunday in April. In the fall, time returns from 1:59:59 AM to 1:00:00 AM on the last Sunday in October.
1	**Hour Mode.** This bit is reset at V_{PP} Power-Up reset only. 0: 12-hour format enabled 1: 24-hour format enabled
2	**Data Mode.** This bit is reset at V_{PP} Power-Up reset only. 0: BCD format enabled 1: Binary format enabled

Host-Controlled Modules and Host Interface (Continued)

Bit	Description
3	**Reserved.** This bit is defined as “Square Wave Enable” by MC146818 and is not supported by the RTC. It is always read as 0.
4	**Update Ended Interrupt Enable.** This interrupt is generated when an update occurs. It is cleared to 0 on RTC reset (i.e., host domain reset) or when the RTC is disabled. 0: Disabled 1: Enabled
5	**Alarm Interrupt Enable.** This interrupt is generated immediately after a time update in which the seconds, minutes, hours, date and month time equal their respective alarm counterparts. It is cleared to 0 as long as bit 7 of CRD register is 0. 0: Disabled 1: Enabled
6	**Periodic Interrupt Enable.** Bits 3-0 of CRA register determine the rate at which this interrupt is generated. It is cleared to 0 on RTC reset (i.e., host domain reset) or when RTC is disabled. 0: Disabled 1: Enabled
7	**Set Mode.** This bit is reset at V_{PP} Power-Up reset only. 0: Timing updates occur normally 1: User copy of time is “frozen”, allowing the time registers to be accessed whether or not an update occurs

RTC Control Register C (CRC)

Location: Index $0C_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**IRQF**	**Periodic Interrupt Flag**	**Alarm Interrupt Flag**	**Update Ended Interrupt Flag**	**Reserved**			
Reset	0	0	0	0	0	0	0	0

Bit	Description
3-0	**Reserved.**
4	**Update Ended Interrupt Flag.** This RO bit is cleared to 0 on RTC reset (i.e., host domain reset) or when the RTC is disabled. In addition, this bit is cleared to 0 when this register is read. 0: No update occurred since the last read 1: Time registers update
5	**Alarm Interrupt Flag.** This RO bit is cleared to 0 as long as bit 7 of CRD register is 0. In addition, this bit is cleared to 0 when this register is read. 0: No alarm detected since the last read 1: Alarm condition detected
6	**Periodic Interrupt Flag.** This RO bit is cleared to 0 on RTC reset (i.e., host domain reset) or when the RTC is disabled. In addition, this bit is cleared to 0 when this register is read. 0: No transition occurred on the selected tap since the last read 1: Transition occurred on the selected tap of the divider chain
7	**IRQF (IRQ Flag).** This RO bit mirrors the value of the interrupt output signal. When interrupt is active, IRQF is 1. To clear this bit (and deactivate the interrupt), read CRC register; this clears flags UF, AF and PF, which results in IRQF being cleared, as well. 0: IRQ inactive 1: Logic equation is true: ((UIE and UF) or (AIE and AF) or (PIE and PF))

Host-Controlled Modules and Host Interface (Continued)

RTC Control Register D (CRD)

Location: Index $0D_{16}$

Type: RO

Bit	7	6	5	4	3	2	1	0
Name	**Valid RAM and Time**	**Reserved**						
Reset	0	0	0	0	0	0	0	0

Bit	Description
6-0	**Reserved.**
7	**Valid RAM and Time.** This bit senses the voltage that feeds the RTC (V_{CC} or V_{BAT}) and indicates whether or not it was too low since the last time this bit was read. If it was too low (i.e., < V_{LOWBAT}), the RTC contents (time/calendar registers and CMOS RAM) are not valid. This bit is set after the CRD is read. 0: The voltage that feeds the RTC was too low 1: RTC contents (time/calendar registers and CMOS RAM) valid

Date of Month Alarm Register (DOMA)

Location: Programmable Index

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Date of Month Alarm Data**							
Reset	1	1	0	0	0	0	0	0

Bit	Description
7-0	**Date of Month Alarm Data.** Values may be 01 to 31 in BCD format or 01 to 1F in binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected (default).

Month Alarm Register (MONA)

Location: Programmable Index

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Month Alarm Data**							
Reset	1	1	0	0	0	0	0	0

Bit	Description
7-0	**Month Alarm Data.** Values may be 01 to 12 in BCD format or 01 to 0C in binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected (default).

Host-Controlled Modules and Host Interface (Continued)

Century Register (CEN)

Location: Programmable Index

Type: R/W

Bit	7	6	5	4	3	2	1	0
Name	**Century Data**							
Reset	0	0	0	0	0	0	0	0

Bit	Description
7-0	**Century Data.** Values may be 00 to 99 in BCD format or 00 to 63 in binary format.

6.2.16 BCD and Binary Formats

Parameter	BCD Format	Binary Format
Seconds	00 to 59	00 to 3B
Minutes	00 to 59	00 to 3B
Hours	12-hour mode:01 to 12 (a.m.); 81 to 92 (p.m.) 24-hour mode:00 to 23	12-hour mode:01 to 0C (a.m.); 81 to 8C (p.m.) 24-hour mode:00 to 17
Day	01 to 07 (Sunday = 01)	01 to 07
Date	01 to 31	01 to 1F
Month	01 to 12 (January = 01)	01 to 0C
Year	00 to 99	00 to 63
Century	00 to 99	00 to 63

6.2.17 Usage Hints

1. Read bit 7 of CRD register at each system power-up to validate the contents of the RTC registers and the CMOS RAM. When this bit is 0, the contents of these registers and the CMOS RAM are questionable. This bit is reset when the backup battery voltage is too low. The voltage level at which this bit is reset is below the minimum recommended battery voltage, 2.4V. Although the RTC oscillator may function properly and the register contents may be correct at lower than 2.4V, this bit is reset since correct functionality cannot be guaranteed. System BIOS may use a checksum method to revalidate the contents of the CMOS-RAM. The checksum byte should be stored in the same CMOS RAM.
2. Change the backup battery while normal operating power is present, and not in backup mode, to maintain valid time and register information. If a low leakage capacitor is connected to V_{BAT}, the battery may be changed in backup mode.
3. A rechargeable NiCd battery may be used instead of a non-rechargeable Lithium battery. This is the preferred solution for portable systems, where small components are essential.
4. A supercap capacitor may be used instead of the normal Lithium battery. In a portable system, typically, the V_{SB} voltage is always present because the power management stops the system before its voltage falls to low. A supercap capacitor in the range of 0.047F-0.47F should be able to supply the power during a battery replacement.

6.2.18 RTC General-Purpose RAM Map

Table 68. Standard RAM Map

Index	Description
$0E_{16}$ - $7F_{16}$[1]	Battery-backed general-purpose 111-byte RAM.

[1] Battery-backed 111-byte RAM (114 – 3 overlaid registers).

Table 69. Extended RAM Map

Index	Description
00_{16} - $7F_{16}$	Battery-backed general-purpose 128-byte RAM.

7.0 Device Specifications

This section provides the power and grounding guidelines for the PC87591x, specifies the device's maximum ratings and electrical characteristics and describes its timing.

7.1 GENERAL DC ELECTRICAL CHARACTERISTICS

7.1.1 Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
V_{DD}	Host Domain Supply Voltage	3.0	3.3	3.6	V
V_{CC}	Core Domain Supply Voltage	3.0	3.3	3.6	V
AV_{CC}	Analog Supply Voltage	3.15	3.3	3.45	V
V_{BAT}	Battery Backup Supply Voltage	2.4		3.6	V
T_A	Operating Temperature	0		+70	°C

7.1.2 Absolute Maximum Ratings

If military- or aerospace-specified devices are required, please contact a National Semiconductor sales office or distributor for availability and specifications.

Storage Temperature: –65°C to +150°C

Temperature Under Bias: 0°C to +70°C

Absolute maximum ratings are values beyond which damage to the device may occur. Unless otherwise specified, all voltages are relative to ground.

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	Host Domain Supply Voltage		–0.5	+4.2	V
V_{CC}	Core Domain Supply Voltage		–0.5	+4.2	V
AV_{CC}	Analog Supply Voltage		–0.5	+4.2	V
V_{BAT}	Battery Backup Supply Voltage		–0.5	+4.2	V
V_I	Input Voltage	All other buffer types	–0.5	5.5	V
		Buffer types: IN_{AC}[1], IN_{AD}, IN_{DI}, IN_{OSC}, IN_{PCI}, IN_{TS} (IOPE0-3, IOPE6-7)	–0.5	V_{SUP}[2] + 0.5	V
V_O	Output Voltage	All other buffer types	–0.5	5.5	V
		Buffer types: O_{DA}, O_{DI}, O_{OSC}, O_{PCI}	–0.5	V_{SUP}[3] + 0.5	V
T_{STG}	Storage Temperature		–65	+165	°C
P_D	Power Dissipation			1	W
T_L	Lead Temperature Soldering (10 s)			+260	°C
	ESD Tolerance	C_{ZAP} = 100 pF R_{ZAP} = 1.5 KΩ[4]	2000		V

1. When ACM is enabled.
2. V_{SUP} is V_{DD}, V_{CC}, AV_{CC} or V_{BAT}, according to the power well of the input.
3. V_{SUP} is V_{DD}, V_{CC}, AV_{CC} or V_{BAT}, according to the power well of the output.
4. Value based on test complying with RAI-5-048-RA human body model ESD testing.

7.0 Device Specifications (Continued)

7.1.3 Capacitance

Symbol	Parameter	Min[2]	Typ[1]	Max[2]	Unit
C_{IN}	Input Pin Capacitance		5	7	pF
C_{IN1}	Clock Input Capacitance	5	8	12	pF
C_{IO}	I/O Pin Capacitance		10	12	pF
C_O	Output Pin Capacitance		6	8	pF

1. T_A = 25°C; f = 1 MHz.
2. Not tested; guaranteed by characterization.

7.1.4 Power Supply Current Consumption under Recommended Operating Conditions

Symbol	Parameter	Conditions[1]	Typ	Max	Unit
I_{DD}	V_{DD} Average Main Supply Current	V_{IL} = 0.5V, V_{IH} = 2.4V No Load	2	3	mA
I_{DDLP}	V_{DD} Quiescent Main Supply Current in Low Power Mode	V_{IL} = GND, V_{IH} = V_{DD} No Load	30	50	µA
I_{CC}	V_{CC} Active Supply Current	t_{CLK} = 250 ns	15		mA
		t_{CLK} = 50 ns	23	30	mA
I_{CCW}	V_{CC} Active Executing WAIT Supply Current	t_{CLK} = 250 ns	6.6		mA
		t_{CLK} = 50 ns	10		mA
I_{CCI}	V_{CC} Idle Mode Supply Current	Idle Mode V_{IL} = GND, V_{IH} = V_{CC} No Load		15	µA
I_{BAT}	V_{BAT} Battery Supply Current	Power Off Mode	0.9	1.5	µA

1. All parameters specified for 0°C ≤ T_A ≤ 70°C; V_{DD} and V_{CC} = 3.3V ±10% unless otherwise specified.

7.1.5 Voltage Thresholds

Symbol	Parameter	Conditions[1]	Min	Typ	Max	Unit
V_{CCON}	V_{CC} Detected as Power-On		2.5		2.95	V
V_{CC2PP}	V_{CC} detected for use as source for V_{PP}		2.3		2.4	V
V_{PP2CC}	V_{CC} detected inactive for use of V_{BAT} as source for V_{PP}		1.9		2.2	V
V_{DDON}	V_{DD} Detected as on		2.5		2.95	V
V_{BATDTC}	Battery Detected		1.0		1.2	V
V_{LOWBAT}	Low Battery Voltage		1.3		1.9	V
V_{BATMIN}	Workable Battery Voltage		2.4			V
V_{BATMAX}	Battery Input Voltage				3.6	V

1. All parameters specified for 0°C ≤ T_A ≤ 70°C.

7.0 Device Specifications (Continued)

7.2 DC CHARACTERISTICS OF PINS BY I/O BUFFER TYPES

The following tables summarize the DC characteristics of all device pins described in Chapter 2 on page 39. The characteristics describe the general I/O buffer types defined in Table 4 on page 41. For exceptions, see Section 7.2.9 on page 376. For the DC characteristics of the analog pins, see Section 7.4 on page 378. The DC characteristics of the system interface meet the PCI 2.1 3.3V DC signaling.

7.2.1 Input, CMOS Compatible with Schmitt Trigger

Symbol: IN_{CS}

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	Input High Voltage		0.75 V_{SUP}[1]	5.5[2]	V
V_{IL}	Input Low Voltage		–0.5[2]	1.1	V
V_H	Input Hysteresis		500[3]		mV
I_{IL}	Input Leakage Current	$0 < V_{IN} < V_{SUP}$		±1[4]	µA

1. V_{SUP} is V_{DD} or V_{CC}, according to the power well of the input.
2. Not tested; guaranteed by design.
3. Not tested; guaranteed by characterization.
4. Maximum 10 µA for all pins together.

7.2.2 Input, PCI 3.3V

Symbol: IN_{PCI}

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	Input High Voltage		0.5 V_{DD}	V_{DD} + 0.5[1]	V
V_{IL}	Input Low Voltage		–0.5[1]	0.3 V_{DD}	V
I_{IL}[2]	Input Leakage Current	$0 < V_{IN} < V_{DD}$		±1[3]	µA

1. Not tested; guaranteed by design.
2. Input leakage current includes the output leakage of the bidirectional buffers with TRI-STATE outputs.
3. Maximum 10 µA for all pins together.

7.2.3 Input, SMBus Compatible

Symbol: IN_{SM}

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	Input High Voltage		1.4	5.5[1]	V
V_{IL}	Input Low Voltage		–0.5 [1]	0.8	V
I_{IL}[2]	Input Leakage Current	$0 < V_{IN} < V_{DD}$		±1[3]	µA

1. Not tested; guaranteed by design.
2. Input leakage current includes the output leakage of the bidirectional buffers with TRI-STATE outputs.
3. Maximum 10 µA for all pins together.

7.0 Device Specifications (Continued)

7.2.4 Input, TTL Compatible

Symbol: IN_T

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	Input High Voltage		2.0	5.5[1]	V
V_{IL}	Input Low Voltage		–0.5[1]	0.8	V
I_{IL}[2]	Input Leakage Current	$0 < V_{IN} < V_{CC}$		±1[3]	µA

[1] Not tested; guaranteed by design.
[2] Input leakage current includes the output leakage of the bidirectional buffers with TRI-STATE outputs.
[3] Maximum 10 µA for all pins together.

7.2.5 Input, TTL Compatible with Schmitt Trigger

Symbol: IN_{TS}

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	Input High Voltage, 5V tolerant pins[1]		2.0	5.5[2]	V
	Input High Voltage, pins without 5V tolerance[1]		2.0	V_{SUP}[3]+0.5[2]	V
V_{IL}	Input Low Voltage		–0.5[2]	0.8	V
V_H	Input Hysteresis		250[4]		mV
I_{IL}	Input Leakage Current	$0 < V_{IN} < V_{SUP}$		±1[5]	µA

[1] See Section 7.1.2 on page 372.
[2] Not tested; guaranteed by design.
[3] V_{SUP} is V_{DD} or V_{CC}, according to the power well of the input.
[4] Not tested; guaranteed by characterization.
[5] Maximum 10 µA for all pins together.

7.2.6 Output, TTL Compatible Push-Pull Buffer

Symbol: $O_{p/n}$

Output, TTL-compatible, rail-to-rail push-pull buffer that is capable of sourcing *p* mA and sinking *n* mA

Symbol	Parameter	Conditions	Min	Max	Unit
V_{OH}	Output High Voltage	$I_{OH} = –p$ mA	2.4		V
		I_{OH} = –50 µA	V_{SUP}[1] – 0.2		V
V_{OL}	Output Low Voltage	$I_{OL} = n$ mA		0.4	V
		I_{OL} = 50 µA		0.2	V

[1] V_{SUP} is V_{DD} or V_{CC}, according to the power well of the input.

7.0 Device Specifications (Continued)

7.2.7 Output, Open-Drain Buffer

Symbol: OD_n

Output, TTL-compatible open-drain output buffer, capable of sinking *n* mA. Output from these signals is open-drain and is never forced high.

Symbol	Parameter	Conditions	Min	Max	Unit
V_{OL}	Output Low Voltage	I_{OL} = *n* mA		0.4	V
		I_{OL} = 50 µA		0.2	V

7.2.8 Output, PCI 3.3V

Symbol: $O_{PCI\ I}$

Symbol	Parameter	Conditions	Min	Max	Unit
V_{OH}	Output High Voltage	I_{out} = –500 µA	0.9 V_{DD}		V
V_{OL}	Output Low Voltage	I_{out} = 1500 µA		0.1 V_{DD}	V

7.2.9 Exceptions

1. All pins are back-drive protected, except for output pins with PCI buffer type (IN_{PCI} or O_{PCI}), oscillator (O_{OSC}) and all analog type pins (O_{DA} and O_{DI}).
2. The following pins have an internal static pull-up resistor and therefore may have leakage current to V_{CC} (when V_{IN} = 0): SCL1-2, SDA1-2, IOPA7-0, IOPB7-0, IOPC7-0, IOPD7-0, IOPE7-4, IOPF7-0, KBSIN0-7, PSCLK1-4, PSDAT1-4.
3. The following strap pins have an internal static pull-down resistor enabled during Power-Up reset and therefore may have leakage current to V_{SS} (when V_{IN} = V_{SUP}): BADDR1-0, ENV1-0, SHBM, TRIS.
4. I_{OH} is valid for a GPIO pin only when it is not configured as open-drain.

7.2.10 Terminology

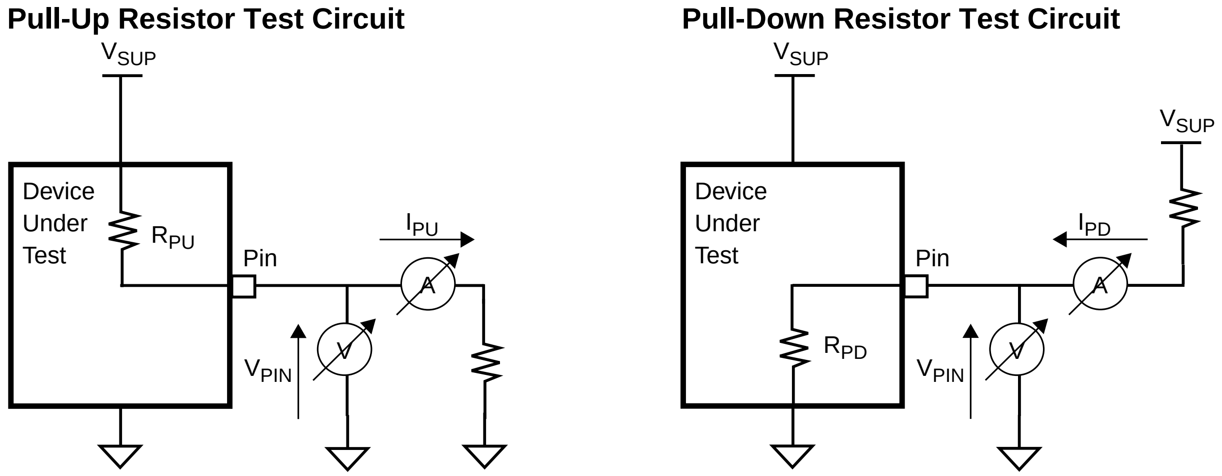
Back-Drive Protection. A pin that is back-drive protected does not sink current into the supply when an input voltage higher than the supply, but below the pin's maximum input voltage, is applied to the pin. This is true even when the supply is inactive. Note that active pull-up resistors and active output buffers are typically not back-drive protected.

5-Volt Tolerance. An input signal that is 5V tolerant can operate with input voltage of up to 5V even though the supply to the device is only 3.3V. The actual maximum input voltage allowed to be supplied to the pin is indicated by the maximum high voltage allowed for the input buffer. Note that some pins have multiple buffers, not all of which are 5V tolerant. In such cases, there is a note that indicates at what conditions a 5V input may be applied to the pin; if there is no note, the low maximum voltage among the buffers is the maximum voltage allowed for the pin.

7.0 Device Specifications (Continued)

7.3 INTERNAL RESISTORS

DC Test Conditions


Pull-Up Resistor Test Circuit

Pull-Down Resistor Test Circuit

Figure 123. Internal Resistor Test Conditions, T_A = 0°C to 70°C, V_{SUP} = 3.3V

Notes:

1. The equivalent resistance of the pull-up resistor is calculated by $R_{PU} = (V_{SUP} - V_{PIN}) / I_{PU}$.
2. The equivalent resistance of the pull-down resistor is calculated by $R_{PD} = V_{PIN} / I_{PD}$.

7.3.1 Pull-Up Resistor

Symbol: PU*nn*

Symbol	Parameter	Conditions	Typical	Min	Max	Unit
R_{PU}[1]	Pull-up equivalent resistance	V_{PIN} = 0V	*nn*	*nn* – 34%	*nn* + 47%	KΩ
		V_{PIN} = 0.17 V_{DD}		*nn* – 44%		KΩ
		V_{PIN} = 0.8 V_{DD}			*nn* – 41%	KΩ

[1] Not tested; guaranteed by characterization.

7.3.2 Pull-Down Resistor

Symbol: PD*nn*

Symbol	Parameter	Conditions	Typical	Min	Max	Unit
R_{PD}[1]	Pull-down equivalent resistance	V_{PIN} = V_{DD}	*nn*	*nn* – 35%	*nn* + 60%	KΩ
		V_{PIN} = 0.17 V_{DD}			*nn* – 37%	KΩ
		V_{PIN} = 0.8 V_{DD}		*nn* – 48%		KΩ

[1] Not tested; guaranteed by characterization.

7.0 Device Specifications (Continued)

7.4 ANALOG CHARACTERISTICS

7.4.1 ADC Characteristics

Voltage Measurement

Parameter	Symbol	Conditions[1]	Min	Typ	Max	Unit
Resolution	RES			10		Bit
Offset Error[2]	OER_L[3]	AD0-9: 12 mV ≤ V_{IN} ≤ 0.9 V			±0.25	LSB
	OER_H	AD0-9: 40 mV ≤ V_{IN} ≤ AV_{CC} –0.8			±0.25	LSB
	OER_B	AD13: 0.9V ≤ V_{IN} ≤ AV_{CC} –0.2			±0.25	LSB
	OER_V	AD10-12: 0.1V ≤ V_{IN} ≤ V_{FS}			±0.25	LSB
Gain Error[4]	GER_L	AD0-9: 12 mV ≤ V_{IN} ≤ 0.9 V			±0.75	LSB
	GER_H	AD0-9: 40 mV ≤ V_{IN} ≤ AV_{CC} –0.8			±0.75	LSB
	GER_B	AD13: 0.9V ≤ V_{IN} ≤ AV_{CC} –0.2			±0.75	LSB
	GER_V	AD10-12: 0.1V ≤ V_{IN} ≤ V_{FS}			±0.75	LSB
Integral Non-linearity Error[5]	INL_L	AD0-9: 12 mV ≤ V_{IN} ≤ 0.9 V			±0.5	LSB
	INL_H	AD0-9: 40 mV ≤ V_{IN} ≤ AV_{CC} –0.8			±0.5	LSB
	INL_B	AD13: 0.9V ≤ V_{IN} ≤ AV_{CC} –0.2			±0.5	LSB
	INL_V	AD10-12: 0.1V ≤ V_{IN} ≤ V_{FS}			±0.5	LSB
Differential Non-linearity Error[6]	DNL_L	AD0-9: 12 mV ≤ V_{IN} ≤ 0.9 V			±0.25[7]	LSB
	DNL_H	AD0-9: 40 mV ≤ V_{IN} ≤ AV_{CC} –0.8			±0.25[7]	LSB
	DNL_B	AD13: 0.9V ≤ V_{IN} ≤ AV_{CC} –0.2			±0.25[7]	LSB
	DNL_V	AD10-12: 0.1V ≤ V_{IN} ≤ V_{FS}			±0.25[7]	LSB
External Inputs Accuracy[8]	$EACU_L$	AD0-9: 12 mV ≤ V_{IN} ≤ 0.9 V			±1	LSB
	$EACU_H$	AD0-9: 40 mV ≤ V_{IN} ≤ AV_{CC} –0.8			±1	LSB

7.0 Device Specifications (Continued)

Parameter	Symbol	Conditions[1]	Min	Typ	Max	Unit
Internal Inputs Accuracy[8, 9]	$IACU_B$	AD13: $0.9V \le V_{IN} \le AV_{CC} -0.2$			±2	LSB
	$IACU_V$	AD10-12: $0.1V \le V_{IN} \le V_{FS}$			±2	LSB
Full Scale Voltage	V_{FSL}	AD0-9 Low Scale		1.000		V
	V_{FSH}	AD0-9 High Scale		3.000		V
	V_{FSB}	AD13		4.000		V
	V_{FSV}	AD10-12		4.000		V
Input Voltage Range	V_{IN}	Note[10]	0		AV_{CC}	V
Analog Input Leakage Current	I_{AL}	AD0-9: $0 \le V_{IN} \le AV_{CC}$			±1	µA
Analog Input Resistance	R_{AIN}	AD0-9, AD13	4			MΩ
Analog Input Capacitance	C_{AIN}				TBD	pF
ADC Clock Frequency	F_{CLK}			0.5		MHz
ADC Enable Delay[11]	t_{END}				100	µs
Voltage Conversion Duration	t_{VC}				8.2	ms

1. All parameters specified for $0°C \le T_A \le 70°C$ and AV_{CC}= 3.3V ± 5% unless otherwise specified.
2. The difference between 0V and the actual voltage value for code 000_{16}.
3. XXX_L = Inputs AD0-9, Low Scale; XXX_H = Inputs AD0-9, High Scale; XXX_V = Inputs AD10-12; XXX_B = Input AD13.
4. The difference between $^{1023}/_{1024} * V_{FS}$ and the actual voltage for code $3FF_{16}$.
5. The maximum difference between the ideal (straight) conversion line and the actual conversion curve, not including the offset, gain and quantization (±0.5 LSB) errors.
6. The maximum difference between an ideal step size (1 LSB) and any actual step size.
7. No missing codes.
8. Total unadjusted error (includes the offset, gain, integral non-linearity and quantization (±0.5 LSB) errors).
9. The internal power supply inputs: $V_{DD,}$ $V_{CC,}$ $AV_{CC.}$
10. Input Voltage allowed in normal operation. Linear range is as defined for the different measurement modes.
11. Time from the moment when ADCEN=1 in ADCCNF register until the beginning of the “ADC cycle”.

7.0 Device Specifications (Continued)

Temperature Measurement

Parameter	Symbol	Conditions[1]	Min	Typ	Max	Unit
Resolution	RESB			8		Bit
	RESC			1		°C
Accuracy Using Remote Diode[2]	RAC1	T_A = +40°C to +100°C			±3	°C
	RAC2	T_A = –40°C to +125°C			±5	°C
Accuracy Using Local Diode	LAC1	T_A = +40°C to +100°C			±1	°C
	LAC2	T_A = –40°C to +125°C			±3	°C
Voltage at DN	V_{DN}	8 µA ≤ I_D ≤ 120 µA		0.7		V
DP Source Current	I_{DPH}	High Level	80		120	µA
	I_{DPL}	Low Level	8		12	µA
Temperature Input Capacitance	C_{TIN}				TBD	pF
ADC Clock Frequency	F_{CLK}			0.5		MHz
ADC Enable Delay[3]	t_{END}				100	µs
Temperature Conversion Duration	t_{TC}				42	ms

1. All parameters specified for 0°C ≤ T_A ≤ 70°C and AV_{CC}= 3.3V ± 5% unless otherwise specified.
2. See “Remote Diode Selection” on page 187.
3. Time from the moment when ADCEN=1 in ADCCNF register until the beginning of the “ADC cycle”.

7.4.2 ACM Characteristics

Parameter	Symbol	Conditions[1]	Min	Typ	Max	Unit
Resolution	RES			6		Bit
Differential Non-linearity Error[2]	DNL	0 ≤ V_{IN} ≤ V_{CC}			±0.5[3]	LSB
Accuracy (Total unadjusted error)	ACU	0 ≤ V_{IN} ≤ V_{CC}			±1.5	LSB
Input Voltage Range	V_{IN}		0		V_{CC}	V
Analog Input Leakage Current	I_{AL}	0 ≤ V_{IN} ≤ V_{CC}			±1	µA
Analog Input Capacitance	C_{AIN}				TBD	pF

1. All parameters specified for 0°C ≤ T_A ≤ 70°C and V_{CC}= 3.3V ± 10% unless otherwise specified.
2. The maximum difference between an ideal step size (1 LSB) and any actual step size.
3. No missing codes.

7.0 Device Specifications (Continued)

7.4.3 DAC Characteristics

Parameter	Symbol	Conditions[1]	Min	Typ	Max	Unit
Resolution	RES			8		Bit
Offset Error[2]	OER	AV_{CC} = 3.3V			±1	LSB
Gain Error[3]	GER	AV_{CC} = 3.3V			±1	LSB
Integral Non-linearity Error[4]	INL	AV_{CC} = 3.3V, $0 \le V_{OUT} \le AV_{CC}$			±0.5	LSB
Differential Non-linearity Error[5]	DNL	AV_{CC} = 3.3V, $0 \le V_{OUT} \le AV_{CC}$			±0.5[6]	LSB
Output Voltage Range	V_{OUT}		0		AV_{CC}	V
Analog Output Resistance	R_S	$0 \le V_{OUT} \le AV_{CC}$	2	3	4	KΩ
Analog Output Capacitance	C_{AO}			10	15	pF
DAC Settling Time[7]	T_{SET}	C_L = 50 pF			1	µs
DAC Enable Delay[8]	T_{END}	C_L = 50 pF			10	µs

1. All parameters specified for 0°C ≤ T_A ≤ 70°C. AV_{CC} = 3.3V ± 5%, unless otherwise specified.
2. The difference between 0V and the actual voltage value for code 00_{16}.
3. The difference between $^{255}/_{256} * AV_{CC}$ and the actual voltage for code FF_{16}.
4. The maximum difference between the ideal (straight) conversion line and the actual conversion curve, not including the offset, gain and quantization (±0.5 LSB) errors.
5. The maximum difference between an ideal step size (1 LSB) and any actual step size.
6. Monotonic.
7. Time from the converter loading with data, to output voltage settling within an error of ±0.5 LSB.
8. Time from the moment when DACEN*n*=1 in DACCTRL register until the settling of the output voltage.

7.5 PACKAGE THERMAL INFORMATION

Thermal resistance (degrees C/W) $Theta_{JC}$ and $Theta_{JA}$ values for the various packages of the PC87591x are as follows:

Table 70. Theta (Θ) J Values

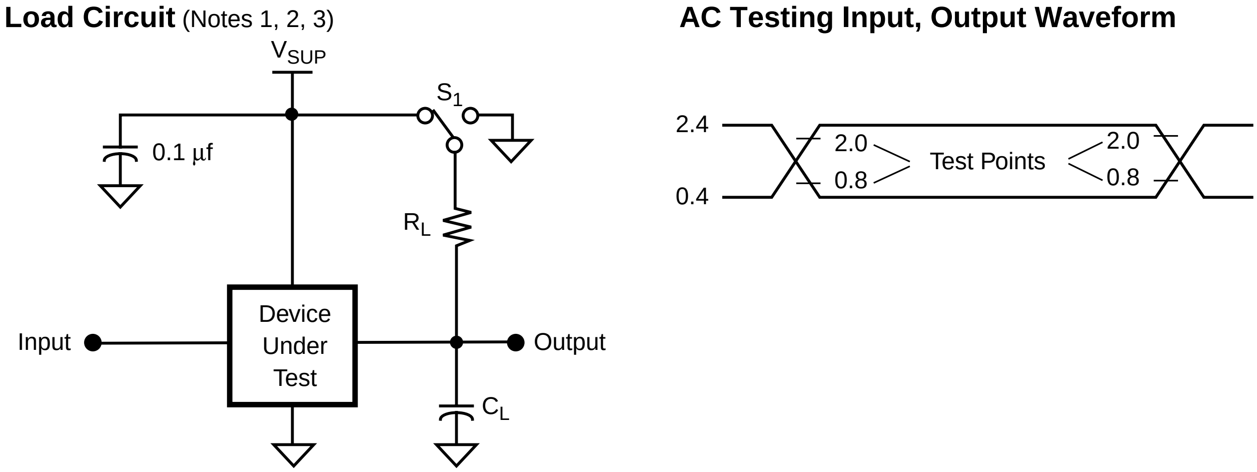
Package Type	$Theta_{JA}$@0 lfpm	$Theta_{JA}$@225 lfpm	$Theta_{JA}$@500 lfpm	$Theta_{JA}$@900 lfpm	$Theta_{JC}$
128 LQFP	47.1	34.6	29.3	28.8	15.3
176 LQFP	39.8	32.3	29.5	26.2	5.1

Airflow for $Theta_{JA}$ values is measured in linear feet per minute (lfpm).

7.0 Device Specifications (Continued)

7.6 AC ELECTRICAL CHARACTERISTICS

7.6.1 AC Test Conditions


Load Circuit (Notes 1, 2, 3)

AC Testing Input, Output Waveform

Figure 124. AC Test Conditions, T_A = 0°C to 70°C, V_{SUP} = 3.3V ±10%

Notes:

1. V_{SUP} is V_{DD}, V_{CC}, AV_{CC} or V_{BAT}, according to the power well of the pin.
2. C_L = 50 pF for all output pins (this value includes both jig and oscilloscope capacitance).
3. S_1 = Open for push-pull output pins.
   S_1 = V_{SUP} for high impedance to active low and active low to high impedance measurements.
   S_1 = V_{SS} for high impedance to active high and active high to high impedance measurements.
   R_L = 1.0 KΩ

The following abbreviations are used in Section 7.6:

RE = Rising Edge
FE = Falling Edge

Definitions

The timing specifications in this section refer to low- or high-level voltage, according to the specific buffer type (TTL or CMOS) on the rising or falling edges of all the signals, as illustrated in the following figures, unless specifically stated otherwise.

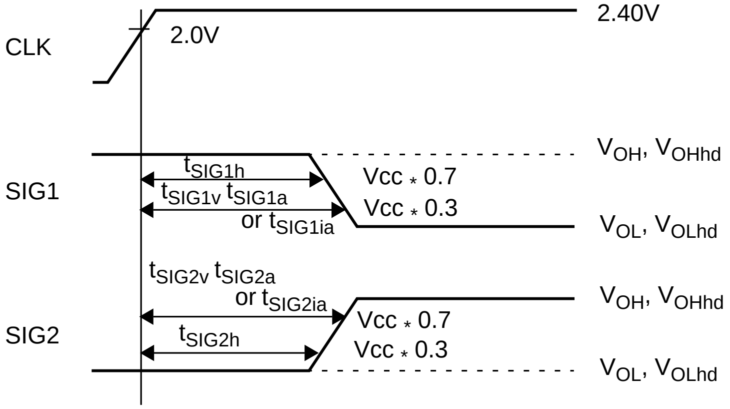


Figure 125. CMOS Output Signals Specification Conventions

7.0 Device Specifications (Continued)

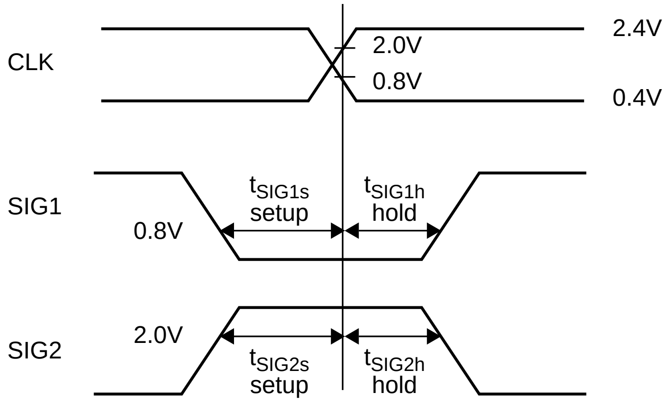


Figure 126. TTL: Input Signal Specification Standard

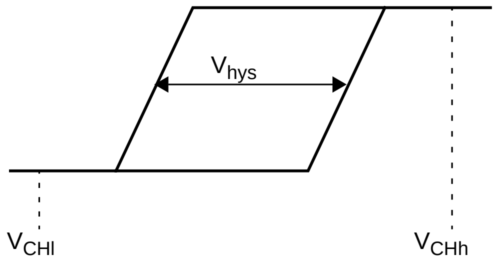


Figure 127. CMOS with Hysteresis Inputs

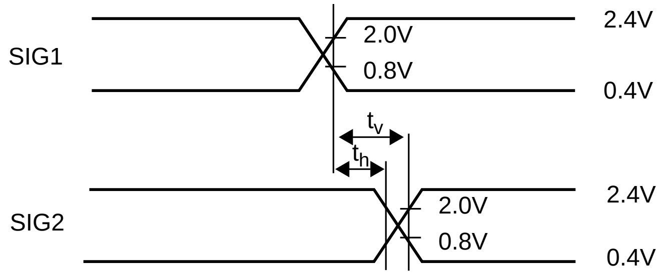


Figure 128. Signal-to-Signal Delay

7.0 Device Specifications (Continued)

7.6.2 Reset Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
t_{SUPUP}	130	Supply wake-up time to 95% V_{CC}	After $V_{CC} > V_{CCON}$		1	ms
t_{IRST}	130	Internal or power-on reset time	Power stable to end of 16th clock cycle of CLK	$16 * t_{CLK} + t_{32KD}$	$16 * t_{CLK} + t_{32KD} + t_{32KW}$	-
t_{IPLv}	130	Valid time: Internal strap pull-up resistors	Before end of 16th clock cycle of CLK	$16 * t_{CLK}$		-
t_{EPLv}	130	Valid time: External strap pull-up resistors	Before end of 16th clock cycle of CLK	$6 * t_{CLK}$		-
t_{WRST}	130	$\overline{\text{RESET1-2}}$ width	$\overline{\text{RESET1-2}}$ FE to $\overline{\text{RESET1-2}}$ RE	$3 * t_{CLK}$		-

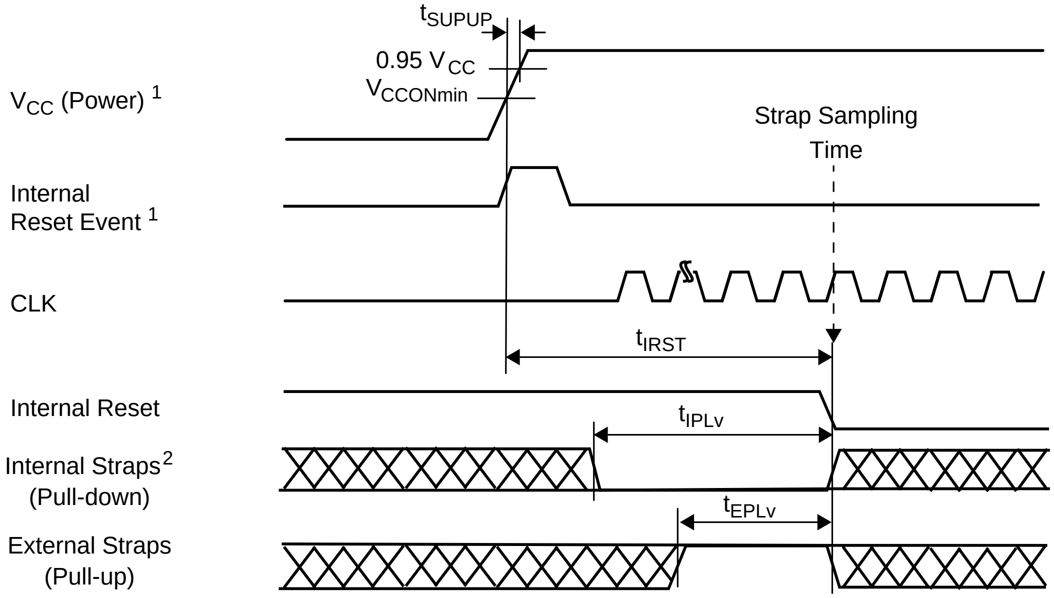


Figure 129. Internal Power-Up Reset

Notes:

1. Either WATCHDOG, Debugger I/F or Power-Up reset.
2. Valid on Power-Up reset only.

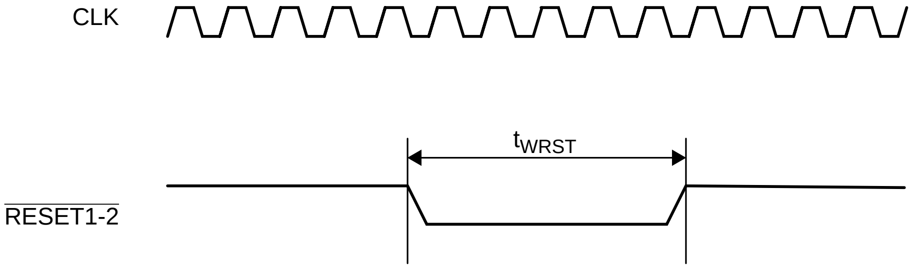


Figure 130. Warm Hardware Reset

7.0 Device Specifications (Continued)

7.6.3 Clock Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
Clock Input Timing						
$t_{32KCLKIN}$	131	Required clock period for 32KCLKIN	From RE to RE of 32KCLKIN t_{32NOM} = 30.517578	30.5145 (t_{32NOM} – 100ppm)	30.5206 (t_{32NOM} + 100ppm)	µs
Clock Output Timing						
t_{CLK}	132	CLK period	At 2.0V (both edges)	50	250	ns
t_{CLKh}	132	CLK high time	At 2.0V (both edges)	$0.5 * t_{CLK} - 5$ ns		-
t_{CLKl}	132	CLK low time	At 0.8V (both edges)	$0.5 * t_{CLK} - 5$ ns		-
t_{CLKr}	132	CLK rise time	0.8V to 2.0V		6	ns
t_{CLKf}	132	CLK fall time	2.0V to 0.8V		6	ns
t_{CLKw}	133	CLK wake-up time	From wake-up event until CLK starts toggling		100	µs
$t_{CLKINTst}$	133	t_{CLK} period	Active mode in steady state	$0.99 * t_{CLKINTnom}$[1]	$1.01 * t_{CLKINTnom}$	-
$t_{CLKINTwk}$	133	t_{CLK} period	After wake-up from Idle	$0.9 * t_{CLKINTnom}$	$1.1 * t_{CLKINTnom}$	-
$t_{CLKstab}$	133	t_{CLK} stabilization time	After wake-up from Idle		0.5	s
t_{32KW}	131	32K oscillator wake-up time	After $V_{BAT} > V_{LOWBAT}$		1	s
t_{32KD}	132	CLK delay time	After $V_{CC} > V_{CCON}$		40	ms
Flash Operation Frequency						
f_{FMAX}		Flash Section 0 and 1 maximal operation frequency in Fast Read	CLK period		20	MHz
t_{FLPMAX}		Flash Section 0 and 1 maximal operation frequency in Low Power	CLK period		TBD	MHz

[1] $t_{CLKINTnom}$ is defined in Table 36 on page 241.

7.0 Device Specifications (Continued)

General

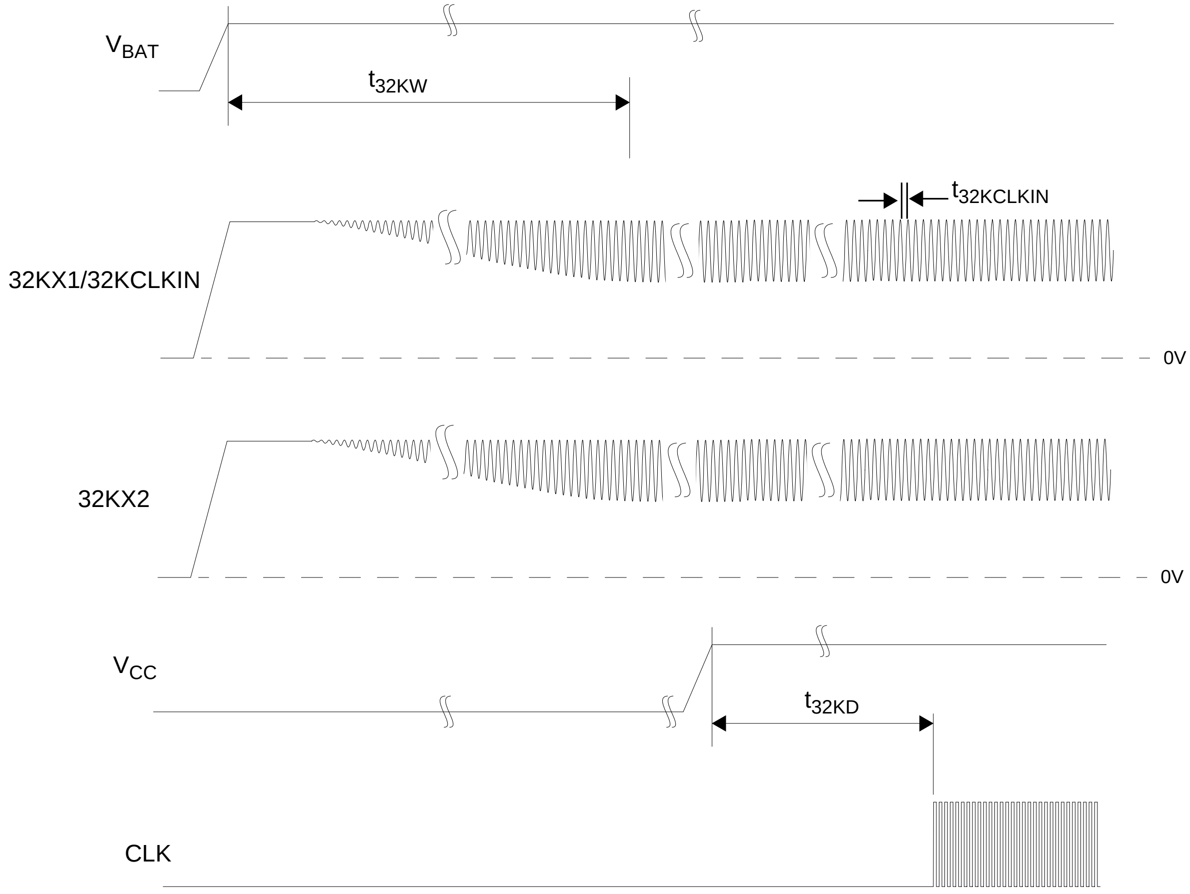


Figure 131. 32K Waveforms

7.0 Device Specifications (Continued)

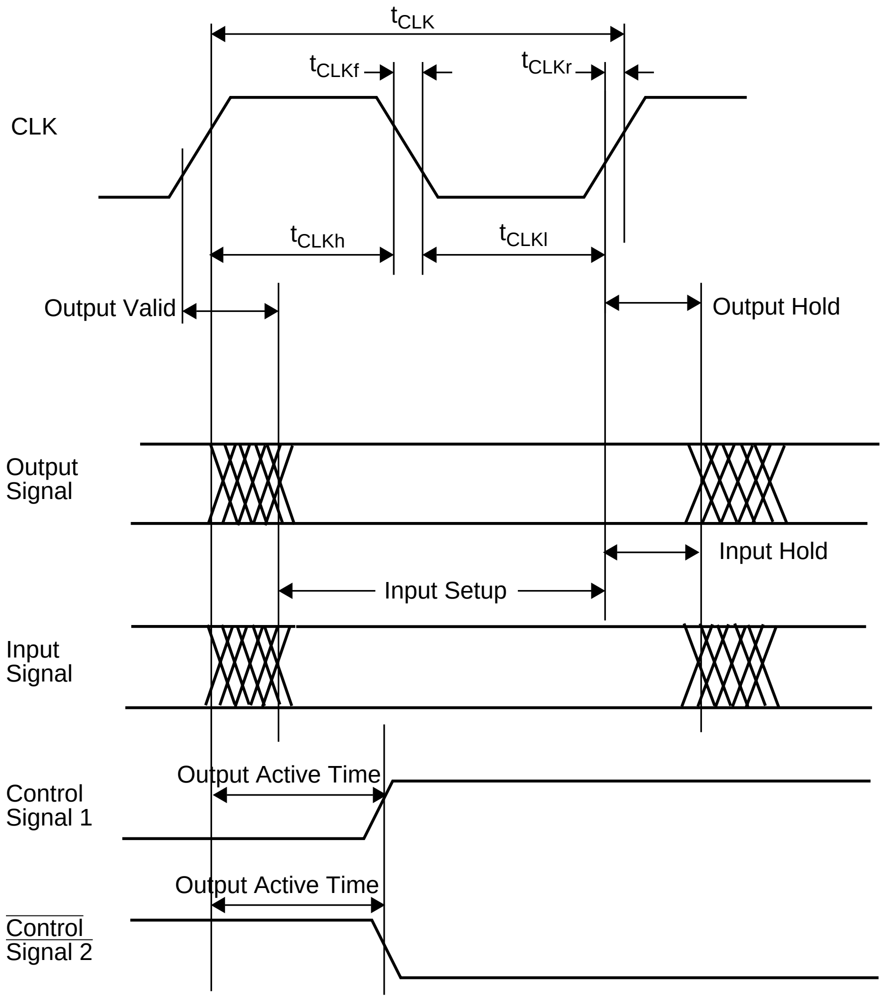


Figure 132. Clock Waveforms

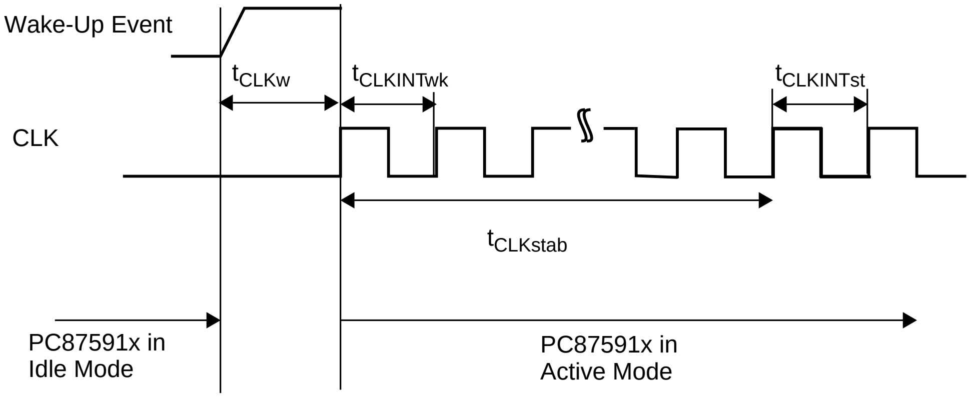


Figure 133. Internal Clock Generator

7.0 Device Specifications (Continued)

7.6.4 BIU Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
BIU Input Timing						
t_1	134, 136 to 138	Input setup time D0-15	Before RE CLK	15		ns
t_2	134, 136 to 138	Input hold time D0-15	After RE CLK	0		ns
BIU Output Timing						
t_3	134 to 139	Output valid time A0-17, BE0,1CBRD, D0-15	After RE CLK		14	ns
t_4	134 to 139	Output valid time $\overline{\text{BST0-2}}$	After RE CLK		$0.5 * t_{CLK}$ + 14 ns	-
t_5	134 to 138	Output active/inactive time $\overline{\text{RD}}$, $\overline{\text{SEL0-1}}$, $\overline{\text{SELIO}}$	After RE CLK		14	ns
t_6	134, 135	Output active/inactive time $\overline{\text{WR}}$0-1	After RE CLK		$0.5 * t_{CLK}$ + 14 ns	-
t_7	136	Minimum inactive time $\overline{\text{RD}}$	After RE $\overline{\text{RD}}$	t_{CLK} – 5 ns		-
t_8	134, 135	Output float time A0-17, D0-15,$\overline{\text{RD}}$, $\overline{\text{SEL0-1}}$,$\overline{\text{SELIO}}$,$\overline{\text{WR0-1}}$	After RE CLK		14	ns
t_9	134, 135	Minimum delay time	From RE $\overline{\text{RD}}$ to D0-15 drive	t_{CLK} – 8 ns		-
t_{10}	135	Minimum delay time	From RE $\overline{\text{RD}}$ to RE $\overline{\text{SELn}}$	0		ns
t_{11}	135	Minimum delay time	From RE $\overline{\text{SELx}}$ to FE $\overline{\text{SELy}}$	0		ns
t_{12}	134 to 139	Output hold time A0-17, BE0-1, CBRD, D0-15, $\overline{\text{RD}}$, $\overline{\text{SEL0-1}}$, $\overline{\text{SELIO}}$	After RE CLK	0		ns
t_{13}	134 to 139	Output hold time $\overline{\text{BST0-2,WR0-1}}$	After RE CLK	$0.5 * t_{CLK}$ – 4 ns		-
t_{14}	135	D0-15 valid in late write bus cycles	Before RE $\overline{\text{WR0-1}}$	$(K + 0.5) * t_{CLK}$ – 8 ns		-

7.0 Device Specifications (Continued)

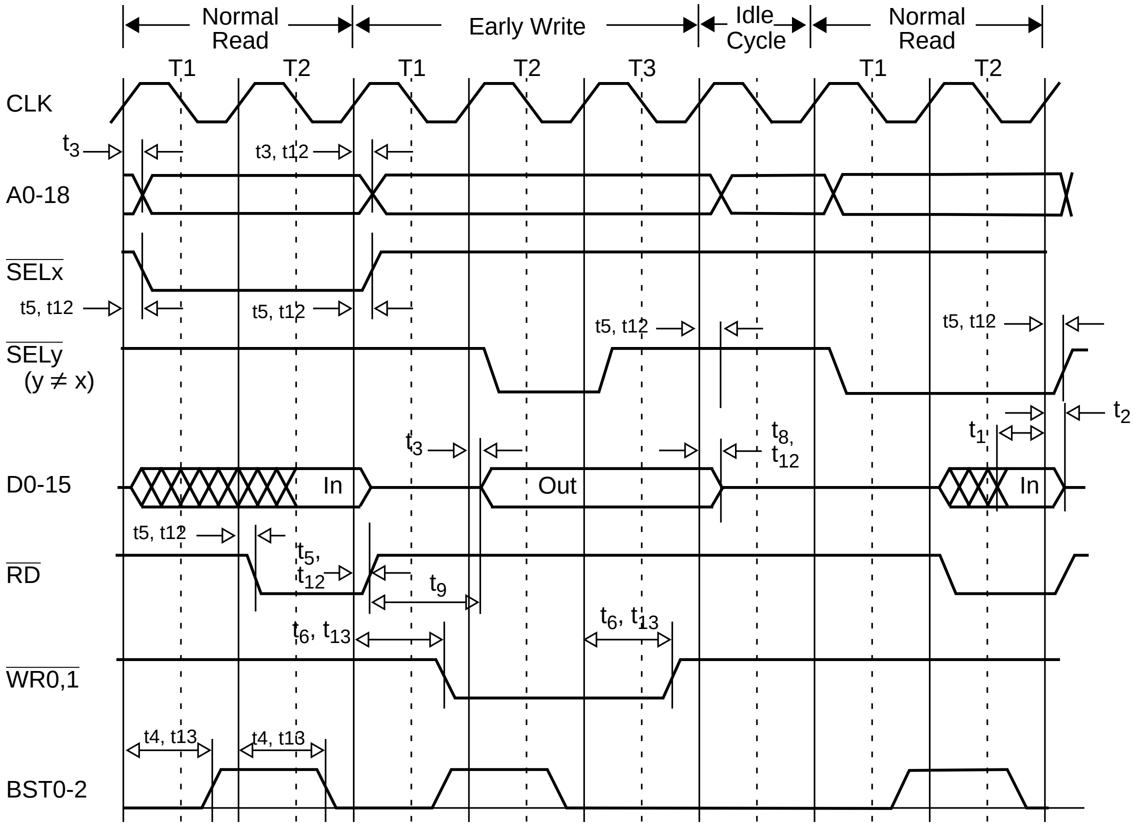


Figure 134. Early Write Between Normal Read Bus Cycles

7.0 Device Specifications (Continued)

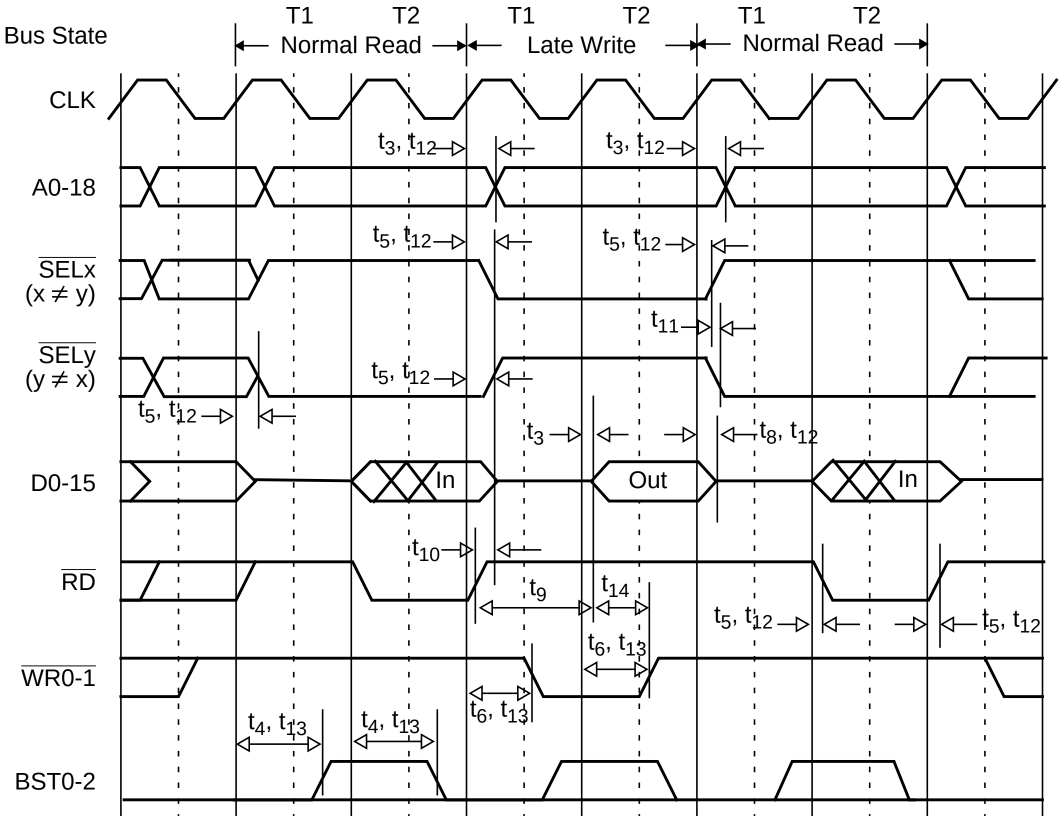


Figure 135. Late Write between Two Normal Read Bus Cycles, 0 Wait, AC Timing

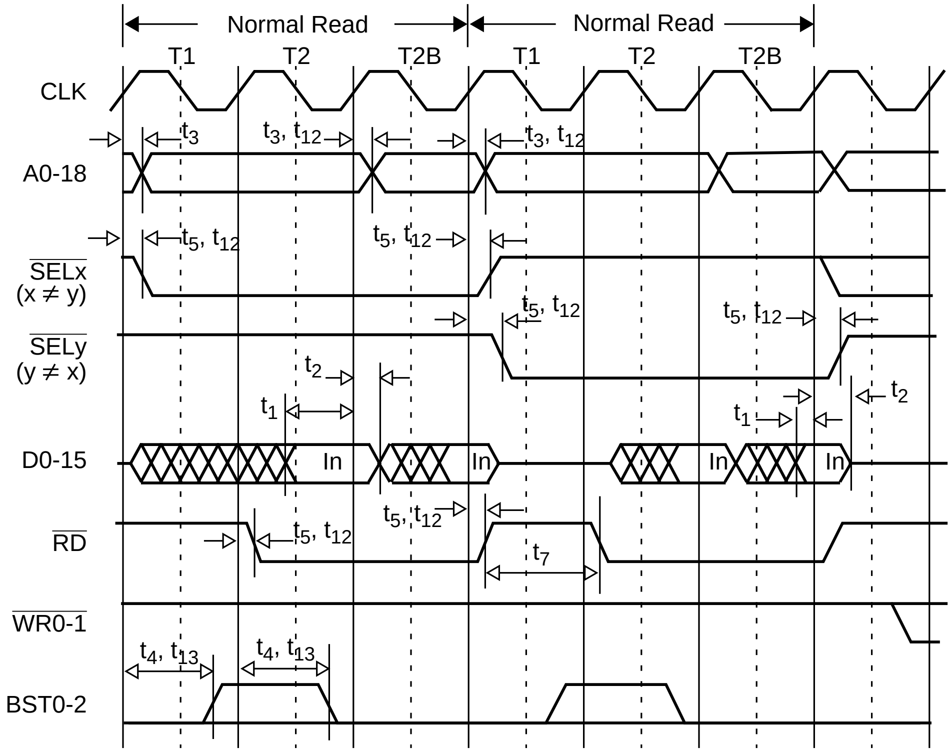


Figure 136. Two Consecutive Normal Read Bus Cycles with Burst, 0 Wait, AC Timing

7.0 Device Specifications (Continued)

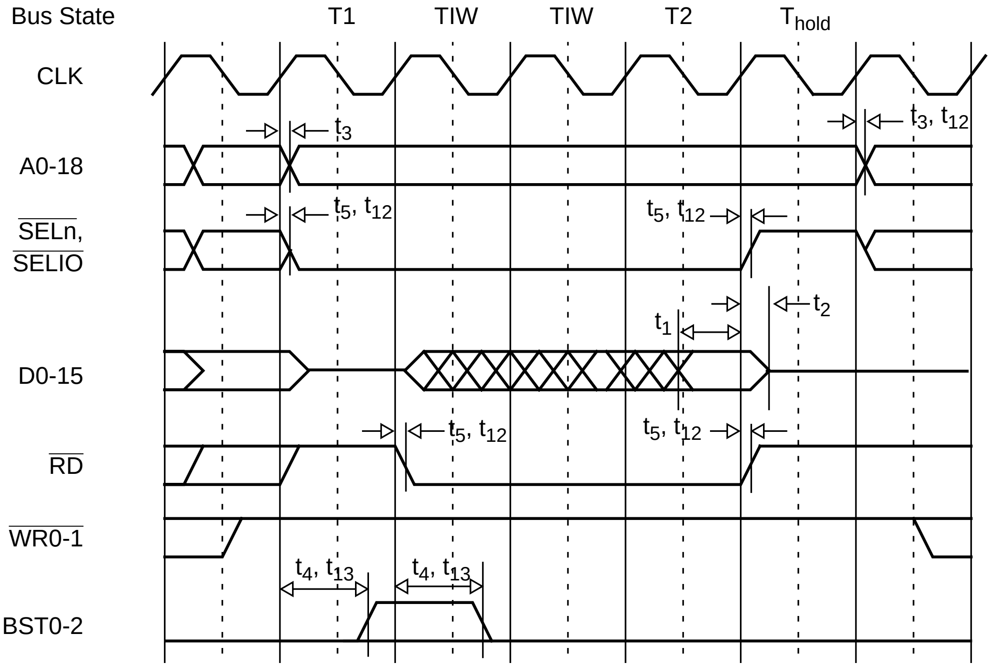


Figure 137. Normal Read Bus Cycle (2 Internal Waits, and 1 Hold), AC Timing

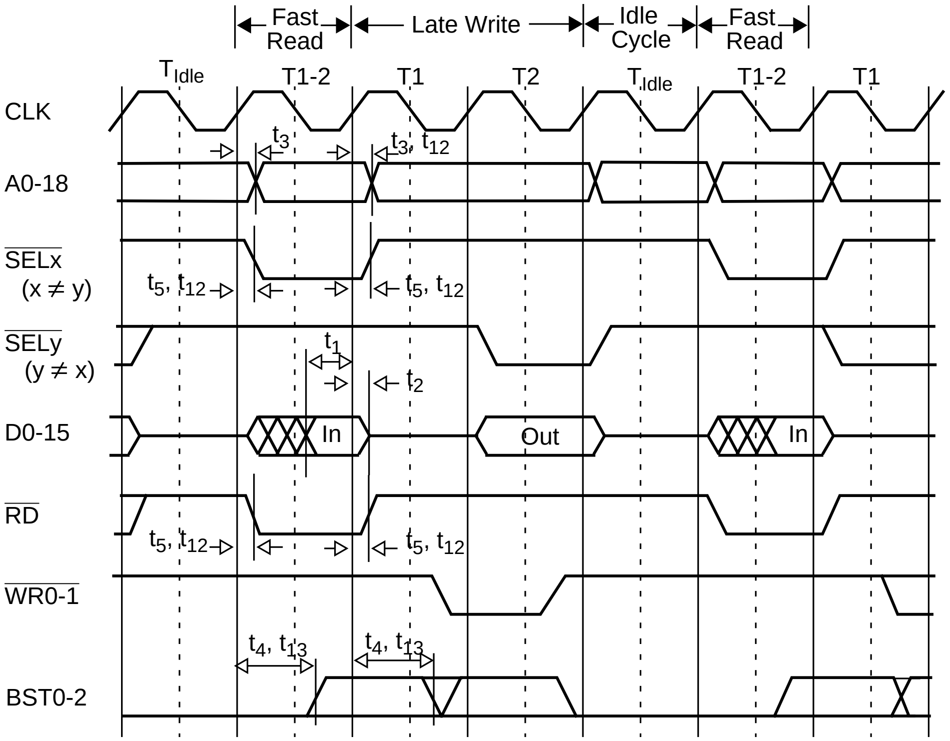


Figure 138. Fast Read Bus Cycle, AC Timing

7.0 Device Specifications (Continued)

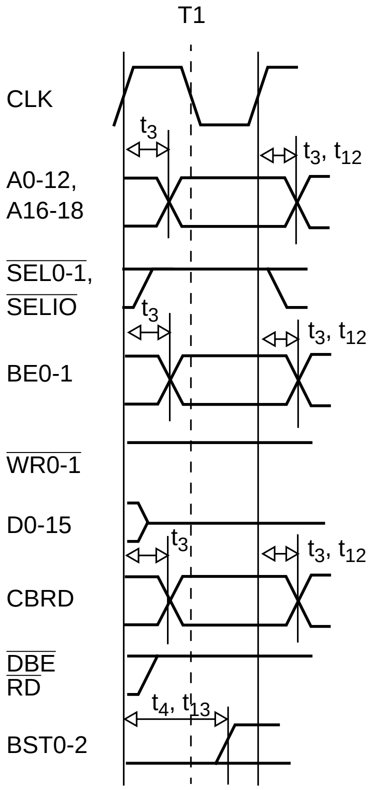


Figure 139. Core Bus Monitoring Bus Cycle, AC Timing

7.6.5 GPIO Ports Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
GPIO Input Timing						
t_{INPs}	140	Input setup time IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPE0-7, IOPF0-7, IOPG0-7, IOPH0-7, IOPI0-7, IOPJ0-7, IOPK0-7, IOPL0-4, IOPM0-7, KBSIN0-7[1]	Before RE CLK	$0.5 * t_{CLK}$		-
t_{INPh}	140	Input hold time IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPE0-7, IOPF0-7, IOPG0-7, IOPH0-7, IOPI0-7, IOPJ0-7, IOPK0-7, IOPL0-4, IOPM0-7, KBSIN0-7[1]	After RE CLK	0		ns
GPIO Output Timing						
t_{OUTv}	141	Output valid time KBSOUT0-15, IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPF0-7, IOPG0-7, IOPH0-7, IOPI0-7, IOPJ0-7, IOPK0-7, IOPL0-4, IOPM0-7	After RE CLK		$0.5 * t_{CLK}$	-
t_{OUTh}	141	Output hold time KBSOUT0-15, IOPA0-7, IOPB0-7, IOPC0-7, IOPD0-7, IOPF0-7, IOPG0-7, IOPH0-7, IOPI0-7, IOPJ0-7, IOPK0-7, IOPL0-4, IOPM0-7	After RE CLK	0		ns

[1] When using the Schmitt input.

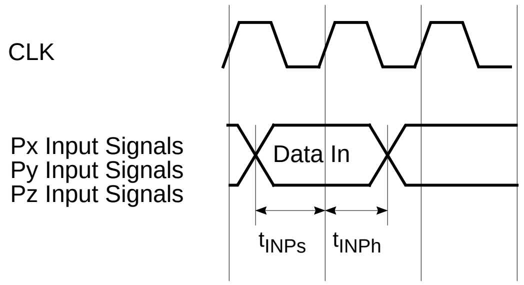


Figure 140. Input Signal Timing for Input and I/O Port Signals

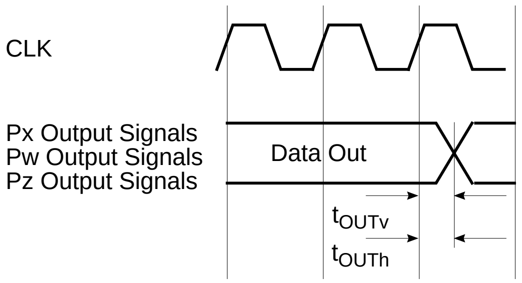


Figure 141. Output Signal Timing for Output and I/O Port Signals

7.0 Device Specifications (Continued)

7.6.6 PWM Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
t_{OUTv}	142	Output valid time	After RE CLK		0.5 * t_{CLK}	-
t_{OUTh}	142	Output hold time	After RE CLK	0		ns

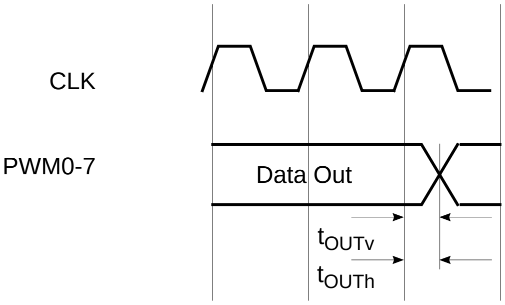


Figure 142. Output Signal Timing for PWM Signals

7.6.7 MSWC Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
t_{wupd}	143	Wake-up propagation delay	After FE $\overline{\text{RI2,1}}$		30	ns
t_L	143	$\overline{\text{RI2,1}}$ and $\overline{\text{RING}}$ Low Time		10		ns
t_H	143	$\overline{\text{RI2,1}}$ and $\overline{\text{RING}}$ High Time		10		ns

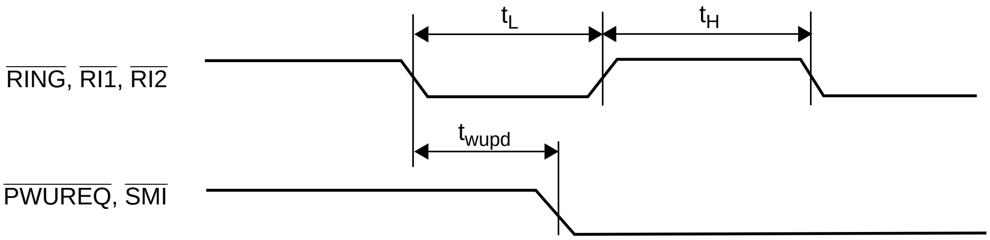

t_L
t_H
$\overline{\text{RING}}$, $\overline{\text{RI1}}$, $\overline{\text{RI2}}$
t_{wupd}
$\overline{\text{PWUREQ}}$, $\overline{\text{SMI}}$

Figure 143. Wake-Up Timing

7.0 Device Specifications (Continued)

7.6.8 PS/2 Interface Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
PS/2 Input Timing						
t_{PSDIs}	144	Input setup time PSDAT1-3	Before FE PSCLK1-3	0		ns
t_{PSDIh}	144	Input hold time PSDAT1-3	After RE PSCLK1-3	0		ns
t_{PSCLKl}	144	PSCLK1-3 low time	At 0.8V (Both Edges)	$(n+1)t_{CLK}$[1] ns		-
t_{PSCLKh}	144	PSCLK1-3 high time	At 2.0V (Both Edges)	$(n+1)t_{CLK}$[1] ns		-
PS/2 Output Timing						
t_{PSDOv}	145	Output valid time PSDAT1-4	After FE PSCLK1-4		$(n + 6) * t_{CLK}$ + 14[2] ns	-
t_{PSDOh}	145	Output hold time PSDAT1-4	After FE PSCLK1-4	0		ns
t_{PSCLKa}	146	Output active time PSCLK1-4	After RE CLK		17	ns
$t_{PSCLKia}$	146	Output inactive time PSCLK1-4	After RE CLK		17	ns

1. ‘n’ is the number of clock cycles, as programed in the IDB field. See “PS/2 Control Register (PSCON)” on page 130.
2. ‘n’ is defined in “PS/2 Control Register (PSCON)” on page 130.

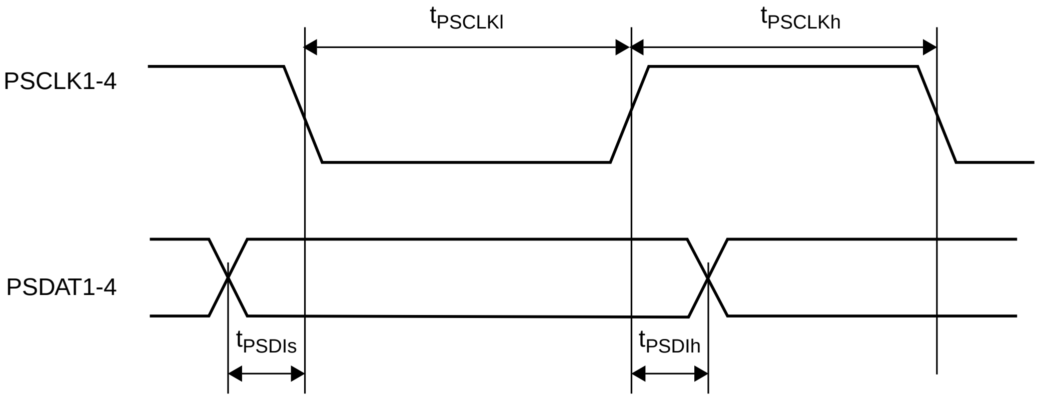


Figure 144. PS/2 Receive Timing

7.0 Device Specifications (Continued)

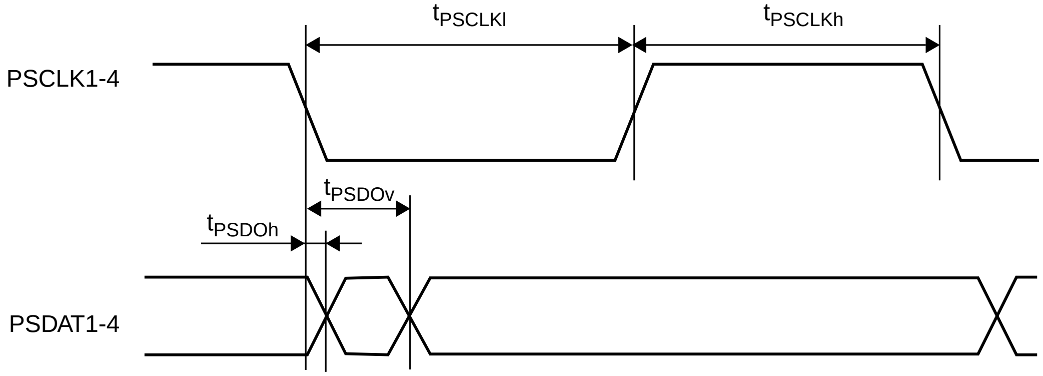


Figure 145. PS/2 Transmit Timing

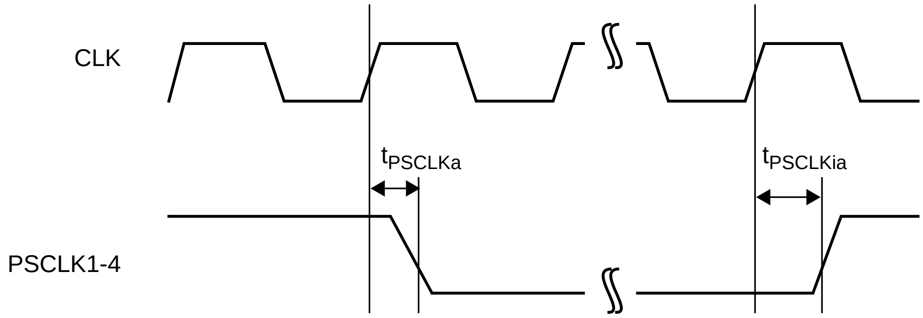


Figure 146. PS/2 Clock Signal Pulled Low by PC87591x

7.6.9 ACCESS.bus Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
ACB Input Timing						
t_{BUFi}	148	Bus free time between Stop and Start condition		$t_{SCLhigho}$		-
t_{CSTOsi}	148	SCL setup time	Before Stop condition	$8 * t_{CLK} - t_{SCLri}$		-
t_{CSTRhi}	148, 149	SCL hold time	After Start condition	$8 * t_{CLK} - t_{SCLri}$		-
t_{CSTRsi}	149	SCL setup time	Before Start condition	$8 * t_{CLK} - t_{SCLri}$		-
t_{DHCsi}	149	Data high setup time	Before SCL RE	$2 * t_{CLK}$		-
t_{DLCsi}	148	Data low setup time	Before SCL RE	$2 * t_{CLK}$		-
t_{SCLfi}	147	SCL signal fall time			300	ns
t_{SCLri}	147	SCL signal rise time			1	µs
$t_{SCLlowi}$	150	SCL low time	After SCL FE	$16 * t_{CLK}$		-

7.0 Device Specifications (Continued)

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
$t_{SCLhighi}$	150	SCL high time	After SCL RE	$16 * t_{CLK}$		-
t_{SDAri}	147	SDA signal fall time			300	ns
t_{SDAfi}	147	SDA signal rise time			1	µs
t_{SDAhi}	150	SDA hold time	After SCL FE	0		ns
t_{SDAsi}	150	SDA setup time	Before SCL RE	$2 * t_{CLK}$		-
ACB Output Timing						
$t_{SCLhigho}$	150	SCL high time	After SCL RE	$K * t_{CLK} – 1$ µs		-
$t_{SCLlowo}$	150	SCL low time	After SCL FE	$K * t_{CLK} – 1$ µs		-
t_{BUFo}	148	Bus free time between Stop and Start condition		$t_{SCLhigho}$		-
t_{CSTOso}	148	SCL setup time	Before Stop condition	$t_{SCLhigho}$		-
t_{CSTRho}	148, 149	SCL hold time	After Start condition	$t_{SCLhigho}$		-
t_{CSTRso}	149	SCL setup time	Before Start condition	$t_{SCLhigho}$		-
t_{DHCso}	149	Data high setup time	Before SCL RE	$t_{SCLhigho} – t_{SDAro}$		-
t_{DLCso}	148	Data low setup time	Before SCL RE	$t_{SCLhigho} – t_{SDAfo}$		-
t_{SCLfo}	147	SDA signal fall time			300	ns
t_{SCLro}	147	SDA signal rise time			See note[1]	
t_{SDAfo}	147	SDA signal fall time			300	ns
t_{SDAro}	147	SDA signal rise time			See note[1]	-
t_{SDAho}	150	SDA hold time	After SCL FE	$7 * t_{CLK} – t_{SCLfo}$		-
t_{SDAvo}	150	SDA valid time	After SCL FE		$7 * t_{CLK} + t_{SDAro}$	-

1. Depends on the signal's capacitance and the pull-up value. Must be less than 1 µs.

7.0 Device Specifications (Continued)

In Figure 147 through Figure 150, an “o” is added to parameter names in the timing tables for output signals and an “i” for input signals, as displayed in the preceding table:

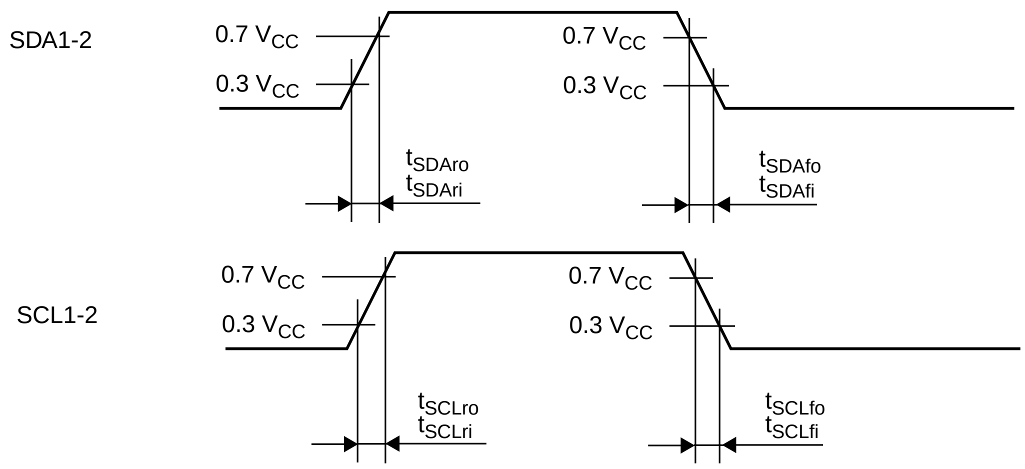


Figure 147. ACB Signals (SDA and SCL) Rising Time and Falling Time

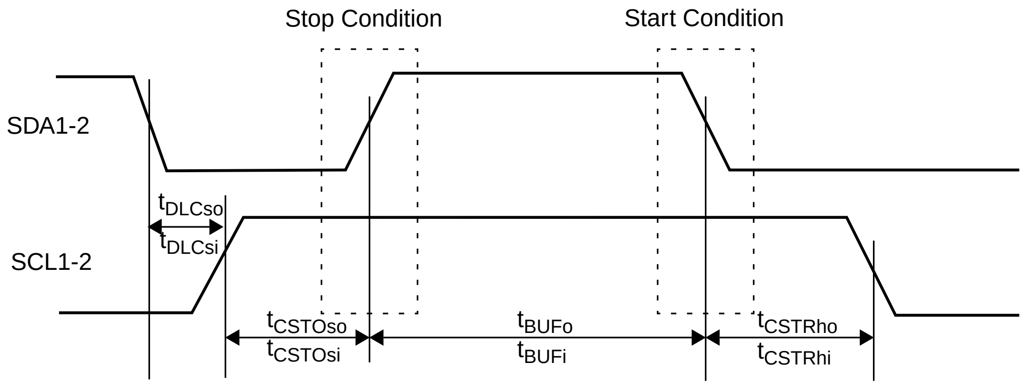


Figure 148. ACB Start and Stop Condition Timing

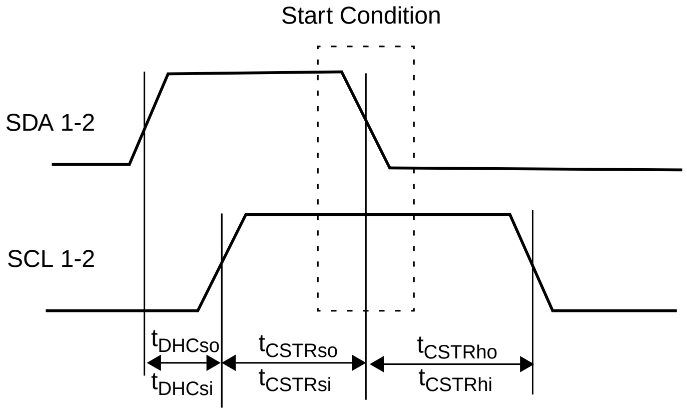


Figure 149. ACB Start Condition TIming

7.0 Device Specifications (Continued)

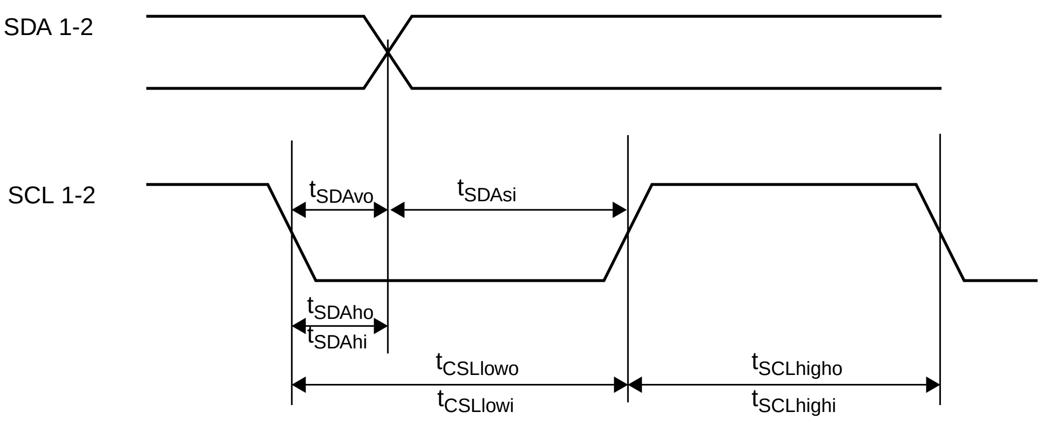


Figure 150. ACB Data Bit Timing

7.6.10 MFT16 Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
MFT16 Input Timing						
t_{TABH}	151	TA1-2/TB1-2 high time		t_{CLK} + 5 ns		-
t_{TABL}	151	TA1-2/TB1-2 low time		t_{CLK} + 5 ns		-

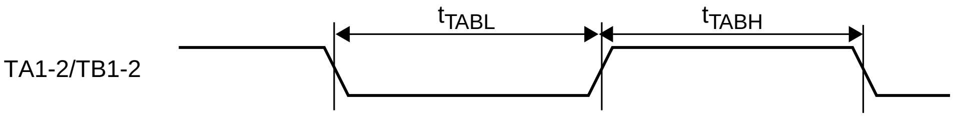


Figure 151. Multi-Function Timer (MFT16) Input Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
t_{OUTv}	152	Output valid time	After RE CLK		0.5 * t_{CLK}	-
t_{OUTh}	152	Output hold time	After RE CLK	0		ns

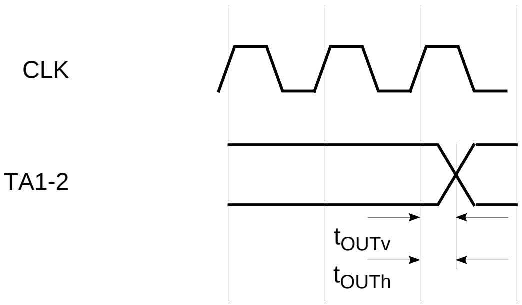


Figure 152. Multi-Function Timer (MFT16) Output Timing

7.0 Device Specifications (Continued)

7.6.11 ICU/Development Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
ICU/Development Input Timing						
$t_{\overline{\text{BRKL}}\text{s}}$	153	Input setup time for $\overline{\text{BRKL}}$	Before RE CLK	10		ns
$t_{\overline{\text{BRKL}}\text{h}}$	153	Input hold time for $\overline{\text{BRKL}}$	After RE CLK	0		ns
ICU/Development Output Timing						
t_{PFSh}	153	Output hold time PFS, PLI	After RE CLK	$0.5 * t_{\text{CLK}}$ – 6 ns		-
t_{PFSv}	153	Output active/inactive time PFS, PLI	After RE CLK		$0.5 * t_{\text{CLK}}$ + 12 ns	-
t_{RSTOd}	154	Output delay time $\overline{\text{RSTO}}$	After RE Internal Reset	$5 * t_{\text{CLK}}$		-
t_{RSTOw}	154	Output pulse width $\overline{\text{RSTO}}$	FE to RE $\overline{\text{RSTO}}$	$3 * t_{\text{CLK}}$		-
t_{RSTOs}	154	Output inactive time $\overline{\text{RSTO}}$	Before FE Internal Reset	0		ns

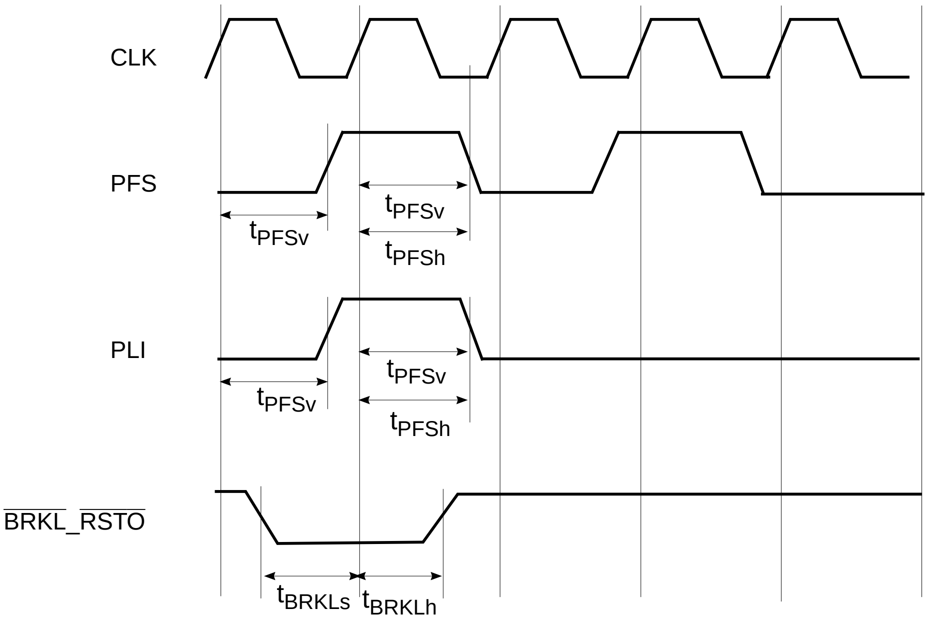


Figure 153. Pipe Status Signal (PFS and PLI) Timing

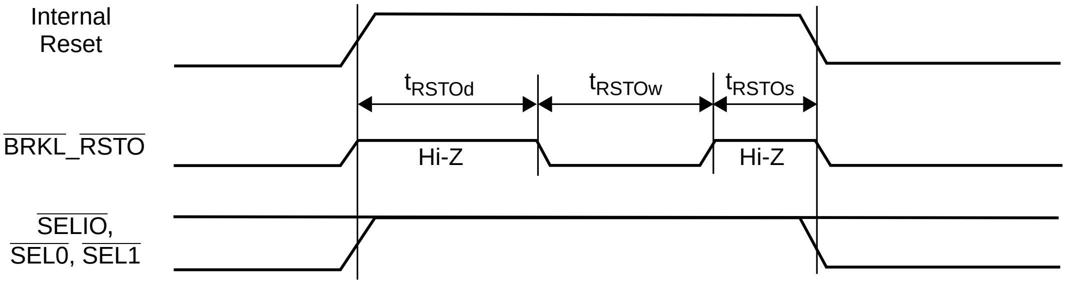

Internal Reset
t_{RSTOd}
t_{RSTOw}
t_{RSTOs}
$\overline{\text{BRKL}}$_$\overline{\text{RSTO}}$
Hi-Z
Hi-Z
$\overline{\text{SELIO}}$,
$\overline{\text{SEL0}}$, $\overline{\text{SEL1}}$

Figure 154. Reset Out Signal Timing

7.0 Device Specifications (Continued)

7.6.12 Asynchronous Edge Detected Signals Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
t_{asw}	155	Wake-up Input Width: EXWINT20-24, EXWINT(40,45-46) KBSIN0-7, $\overline{\text{PFAIL}}$ PSCLK1-4, PSDAT1-4, SWIN	Pulse width that guarantees detection on edge	15		ns
t_{is}	155	Input setup time: EXWINT20-24, EXWINT(40,45-46, $\overline{\text{PFAIL}}$, PSCLK1-4, PSDAT1-4, SWIN	See note[1]	10		ns
t_{ih}	155	Input hold time: EXWINT20-24, EXWINT(40,45-46), $\overline{\text{PFAIL}}$, PSCLK1-4, PSDAT1-4, SWIN	See note[1]	0		ns

[1] All wake-ups are asynchronous. Meeting the setup and hold are required for repeatability of the wake cycle only.

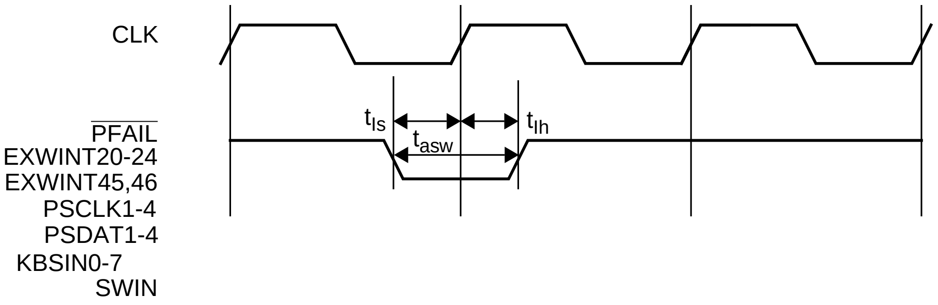


Figure 155. $\overline{\text{ISE}}$, $\overline{\text{PFAIL}}$, EXINTn and MIWU Input Signal Timing

7.0 Device Specifications (Continued)

7.6.13 Debugger Interface Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
Debugger Interface Input Signals						
t_S	156	TMS Setup Time	Before RE TCK	7		ns
t_H	156	TMS Hold Time	After RE TCK	2		ns
t_S	156	TDI Setup Time	Before RE TCK	1		ns
t_H	156	TDI Hold Time	After RE TCK	3.5		ns
t_{HW}	157	TCK High Pulse Width		15		ns
t_{LW}	157	TCK Low Pulse Width		5		ns
f_{MAX}		Maximum TCK Clock Frequency			20	MHz
T_{pu}	129	Wait Time Power-Up to TCK		t_{IRST}[1]		-
Debugger Interface Output Signals						
t_{PLH} t_{PLL}	158	Propagation Delay TCK to TDO	After FE TCK	3 3	22 22	ns
t_{PLZ} t_{PHZ}	160 159	Disable Time TCK to TDO		2 2	22 22	ns
t_{PZL} t_{PZH}	160 159	Enable Time TCK to TDO		3 3	22 22	ns
t_{TINTh}	161	$\overline{\text{TINT}}$ hld time	After FE TCK	10		ns
t_{TINTv}	161	$\overline{\text{TINT}}$ valid time	After RE CLK		25	ns

[1] See Section 7.6.2 on page 384.

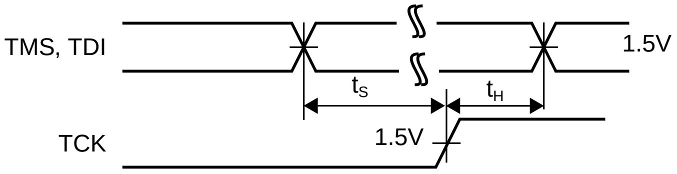


Figure 156. Debugger I/F Setup Time, Hold Time and Recovery Time

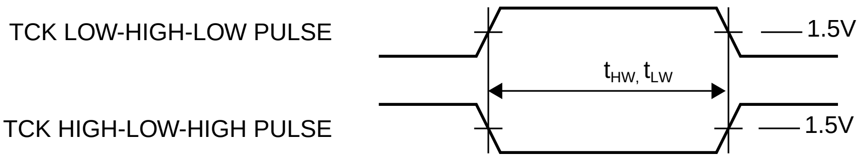


Figure 157. TCK Pulse Width

7.0 Device Specifications (Continued)

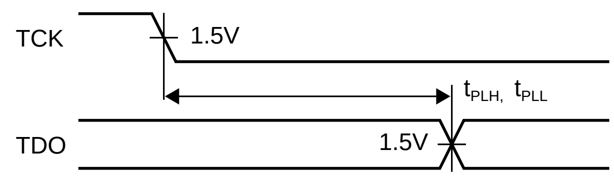


Figure 158. Debugger Interface Propagation Delay

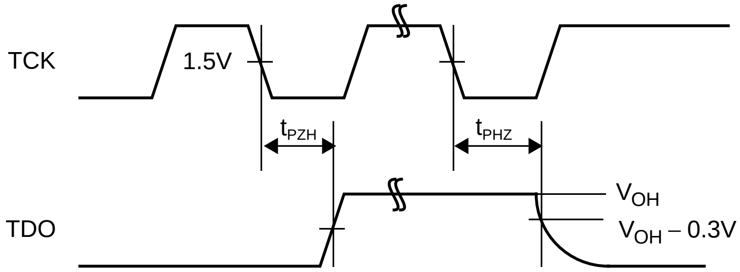


Figure 159. TDO TRI-STATE Output High Enable and Disable Times

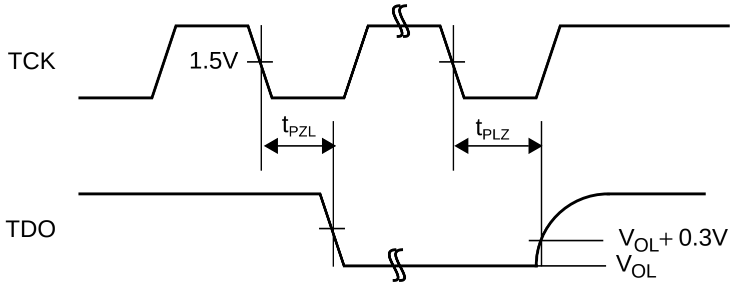


Figure 160. TDO TRI-STATE Output Low Enable and Disable Times

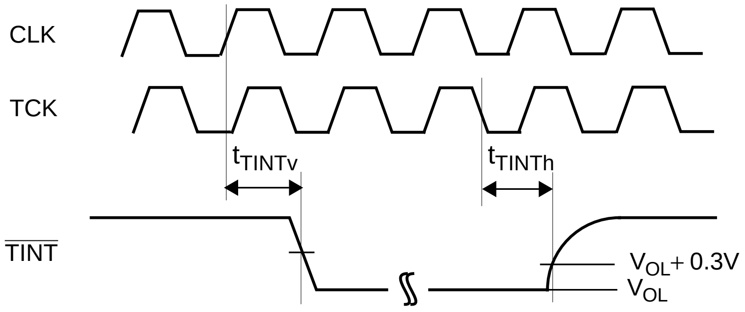


Figure 161. $\overline{\text{TINT}}$ Output Timing

7.0 Device Specifications (Continued)

7.6.14 USART Timing

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
t_r	162	CMOS output rise time.	Note [1]		6	ns
t_f	162	CMOS output fall time.	Note [1]		6	ns
t_{COh2}	163	Output hold (synchr. mode)	After RE USCLK	$-(2t_{CLK}$ +10 ns)		-
t_{COv2}	163	Output valid (synchr. mode)	After RE USCLK		$2t_{CLK}$ +10 ns	-
t_{COh}	162	Output hold (synchr. mode)	After RE USCLK	–20		ns
t_{Is}	162	Input setup time (asynchr. mode)	Before RE CLK	15		ns
t_{Ih}	162	Input hold time (asynchr. mode)	After RE CLK	0		ns
t_{CLKX}	162	USCLK input period (synchr. mode)		250		ns
t_{CLKh}	162	USCLK input low time.		$2t_{CLK}$		-
t_{CLKl}	162	USCLK input high time.		$2t_{CLK}$		-
t_{SIs}	163	Input Setup time (synchr. mode)	Before FE USCLK (in) in synchronous mode	t_{CLK} + 10 ns		-
t_{SIh}	163	Input Hold time (synchr. mode)	After FE USCLK (in) in synchronous mode	t_{CLK} + 10 ns		-
t_{SIs1}	164	Input Setup time (synchr. mode)	Before FE USCLK (out) in synchronous mode	20		ns
t_{SIh1}	164	Input Hold time (synchr. mode)	After FE USCLK (out) in synchronous mode	20		ns

[1] Not tested; guaranteed by characterization.

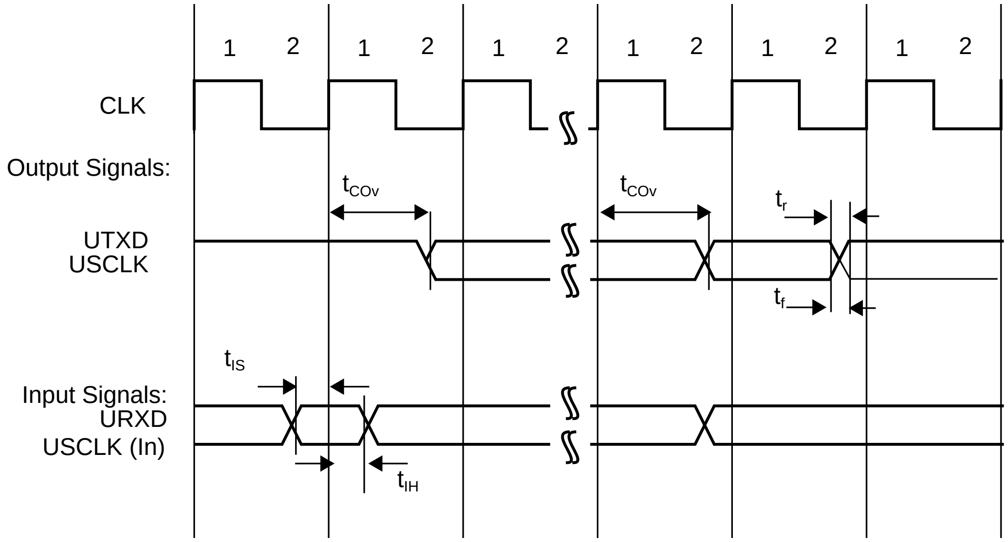


Figure 162. USART Asynchronous Mode Timing

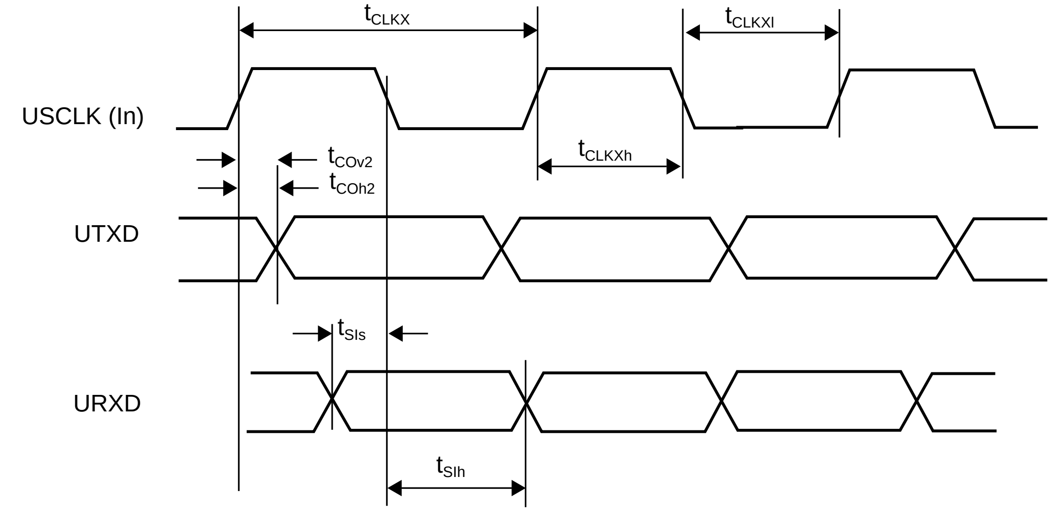


Figure 163. USART Synchronous Mode Timing (USCLK Input)

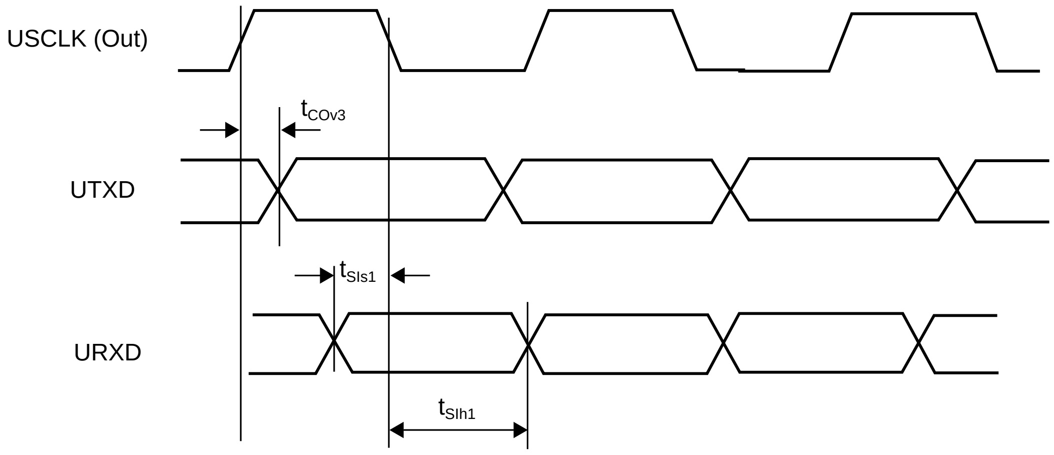


Figure 164. USART Synchronous Mode Timing (USCLK Out)

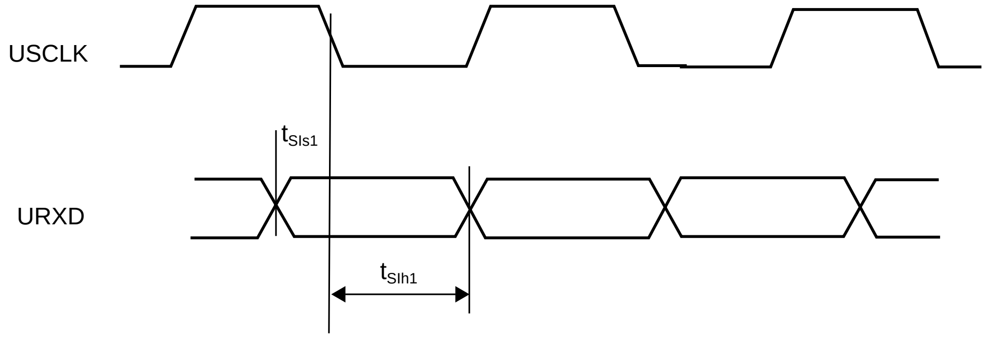


Figure 165. USART Synchronous Mode Timing (USCLK)

7.0 Device Specifications (Continued)

7.6.15 LCLK and $\overline{\text{RESET1-2}}$

Symbol	Parameter	Min	Max	Units
t_{CYC}[1]	LCLK Cycle Time	30		ns
t_{HIGH}	LCLK High Time	11		ns
t_{LOW}	LCLK Low Time	11		ns
-	LCLK Slew Rate[2]	1	4	V/ns
-	$\overline{\text{RESET1-2}}$ Slew Rate[3]	50		mV/ns

1. The PCI may have any clock frequency between nominal DC and 33 MHz. Device operational parameters at frequencies under 16 MHz may be guaranteed by design rather than by testing. The clock frequency may be changed at any time during the operation of the system as long as the clock edges remain “clean” (monotonic) and the minimum cycle and high and low times are not violated. The clock may only be stopped in a low state.
2. Rise and fall times are specified in terms of the edge rate measured in V/ns. This slew rate must be met across the minimum peak-to-peak portion of the clock wavering as shown below.
3. The minimum $\overline{\text{RESET1-2}}$ slew rate applies only to the rising (de-assertion) edge of the reset signal, and ensures that system noise cannot render an otherwise monotonic signal to appear to bounce in the switching range.

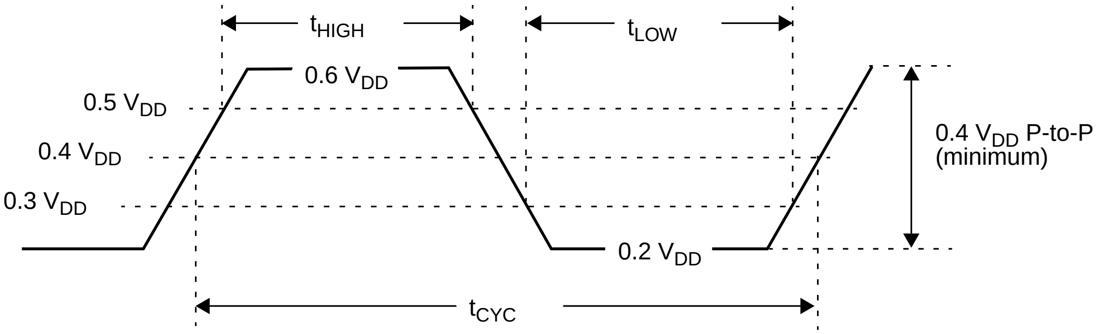


Figure 166. LCLK Waveform

7.0 Device Specifications (Continued)

7.6.16 LPC and SERIRQ Signals

Symbol	Figure	Description	Reference Conditions	Min	Max	Unit
t_{VAL}	167	Output Valid Delay	After RE CLK		11	ns
t_{ON}	167	Float to Active Delay	After RE CLK	2		ns
t_{OFF}	167	Active to Float Delay	After RE CLK		28	ns
t_{SU}	168	Input Setup Time	Before RE CLK	7		ns
t_{HI}	168	Input Hold Time	After RE CLK	0		ns

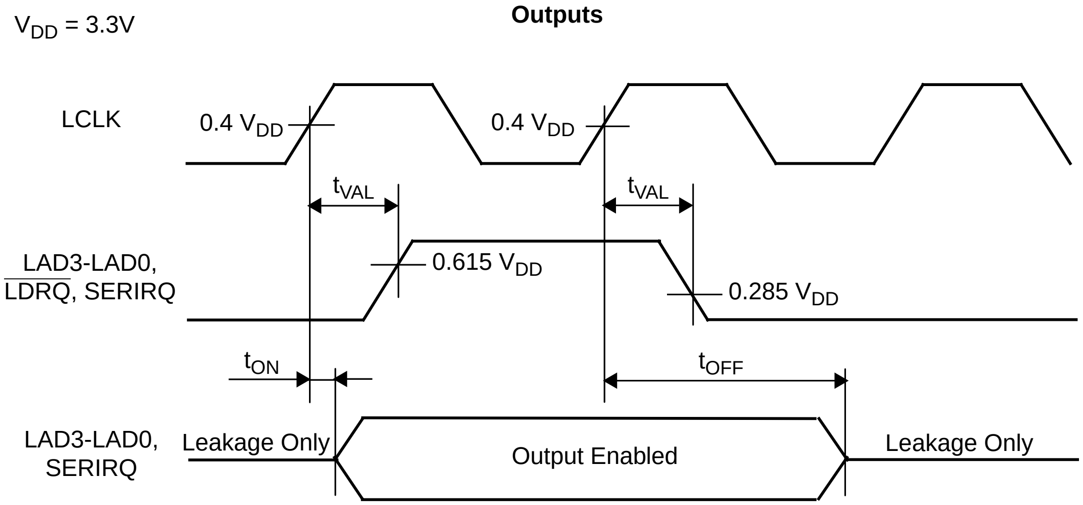


Figure 167. LPC/SERIRQ Interface Output Timing

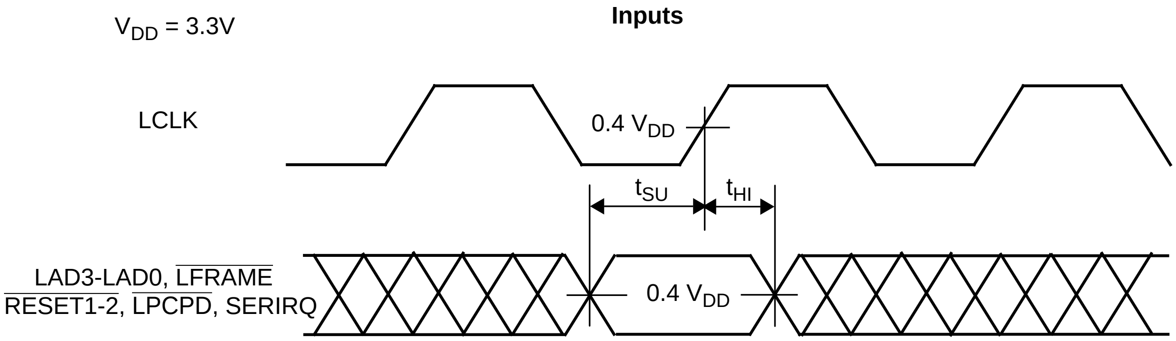

V_{DD} = 3.3V
Inputs
LCLK
0.4 V_{DD}
t_{SU}
t_{HI}
LAD3-LAD0, $\overline{LFRAME}$
$\overline{RESET1-2}$, $\overline{LPCPD}$, SERIRQ
0.4 V_{DD}

Figure 168. LPC/SERIRQ Interface Input Timing

-

A. Software for Hardware Interface

A.1 CORE DOMAIN REGISTER LIST

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments

A.1.1 Module Configuration

(Section 2.3 on page 53 and Section 2.4.2 on page 58)

	EICFG	Byte	00 FF00$_{16}$	Read/Write	00$_{16}$	
	IOEE1	Byte	00 FF02$_{16}$	Read/Write	00$_{16}$	
	IOEE2	Byte	00 FF04$_{16}$	Read/Write	00$_{16}$	
	MCFG	Byte	00 FF10$_{16}$	Read/Write	00$_{16}$ or 80$_{16}$	
	MCFGSH	Byte	00 FBFE$_{16}$	Write Only	MCFG shadow	
	STRPST	Byte	00 FF12$_{16}$	Read Only	According to external straps	

A.1.2 Bus Interface Unit (BIU)

(Section 4.1.10 on page 83)

	BCFG	Byte	00 F980$_{16}$	Read/Write	07$_{16}$	
	IOCFG	Word	00 F982$_{16}$	Read/Write	069F$_{16}$	
	SZCFG0	Word	00 F984$_{16}$	Read/Write	069F$_{16}$	
	SZCFG1	Word	00 F986$_{16}$	Read/Write	069F$_{16}$	
	SZCFG2	Word	00 F988$_{16}$	Read/Write	069F$_{16}$	

A.1.3 DMA Controller

(Section 4.2.8 on page 92)

	ADCA0	Double W.	00 FA00$_{16}$	Read/Write		
	ADRA0	Double W.	00 FA04$_{16}$	Read/Write		
	ADCB0	Double W.	00 FA08$_{16}$	Read/Write		
	ADRB0	Double W.	00 FA0C$_{16}$	Read/Write		
	BLTC0	Double W.	00 FA10$_{16}$	Read/Write		
	BLTR0	Double W.	00 FA14$_{16}$	Read/Write		
	CNTL0	Word	00 FA1C$_{16}$	Read/Write	0000$_{16}$	
	STAT0	Byte	00 FA1E$_{16}$	Read/Write	00$_{16}$	
	ADCA1	Double W.	00 FA20$_{16}$	Read/Write		
	ADRA1	Double W.	00 FA24$_{16}$	Read/Write		
	ADCB1	Double W.	00 FA28$_{16}$	Read/Write		

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	ADRB1	Double W.	00 FA2C$_{16}$	Read/Write		
	BLTC1	Double W.	00 FA30$_{16}$	Read/Write		
	BLTR1	Double W.	00 FA34$_{16}$	Read/Write		
	CNTL1	Word	00 FA3C$_{16}$	Read/Write	0000$_{16}$	
	STAT1	Byte	00 FA3E$_{16}$	Read Only	00$_{16}$	
	ADCA2	Double W.	00 FA40$_{16}$	Read/Write		
	ADRA2	Double W.	00 FA44$_{16}$	Read/Write		
	ADCB2	Double W.	00 FA48$_{16}$	Read/Write		
	ADRB2	Double W.	00 FA4C$_{16}$	Read/Write		
	BLTC2	Double W.	00 FA50$_{16}$	Read/Write		
	BLTR2	Double W.	00 FA54$_{16}$	Read/Write		
	CNTL2	Word	00 FA5C$_{16}$	Read/Write	0000$_{16}$	
	STAT2	Byte	00 FA5E$_{16}$	Read Only	00$_{16}$	
	ADCA3	Double W.	00 FA60$_{16}$	Read/Write		
	ADRA3	Double W.	00 FA64$_{16}$	Read/Write		
	ADCB3	Double W.	00 FA68$_{16}$	Read/Write		
	ADRB3	Double W.	00 FA6C$_{16}$	Read/Write		
	BLTC3	Double W.	00 FA70$_{16}$	Read/Write		
	BLTR3	Double W.	00 FA74$_{16}$	Read/Write		
	CNTL3	Word	00 FA7C$_{16}$	Read/Write	0000$_{16}$	
	STAT3	Byte	00 FA7E$_{16}$	Read Only	00$_{16}$	

A.1.4 General-Purpose I/O (GPIO) Ports

(Section 4.5.6 on page 119)

	PADIR	Byte	00 FE20$_{16}$	Read/Write	00$_{16}$	
	PADIN	Byte	00 FE22$_{16}$	Read Only		
	PADOUT	Byte	00 FE24$_{16}$	Read/Write		
	PAWPU	Byte	00 FE26$_{16}$	Read/Write	00$_{16}$	
	PAALT	Byte	00 FE28$_{16}$	Read/Write	00$_{16}$	
	PBDIR	Byte	00 FE2A$_{16}$	Read/Write	20$_{16}$	
	PBDIN	Byte	00 FE2C$_{16}$	Read Only		
	PBDOUT	Byte	00 FE2E$_{16}$	Read/Write	Bit 5 is 1; the others are undefined	
	PBWPU	Byte	00 FE30$_{16}$	Read/Write	00$_{16}$	

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	PBALT	Byte	$00\ \mathrm{FE32}_{16}$	Read/Write	40_{16}	
	PCDIR	Byte	$00\ \mathrm{FE34}_{16}$	Read/Write	00_{16}	PC0 is reset on V_{CC} Power-up and WATCHDOG reset only.
	PCDIN	Byte	$00\ \mathrm{FE36}_{16}$	Read Only		
	PCDOUT	Byte	$00\ \mathrm{FE38}_{16}$	Read/Write		
	PCWPU	Byte	$00\ \mathrm{FE3A}_{16}$	Read/Write	00_{16}	
	PCALT	Byte	$00\ \mathrm{FE3C}_{16}$	Read/Write	00_{16}	
	PDDIR	Byte	$00\ \mathrm{FE3E}_{16}$	Read/Write	00_{16}	
	PDDIN	Byte	$00\ \mathrm{FE40}_{16}$	Read Only		
	PDDOUT	Byte	$00\ \mathrm{FE42}_{16}$	Read/Write		
	PDWPU	Byte	$00\ \mathrm{FE44}_{16}$	Read/Write	00_{16}	
	PDALT	Byte	$00\ \mathrm{FE46}_{16}$	Read/Write	00_{16}	
	PEDIN	Byte	$00\ \mathrm{FE48}_{16}$	Read Only		
	PEWPU	Byte	$00\ \mathrm{FE4A}_{16}$	Read/Write	00_{16}	
	PEALT	Byte	$00\ \mathrm{FE4C}_{16}$	Read/Write	00_{16}	
	KBSIN	Byte	$00\ \mathrm{FE4E}_{16}$	Read Only		
	KBSINPU	Byte	$00\ \mathrm{FE50}_{16}$	Read/Write		
	KBSOUT	Word	$00\ \mathrm{FE52}_{16}$	Read/Write	FFFF_{16}	
	PFDIR	Byte	$00\ \mathrm{FE54}_{16}$	Read/Write	00_{16}	
	PFDIN	Byte	$00\ \mathrm{FE56}_{16}$	Read Only		
	PFDOUT	Byte	$00\ \mathrm{FE58}_{16}$	Read/Write		
	PFWPU	Byte	$00\ \mathrm{FE5A}_{16}$	Read/Write	00_{16}	
	PFALT	Byte	$00\ \mathrm{FE5C}_{16}$	Read/Write	00_{16}	
	PHDIR	Byte	$00\ \mathrm{FB00}_{16}$	Read/Write	00_{16}	
	PHDIN	Byte	$00\ \mathrm{FB02}_{16}$	Read Only		
	PHDOUT	Byte	$00\ \mathrm{FB04}_{16}$	Read/Write		
	PIDIR	Byte	$00\ \mathrm{FB06}_{16}$	Read/Write	00_{16}	
	PIDIN	Byte	$00\ \mathrm{FB08}_{16}$	Read Only		
	PIDOUT	Byte	$00\ \mathrm{FB0A}_{16}$	Read/Write		
	PJDIR	Byte	$00\ \mathrm{FB0C}_{16}$	Read/Write	00_{16}	
	PJDIN	Byte	$00\ \mathrm{FB0E}_{16}$	Read Only		
	PJDOUT	Byte	$00\ \mathrm{FB10}_{16}$	Read/Write		
	PKDIR	Byte	$00\ \mathrm{FB12}_{16}$	Read/Write	00_{16}	
	PKDIN	Byte	$00\ \mathrm{FB14}_{16}$	Read Only		
	PKDOUT	Byte	$00\ \mathrm{FB16}_{16}$	Read/Write		

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	PLDIR	Byte	00 FB18$_{16}$	Read/Write	00$_{16}$	
	PLDIN	Byte	00 FB1A$_{16}$	Read Only		
	PLDOUT	Byte	00 FB1C$_{16}$	Read/Write		
	PMDIR	Byte	00 FB1E$_{16}$	Read/Write	00$_{16}$	
	PMDIN	Byte	00 FB20$_{16}$	Read Only		
	PMDOUT	Byte	00 FB22$_{16}$	Read/Write		

A.1.5 PS/2 Ports

(Section 4.6.5 on page 128)

	PSDAT	Byte	00 FE80$_{16}$	Read/Write		
	PSTAT	Byte	00 FE82$_{16}$	Read Only	00$_{16}$	
	PSCON	Byte	00 FE84$_{16}$	Read/Write	00$_{16}$	
	PSOSIG	Byte	00 FE86$_{16}$	Read/Write	47$_{16}$	
	PSISIG	Byte	00 FE88$_{16}$	Read Only		
	PSIEN	Byte	00 FE8A$_{16}$	Read/Write	00$_{16}$	

A.1.6 Host Interface (KBC, PM1 and PM2Channels)

(Section 5.1.4 on page 282 and Section 5.2.3 on page 290)

	HICTRL	Byte	00 FEA0$_{16}$	Read/Write	00$_{16}$	
	HIIRQC	Byte	00 FEA2$_{16}$	Read/Write	07$_{16}$	
	HIKMST	Byte	00 FEA4$_{16}$	Read/Write	00$_{16}$	
	HIKDO	Byte	00 FEA6$_{16}$	Write Only		
	HIMDO	Byte	00 FEA8$_{16}$	Write Only		
	HIKMDI	Byte	00 FEAA$_{16}$	Read Only		
	HIPM1ST	Byte	00 FEAC$_{16}$	Varies per bit	00$_{16}$	
	HIPM1DO	Byte	00 FEAE$_{16}$	Write Only		
	HIPM1DI	Byte	00 FEB0$_{16}$	Read Only		
	HIPM1DOC	Byte	00 FEB2$_{16}$	Write Only		
	HIPM1DOM	Byte	00 FEB4$_{16}$	Write Only		
	HIPM1DIC	Byte	00 FEB6$_{16}$	Read Only		
	HIPM1CTL	Byte	00 FEB8$_{16}$	Read/Write	40$_{16}$	
	HIPM1IC	Byte	00 FEBA$_{16}$	Read/Write	41$_{16}$	
	HIPM1IE	Byte	00 FEBC$_{16}$	Read/Write	00$_{16}$	
	HIPM2ST	Byte	00 FEBE$_{16}$	Varies per bit	00$_{16}$	

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	HIPM2DO	Byte	00 FEC0$_{16}$	Write Only		
	HIPM2DI	Byte	00 FEC2$_{16}$	Read Only		
	HIPM2DOC	Byte	00 FEC4$_{16}$	Write Only		
	HIPM2DOM	Byte	00 FEC6$_{16}$	Write Only		
	HIPM2DIC	Byte	00 FEC8$_{16}$	Read Only		
	HIPM2CTL	Byte	00 FECA$_{16}$	Read/Write	C0$_{16}$	
	HIPM2IC	Byte	00 FECC$_{16}$	Read/Write	41$_{16}$	
	HIPM2IE	Byte	00 FECE$_{16}$	Read/Write	00$_{16}$	

A.1.7 Multi-Function Timer (MTF16) 1

(Section 4.7.7 on page 142)

	T1CNT1	Word	00 FD80$_{16}$	Read/Write		
	T1CRA	Word	00 FD82$_{16}$	Read/Write		
	T1CRB	Word	00 FD84$_{16}$	Read/Write		
	T1CNT2	Word	00 FD86$_{16}$	Read/Write		
	T1PRSC	Byte	00 FD88$_{16}$	Read/Write	00$_{16}$	
	T1CKC	Byte	00 FD8A$_{16}$	Read/Write	00$_{16}$	
	T1CTRL	Byte	00 FD8C$_{16}$	Read/Write	00$_{16}$	
	T1ICTL	Byte	00 FD8E$_{16}$	Read/Write	00$_{16}$	
	T1ICLR	Byte	00 FD90$_{16}$	Write Only		

A.1.8 Multi-Function Timer (MFT16) 2

(Section 4.7.7 on page 142)

	T2CNT1	Word	00 FDA0$_{16}$	Read/Write		
	T2CRA	Word	00 FDA2$_{16}$	Read/Write		
	T2CRB	Word	00 FDA4$_{16}$	Read/Write		
	T2CNT2	Word	00 FDA6$_{16}$	Read/Write		
	T2PRSC	Byte	00 FDA8$_{16}$	Read/Write	00$_{16}$	
	T2CKC	Byte	00 FDAA$_{16}$	Read/Write	00$_{16}$	
	T2CTRL	Byte	00 FDAC$_{16}$	Read/Write	00$_{16}$	
	T2ICTL	Byte	00 FDAE$_{16}$	Read/Write	00$_{16}$	
	T2ICLR	Byte	00 FDB0$_{16}$	Write Only		

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
A.1.9 Timing and WATCHDOG (TWD) (Section 4.10.3 on page 166)						
	TWCFG	Byte	00 FEE0$_{16}$	Read/Write	00$_{16}$	
	TWCP	Byte	00 FEE2$_{16}$	Read/Write	00$_{16}$	
	TWDT0	Word	00 FEE4$_{16}$	Read/Write	FFFF$_{16}$	
	T0CSR	Byte	00 FEE6$_{16}$	Read/Write	00$_{16}$	
	WDCNT	Byte	00 FEE8$_{16}$	Write Only	0F$_{16}$	
	WDSDM	Byte	00 FEEA$_{16}$	Write Only		Write 5C$_{16}$
A.1.10 Analog to Digital Converter (ADC) (Section 4.11.6 on page 177)						
	ADCSTS	Byte	00 FF20$_{16}$	Varies per bit	00$_{16}$	Bit 2 is reset on V_{CC} power up reset only
	ADCCNF	Byte	00 FF22$_{16}$	Read/Write	00$_{16}$	
	ACLKCTL	Byte	00 FF24$_{16}$	Read/Write	3F$_{16}$	
	ADLYCTL	Byte	00 FF26$_{16}$	Read/Write	A7$_{16}$	
	TLOCOTL	Byte	00 FF28$_{16}$	Read/Write	7F$_{16}$	
	ADCPINX	Byte	00 FF2A$_{16}$	Read/Write	00$_{16}$	
	ADCPD	Word	00 FF2C$_{16}$	Read/Write		
	TCHANCTL	Byte	00 FF30$_{16}$	Varies per bit	10$_{16}$	
	TCHANDAT	Byte	00 FF32$_{16}$	Read Only		
	VCHN1CTL	Byte	00 FF34$_{16}$	Varies per bit	1F$_{16}$	
	VCHN1DAT	Word	00 FF36$_{16}$	Read Only		
	VCHN2CTL	Byte	00 FF38$_{16}$	Varies per bit	1F$_{16}$	
	VCHN2DAT	Word	00 FF3A$_{16}$	Read Only		
	VCHN3CTL	Byte	00 FF3C$_{16}$	Varies per bit	1F$_{16}$	
	VCHN3DAT	Word	00 FF3E$_{16}$	Read Only		
A.1.11 Digital to Analog Converter (DAC) (Section 4.12.5 on page 193)						
	DACCTRL	Byte	00 FF40$_{16}$	Read/Write	00$_{16}$	
	DACDAT0	Byte	00 FF42$_{16}$	Read/Write		
	DACDAT1	Byte	00 FF44$_{16}$	Read/Write		

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	DACDAT2	Byte	00 FF46$_{16}$	Read/Write		
	DACDAT3	Byte	00 FF48$_{16}$	Read/Write		

A.1.12 ACCESS.bus Interface (ACB) 1

(Section 4.13.8 on page 202)

	ACB1SDA	Byte	00 FF60$_{16}$	Read/Write		
	ACB1ST	Byte	00 FF62$_{16}$	Varies per bit	00$_{16}$	
	ACB1CST	Byte	00 FF64$_{16}$	Varies per bit	00$_{16}$	
	ACB1CTL1	Byte	00 FF66$_{16}$	Read/Write	00$_{16}$	
	ACB1ADDR	Byte	00 FF68$_{16}$	Read/Write		
	ACB1CTL2	Byte	00 FF6A$_{16}$	Read/Write	00$_{16}$	

A.1.13 ACCESS.bus Interface (ACB) 2

(Section 4.13.8 on page 202)

	ACB2SDA	Byte	00 FFE0$_{16}$	Read/Write		
	ACB2ST	Byte	00 FFE2$_{16}$	Varies per bit	00$_{16}$	
	ACB2CST	Byte	00 FFE4$_{16}$	Varies per bit	00$_{16}$	
	ACB2CTL1	Byte	00 FFE6$_{16}$	Read/Write	00$_{16}$	
	ACB2ADDR	Byte	00 FFE8$_{16}$	Read/Write		
	ACB2CTL2	Byte	00 FFEA$_{16}$	Read/Write	00$_{16}$	

A.1.14 Analog Comparators Monitor (ACM)

(Section 4.14.5 on page 212)

	ACMCTS	Byte	00 FD40$_{16}$	Varies per bit	00$_{16}$	
	ACMCNF	Byte	00 FD42$_{16}$	Read/Write	00$_{16}$	
	ACMTIM	Byte	00 FD44$_{16}$	Read/Write	37$_{16}$	
	THRDAT	Byte	00 FD46$_{16}$	Read/Write	00$_{16}$	
	CMPRES	Byte	00 FD48$_{16}$	Read Only		
	VOLDAT0	Byte	00 FD50$_{16}$	Read Only		
	VOLDAT1	Byte	00 FD52$_{16}$	Read Only		
	VOLDAT2	Byte	00 FD54$_{16}$	Read Only		
	VOLDAT3	Byte	00 FD56$_{16}$	Read Only		
	VOLDAT4	Byte	00 FD58$_{16}$	Read Only		
	VOLDAT5	Byte	00 FD5A$_{16}$	Read Only		

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	VOLDAT6	Byte	00 FD5C$_{16}$	Read Only		
	VOLDAT7	Byte	00 FD5E$_{16}$	Read Only		

A.1.15 Power Management (PM)

(Section 4.17.4 on page 237)

	PMCSR	Byte	00 FF80$_{16}$	Read/Write	00$_{16}$	

A.1.16 High Frequency Clock Generator (HFCG)

(Section 4.18.7 on page 243)

	HFCGCTRL1	Byte	00 FFA0$_{16}$	Varies per bit	0C$_{16}$	
	HFCGML	Byte	00 FFA2$_{16}$	Read/Write	CF$_{16}$	
	HFCGMH	Byte	00 FFA4$_{16}$	Read/Write	03$_{16}$	
	HFCGN	Byte	00 FFA6$_{16}$	Read/Write	08$_{16}$	
	HFCGIL	Byte	00 FFA8$_{16}$	Read/Write		
	HFCGIH	Byte	00 FFAA$_{16}$	Read/Write		
	HFCGP	Byte	00 FFAC$_{16}$	Read/Write	17$_{16}$	
	HFCGCTRL2	Byte	00 FFAE$_{16}$	Varies per bit	00$_{16}$	

A.1.17 Development System Support

(Section 4.20.8 on page 275)

	DBGCFG	Byte	00 FF16$_{16}$	Read/Write	00$_{16}$	
	DBGFRZEN	Byte	00 FF18$_{16}$	Read/Write	FF$_{16}$	

A.1.18 Multi-Input Wake-Up (MIWU)

(Section 4.4.3 on page 109)

	WKEDG1	Byte	00 FFC0$_{16}$	Read/Write	00$_{16}$	
	WKEDG2	Byte	00 FFC2$_{16}$	Read/Write	00$_{16}$	
	WKEDG3	Byte	00 FFC4$_{16}$	Read/Write	00$_{16}$	
	WKEDG4	Byte	00 FFC6$_{16}$	Read/Write	00$_{16}$	
	WKPND1	Byte	00 FFC8$_{16}$	Read/Write	00$_{16}$	
	WKPCL1	Byte	00 FFCA$_{16}$	Write Only		
	WKPND2	Byte	00 FFCC$_{16}$	Read/Write	00$_{16}$	
	WKPCL2	Byte	00 FFCE$_{16}$	Write Only		
	WKPND3	Byte	00 FFD0$_{16}$	Read/Write	00$_{16}$	

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	WKPCL3	Byte	00 FFD2$_{16}$	Write Only		
	WKPND4	Byte	00 FFD4$_{16}$	Read/Write	00$_{16}$	
	WKPCL4	Byte	00 FFD6$_{16}$	Write Only		
	WKEN1	Byte	00 FFD8$_{16}$	Read/Write	00$_{16}$	
	WKEN2	Byte	00 FFDA$_{16}$	Read/Write	00$_{16}$	
	WKEN3	Byte	00 FFDC$_{16}$	Read/Write	00$_{16}$	
	WKEN4	Byte	00 FFDE$_{16}$	Read/Write	00$_{16}$	

A.1.19 Interrupt Control Unit (ICU)

(Section 4.3.4 on page 101)

IVCT	Byte	00 FE00$_{16}$	Read Only	10$_{16}$	
NMISTAT	Byte	00 FE02$_{16}$	Read Only	00$_{16}$	
PFAIL	Byte	00 FE04$_{16}$	Read/Write	00$_{16}$	
ISTAT0	Word	00 FE0A$_{16}$	Read Only	0000$_{16}$	
ISTAT1	Word	00 FE0C$_{16}$	Read Only	0000$_{16}$	
IENAM0	Word	00 FE0E$_{16}$	Read/Write	0000$_{16}$	
IENAM1	Word	00 FE10$_{16}$	Read/Write	0000$_{16}$	
IECLR0	Word	00 FE12$_{16}$	Write Only		
IECLR1	Word	00 FE14$_{16}$	Write Only		

A.1.20 Debugger Interface

(Section 4.19.9 on page 267)

DBGRXD0	Word	00 FDC0$_{16}$	Read Only		
DBGRXD2	Word	00 FDC2$_{16}$	Read Only		
DBGRXD4	Word	00 FDC4$_{16}$	Read Only		
DBGRXD6	Word	00 FDC6$_{16}$	Read Only		
DBGRXD8	Word	00 FDC8$_{16}$	Read Only		
DBGRXD10	Word	00 FDCA$_{16}$	Read Only		
DBGRXD12	Word	00 FDCC$_{16}$	Read Only		
DBGRXD14	Word	00 FDCE$_{16}$	Read Only		
DBGTXD0	Word	00 FDD0$_{16}$	Read/Write		
DBGTXD2	Word	00 FDD2$_{16}$	Read/Write		
DBGTXD4	Word	00 FDD4$_{16}$	Read/Write		
DBGTXD6	Word	00 FDD6$_{16}$	Read/Write		

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	DBGTXD8	Word	00 FDD8$_{16}$	Read/Write		
	DBGTXD10	Word	00 FDDA$_{16}$	Read/Write		
	DBGTXD12	Word	00 FDDC$_{16}$	Read/Write		
	DBGTXD14	Word	00 FDDE$_{16}$	Read/Write		
	DBGRXST	Byte	00 FDE0$_{16}$	Varies per bit		
	DBGTXST	Byte	00 FDE2$_{16}$	Read/Write	00$_{16}$	
	DBGTXLOC	Byte	00 FDE4$_{16}$	Read/Write	0F$_{16}$	
	DBGTINT	Byte	00 FDE6$_{16}$	Write Only		
	DBGABORT	Word	00 FDE8$_{16}$	Write Only		
	DBGISESRCA	Word	00 FDEA$_{16}$	Read/Write	0000$_{16}$	

A.1.21 Pulse Width Modulator (PWM)

(Section 4.8.5 on page 148)

PRSC	Byte	00 FD00$_{16}$	Read/Write	00$_{16}$	
CTR	Byte	00 FD02$_{16}$	Read/Write	FF$_{16}$	
PWMPOL	Byte	00 FD04$_{16}$	Read/Write	00$_{16}$	
PWMCNT	Byte	00 FD06$_{16}$	Read/Write	00$_{16}$	
DCR0	Byte	00 FD08$_{16}$	Read/Write	00$_{16}$	
DCR1	Byte	00 FD0A$_{16}$	Read/Write	00$_{16}$	
DCR2	Byte	00 FD0C$_{16}$	Read/Write	00$_{16}$	
DCR3	Byte	00 FD0E$_{16}$	Read/Write	00$_{16}$	
DCR4	Byte	00 FD10$_{16}$	Read/Write	00$_{16}$	
DCR5	Byte	00 FD12$_{16}$	Read/Write	00$_{16}$	
DCR6	Byte	00 FD14$_{16}$	Read/Write	00$_{16}$	
DCR7	Byte	00 FD16$_{16}$	Read/Write	00$_{16}$	

A.1.22 Universal Synchronous/Asynchronous Receiver Transmitter (USART)

(Section 4.9.4 on page 158)

UTBUF	Byte	00 FD20$_{16}$	Read/Write		
URBUF	Byte	00 FD22$_{16}$	Read Only		
UICTRL	Byte	00 FD24$_{16}$	Varies per bit	01$_{16}$	
USTAT	Byte	00 FD26$_{16}$	Read Only	00$_{16}$	
UFRS	Byte	00 FD28$_{16}$	Read/Write	00$_{16}$	
UMDSL	Byte	00 FD2A$_{16}$	Read/Write	00$_{16}$	

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	UBAUD	Byte	00 FD2C$_{16}$	Read/Write	00$_{16}$	
	UPSR	Byte	00 FD2E$_{16}$	Read/Write	00$_{16}$	

A.1.23 Flash Interface

(Section 4.16.7 on page 228)

	IBAI	Word	00 F880$_{16}$	Read/Write		
	IBD	Word	00 F882$_{16}$	Read/Write		
	FCWP0	Word	00 F890$_{16}$	Read/Write	0000$_{16}$	
	FLCR	Word	F8C0$_{16}$	Read/Write	0000$_{16}$	
	FLSR	Byte	F8C2$_{16}$	Varies per bit	00$_{16}$	
	FLPSLR	Byte	F8C4$_{16}$	Read/Write	00$_{16}$	
	FLSTART	Byte	F8C6$_{16}$	Read/Write	14$_{16}$	
	FLTRAN	Byte	F8CA$_{16}$	Read/Write	28$_{16}$	
	FLPROG	Byte	F8D0$_{16}$	Read/Write	0A$_{16}$	
	FLPERASE	Byte	F8D2$_{16}$	Read/Write	09$_{16}$	
	FLSERASE	Byte	F8D4$_{16}$	Read/Write	09$_{16}$	
	FLEND	Byte	F8D8$_{16}$	Read/Write	14$_{16}$	
	FLSEND	Byte	F8DC$_{16}$	Read/Write	32$_{16}$	
	FLRCV	Byte	F8DE$_{16}$	Read/Write	04$_{16}$	

A.1.24 Shared Memory Core

(Section 5.3.9 on page 307)

	SMCCST	Byte	00 F900$_{16}$	Read/Write	00$_{16}$	
	SMCTA	Byte	00 F902$_{16}$	Read/Write and Read Only		See description
	SMHSEM	Byte	00 F904$_{16}$	Varies per bit	00$_{16}$	
	SMCORP0	Word	00 F910$_{16}$	Varies per bit		See description
	SMCORP1	Word	00 F912$_{16}$	Varies per bit		See description
	SMCORP2	Word	00 F914$_{16}$	Varies per bit		See description
	SMCOWP0	Word	00 F920$_{16}$	Read/Write	FFFF$_{16}$	
	SMCOWP1	Word	00 F922$_{16}$	Read/Write	FFFF$_{16}$	
	SMCOWP2	Word	00 F924$_{16}$	Read/Write	FFFF$_{16}$	
	RNGCS	Byte	00 F930$_{16}$	Varies per bit	00$_{16}$	
	RNGD	Byte	00 F932$_{16}$	Read Only		

Software for Hardware Interface (Continued)

A.1.25 Core Access to SuperI/O

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	IHIOA	Word	00 FCE0$_{16}$	Read/Write	00$_{16}$	
	IHD	Byte	00 FCE2$_{16}$	Read/Write	00$_{16}$	
	LKSIOHA	Word	00 FCE4$_{16}$	Read/Write	0002$_{16}$/0000$_{16}$	
	SIOLV	Word	00 FCE6$_{16}$	R/W1C	0000$_{16}$	
	CRSMAE	Word	00 FCE8$_{16}$	Read/Write	0000$_{16}$	
	SIBCTRL	Byte	00 FCEA$_{16}$	Varies per bit	00$_{16}$	

A.1.26 Mobile System Wake-Up Control (MSWC)

(Section 5.5.6 on page 328)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	MSWCTL1	Byte	00 FCC0$_{16}$	Varies per bit	00$_{16}$	Vcc power-up only
	MSWCTL2	Byte	00 FCC2$_{16}$	Varies per bit	00$_{16}$	
	MSWCTL3	Byte	00 FCC4$_{16}$	Varies per bit	01$_{16}$	V_{PP} power-up only
	HCFGBAL	Byte	00 FCC8$_{16}$	Read/Write	00$_{16}$	
	HCFGBAH	Byte	00 FCCA$_{16}$	Read/Write	00$_{16}$	
	MSIEN2	Byte	00 FCCC$_{16}$	Read/Write	00$_{16}$	
	MSHES0	Byte	00 FCCE$_{16}$	R/W1C	00$_{16}$	
	MSHEIE0	Byte	00 FCD0$_{16}$	Read/Write	00$_{16}$	

A.2 HOST DOMAIN REGISTER LIST

A.2.1 Configuration Registers

Access to all host configuration registers is via an index/data scheme that uses the host configuration index/data pair.

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
Common SuperI/O Configuration (Section 6.1.8 on page 343)						
	SID	Byte	Index 20_{16}	Read Only	EC_{16}	
	SIOCF1	Byte	Index 21_{16}	Varies per bit	11_{16}	
	SIOCF5	Byte	Index 25_{16}	Read/Write	00_{16}	
	SIOCF6	Byte	Index 26_{16}	Read/Write	00_{16}	
	SRID	Byte	Index 27_{16}	Read Only		
	SIOCF8	Byte	Index 28_{16}	Read/Write	00_{16}	
	SIOCF9	Byte	Index 29_{16}	Read/Write	01_{16}	
	SIOCFD	Byte	Index $2D_{16}$	Read/Write	00_{16}	
Shared Memory (Section 6.1.11 on page 349)						
	Shared Memory Configuration	Byte	Index $F4_{16}$	Read/Write	00_{16} or 09_{16} depending on $\overline{\text{SHBM}}$ strap	
	Shared Memory Base Address High Byte	Byte	Index $F5_{16}$	Read/Write	00_{16}	
	Shared Memory Base Address Low Byte	Byte	Index $F6_{16}$	Read/Write	00_{16}	
	Shared Memory Size Config	Byte	Index $F7_{16}$	Read/Write	00_{16}	
RTC Configuration When LDN is set to 10_{16}. (Section 6.1.12 on page 353)						
	RLR	Byte	Device Specific	Read/Write	00_{16}	Cleared by H/W reset only
	DOMAO	Byte	Index $F0_{16}$	Read/Write	00_{16}	
	MONAO	Byte	Index $F1_{16}$	Read/Write	00_{16}	
	CENO	Byte	Index $F3_{16}$	Read/Write	00_{16}	

A.2.2 Host Runtime Registers

Software for Hardware Interface (Continued)

Shared Memory Host

The base address is defined by LDN $0F_{16}$ (Section 5.3.8 on page 303).

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	SMIMA0	Byte	Offset 00_{16}	Read/Write		
	SMIMA1	Byte	Offset 01_{16}	Read/Write		
	SMIMA2	Byte	Offset 02_{16}	Read/Write		
	SMIMA3	Byte	Offset 03_{16}	Read/Write		
	SMIMD	Byte	Offset 04_{16}	Read/Write		
	SMHC	Byte	Offset 05_{16}	Read/Write	00_{16}	
	SMHST	Byte	Offset 06_{16}	Varies per bit	00_{16}	
	SMHAP1	Byte	Offset 07_{16}	Varies per bit	02_{16}	
	SMHAP2	Byte	Offset 08_{16}	Varies per bit	02_{16}	
	SMHSEM	Byte	Offset $0C_{16}$	Varies per bit	00_{16}	

MSWC Host Registers

The base address is defined by LDN 04_{16} (Section 5.5.5 on page 323).

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	WK_STS0	Byte	Offset 00_{16}	R/W1C	00_{16}	
	WK_EN0	Byte	Offset 02_{16}	Read/Write	00_{16}	
	WK_CFG	Byte	Offset 04_{16}	Read/Write	00_{16}	
	WK_SIGV	Byte	Offset 06_{16}	Read Only		
	WK_STATE	Byte	Offset 07_{16}	Read/Write		
	WK_SMIEN0	Byte	Bank 2 Offset 13_{16}	Read/Write	00_{16}	
	WK_IRQEN0	Byte	Bank 2 Offset 15_{16}	Read/Write	00_{16}	

Host Interface (HI)

The base address is defined by LDN 06_{16} (“Host Addresses” on page 277).

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	DBBOUT	Byte	Defined in LDN 06_{16} index 60_{16}, 61_{16}	R		
	STATUS	Byte	Defined in LDN 06_{16} index 62_{16}, 63_{16}	R	00_{16}	
	DBBIN	Byte	Defined in LDN 06_{16} index 60_{16}, 61_{16}	W		

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	COMAND	Byte	Defined in LDN 06_{16} index 62_{16}, 63_{16}	W		

Power Management Channel 1

The base address is defined by LDN 11_{16} (“Host Addresses” on page 277).

DBBOUT	Byte	Defined in LDN 11_{16} index 60_{16}, 61_{16}	R		
STATUS	Byte	Defined in LDN 11_{16} index 62_{16}, 63_{16}	R	00_{16}	
DBBIN	Byte	Defined in LDN 11_{16} index 60_{16}, 61_{16}	W		
COMAND	Byte	Defined in LDN 11_{16}h index 62_{16}, 63_{16}	W		

Power Management Channel 2

The base address is defined by LDN 12_{16} (“Host Addresses” on page 277).

DBBOUT	Byte	Defined in LDN 12_{16} index 60_{16}, 61_{16}	R		
STATUS	Byte	Defined in LDN 12_{16} index 62_{16}, 63_{16}	R	00_{16}	
DBBIN	Byte	Defined in LDN 12_{16} index 60_{16}, 61_{16}	W		
COMAND	Byte	Defined in LDN 12_{16} index 62_{16}, 63_{16}	W		

RTC

Access is via a index/data scheme that uses an index/data pair pointed to by LDN 10_{16} (Section 6.2.15 on page 363).

Software for Hardware Interface (Continued)

Module	Register Name	Size	Register Address	Access Type	Value After Reset	Comments
	SEC	Byte	Index 00_{16}	Read/Write	00_{16}	
	SECA	Byte	Index 01_{16}	Read/Write	00_{16}	
	MIN	Byte	Index 02_{16}	Read/Write	00_{16}	
	MINA	Byte	Index 03_{16}	Read Only	00_{16}	
	HOR	Byte	Index 04_{16}	Read/Write	00_{16}	
	HORA	Byte	Index 05_{16}	Read/Write	00_{16}	
	DOW	Byte	Index 06_{16}	Read/Write	00_{16}	
	DOM	Byte	Index 07_{16}	Read/Write	00_{16}	
	MON	Byte	Index 08_{16}	Read/Write	00_{16}	
	YER	Byte	Index FF_{16}	Read/Write	00_{16}	
	CRA	Byte	Index $0A_{16}$	Read/Write	20_{16}	
	CRB	Byte	Index $0B_{16}$	Read/Write	00_{16}	
	CRC	Byte	Index $0C_{16}$	Read Only	00_{16}	
	CRD	Byte	Index $0D_{16}$	Read Only	00_{16}	
	DOMA	Byte	Prog. Index	Read/Write	$C0_{16}$	
	MONA	Byte	Prog. Index	Read/Write	$C0_{16}$	
	CEN	Byte	Prog. Index	Read/Write	00_{16}	

Software for Hardware Interface (Continued)

A.3 CORE DOMAIN REGISTER LAYOUT

A.3.1 Module Configuration

(Section 2.3 on page 53 and Section 2.4.2 on page 58)

	7	6	5	4	3	2	1	0
MCFG/MCFGSH	GTMON	Reserved		CLKOM		EXMEM16	ENEMEM	ENEIO
EICFG	Reserved						EXWINT46	EXWINT45
IOEE1	Reserved			EEPA4	EEPA3	EEPA2	EEPA1	EEPA0
IOEE2	Reserved			EEPC0	EEPD3	EEPB2	EEPB1	EEPB0
STRPST (p. 53)	Reserved	Reserved	Reserved	Reserved	Reserved	BADDR1	BADDR0	SHBM

A.3.2 Bus Interface Unit (BIU)

(Section 4.1.10 on page 83)

	15 12	11	10	9	8	7	6	5	4	3	2	1	0
BCFG	N/A					Reserved					ISTL	OBR	EWR
IOCFG	Reserved			IPST	Res	BW	Reserved		HOLD		WAIT		
SZCFGn	Reserved	FRE	IPRE	IPST	Res	BW	WBR	BRE	HOLD		WAIT		

A.3.3 DMA Controller

(Section 4.2.8 on page 92)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAn	Device A Address Counter																															
ADRAn	Device A Address																															
ADCBn	Device B Address Counter																															
ADRBn	Device B Address																															
BLTCn	Reserved																Block Length Counter															
BLTRn	Reserved																Block Length															
CNTLn	N/A																Res	INCB		ADB	INCA		ADA	SWRQ	BPC	OT	DIR	IND	TCS	EOVR	ETC	CHEN
STATn	N/A																								Reserved				VLD	CHAC	OVR	TC

A.3.4 General-Purpose I/O (GPIO) Port

(Section 4.5.6 on page 119)

	7	6	5	4	3	2	1	0
PADIR	PA Port Direction							
PBDIR	PB Port Direction							
PCDIR	PC Port Direction							
PDDIR	PD Port Direction							
PFDIR	PF Port Direction							
PGDIR	PG Port Direction							
PIDIR	PI Port Direction							
PJDIR	PJ Port Direction							
PKDIR	PK Port Direction							
PLDIR	Reserved			PL Port Direction				
PMDIR	PM Port Direction							
PADIN	PA Port Input Data							
PBDIN	PB Port Input Data							

Software for Hardware Interface (Continued)

	7	6	5	4	3	2	1	0
PCDIN	PC Port Input Data							
PDDIN	PD Port Input Data							
PEDIN	PE Port Input Data							
PFDIN	PF Port Input Data							
PHDIN	PH Port Input Data							
PIDIN	PI Port Input Data							
PJDIN	PJ Port Input Data							
PKDIN	PK Port Input Data							
PLDIN	Reserved			PL Port Input Data				
PMDIN	PM Port Input Data							
PADOUT	PA Port Output Data							
PBDOUT	PB Port Output Data							
PCDOUT	PC Port Output Data							
PDDOUT	PD Port Output Data							
PFDOUT	PF Port Output Data							
PHDOUT	PH Port Output Data							
PIDOUT	PI Port Output Data							
PJDOUT	PJ Port Output Data							
PKDOUT	PK Port Output Data							
PLDOUT	Reserved			PL Port Output Data				
PMDOUT	PM Port Output Data							
PAWPU	PA Port Weak Pull-up Enable							
PBWPU	PB Port Weak Pull-up Enable							
PCWPU	PC Port Weak Pull-up Enable							
PDWPU	PD Port Weak Pull-up Enable							
PEWPU	PE Port Weak Pull-up Enable			Reserved	PE Port Weak Pull-up Enable	Reserved		
PFWPU	PF Port Weak Pull-up Enable							
PAALT	PA Pins Alternate Function Enable							
PBALT	PB Pins Alternate Function Enable							
PCALT	PC Pins Alternate Function Enable							
PDALT	PD Pins Alternate Function Enable							
PEALT	PE Pins Alternate Function Enable							
PFALT	PF Pins Alternate Function Enable							

A.3.5 PS/2 Interface

(Section 4.6.5 on page 128)

	7	6	5	4	3	2	1	0
PSDAT	Data							
PSTAT	Reserved	RFERR	ACH			PERR	EOT	SOT
PSCON	WPUEN	IDB			HDRV		XMT	EN
PSOSIG	CLK4	WDAT4	CLK3	CLK2	CLK1	WDAT3	WDAT2	WDAT1
PSISIG	RCLK4	RDAT4	RCLK3	RCLK2	RCLK1	RDAT3	RDAT2	RDAT1
PSIEN	Reserved					DSMIE	EOTIE	SOTIE

A.3.6 Core Interface

(Section 5.1.4 on page 282 and Section 5.2.3 on page 290)

	7	6	5	4	3	2	1	0
HICTRL	Reserved	PMICIE	PMOCIE	PMHIE	IBFCIE	OBECIE	OBFMIE	OBFKIE
HIIRQC	Reserved	IRQNPOL	IRQM			IRQ11B	IRQ12BO BFMIE	IRQ1B
HIKMST	ST3	ST2	ST1	ST0	A2	F0	IBF	OBF
HIKDO	Keyboard DBBOUT Data							
HIMDO	Mouse DBBOUT Data							
HIKMDI	Keyboard/Mouse DBBIN Data							

Software for Hardware Interface (Continued)

	7	6	5	4	3	2	1	0
HIPMnST	ST3	ST2	ST1	ST0	A2	F0	IBF	OBF
HIPMnDO	PM Channel DBBOUT Data							
HIPMnDOC	PM Channel DBBOUT Data							
HIPMnDOM	PM Channel DBBOUT Data							
HIPMnDI	PM Channel DBBIN Data							
HIPMnDIC	PM Channel DBBIN Data							
HIPMnCTRL	EME	SCIPOL	PLMS			Reserved	OBEIE	IBFIE
HIPMnIC	SCIIS	SMIPOL	PLMM			SCIB	SMIB	IRQB
HIPMnIE	Reserved		HSMIE	HSCIE	HIRQE	SMIE	SCIE	IRQE

A.3.7 Multi-Function Timer (MFT16)

(Section 4.7.7 on page 142)

	15 8	7	6	5	4	3	2	1	0
TnPRSC	N/A	Reserved			CLKPS				
TnCKC	N/A	Reserved		C2CSEL			C1CSEL		
TnCTRL	N/A	Reserved	TAOUT	TBEN	TAEN	TBEDG	TAEDG	MDSEL	
TnICTL	N/A	TDIEN	TCIEN	TBIEN	TAIEN	TDPND	TCPND	TBPND	TAPND
TnICLR	N/A	Reserved				TDCLR	TCCLR	TBCLR	TACLR

A.3.8 Timing and WATCHDOG (TWD)

(Section 4.10.3 on page 166)

	15 8	7	6	5	4	3	2	1	0
TWCFG	N/A	Reserved		WDSDME	WDCT0I	LWDCNT	LTWDT0	LTWCP	LTWCFG
TWCP	N/A	Reserved					MDIV		
TWDT0	Preset								
T0CSR	N/A	Reserved				WDLTD	Reserved	TC	RST
WDCNT	N/A	PRESET							
WDSDM	N/A	RSDATA							

A.3.9 Analog to Digital Converter (ADC)

(Section 4.11.6 on page 177)

	15 10	9 8	7	6	5	4	3	2	1	0
ADCSTS	N/A		Reserved			OPSHTYPE	OPSHEV	OTSEV	OVFEV	EOCEV
ADCCNF	N/A		Reserved			INTOSEN	INTOTEN	INTECEN	REMDEN	ADCEN
ACLKCTL	N/A		Reserved		SCLKDIV					
ADLYCTL	N/A		TMPDLY				Res	VOLDLY		
TLOCOTL	N/A		OTSLIM							
ADCPINX	N/A		Index							
ADCPD	Parameter Data									
TCHANCTL	N/A		DATVAL	RDMEASEN	INTDVEN	SELIN				
TCHANDAT	N/A		TCHDAT							
VCHN1CTL	N/A		DATVAL	CSCALE	INTDVEN	SELIN				
VCHN1DAT	Reserved	VCHDAT								
VCHN2CTL	N/A		DATVAL	CSCALE	INTDVEN	SELIN				
VCHN2DAT	Reserved	VCHDAT								
VCHN3CTL	N/A		DATVAL	CSCALE	INTDVEN	SELIN				
VCHN3DAT	Reserved	VCHDAT								

Software for Hardware Interface (Continued)

A.3.10 Digital to Analog (DAC)

(Section 4.12.5 on page 193)

	7	6	5	4	3	2	1	0
DACCTRL	Reserved			ENIDLE	DACEN3	DACEN2	DACEN1	DACEN0
DACDATi	DAC DATAi							

A.3.11 ACCESS.bus Interface (ACB)

(Section 4.13.8 on page 202)

	7	6	5	4	3	2	1	0
ACBnSDA	DATA							
ACBnST	SLVSTP	SDAST	BER	NEGACK	STASTR	NMATCH	MASTER	XMIT
ACBnCST	ARP-MATCH	MATCHAF	TGSCL	TSDA	GMATCH	MATCH	BB	BUSY
ACBnCTL1	STASTRE	NMINTE	GCMEN	ACK	Reserved	INTEN	STOP	START
ACBnADDR	SAEN	ADDR						
ACBnCTL2	SCLFRQ6	SCLFRQ5	SCLFRQ4	SCLFRQ3	SCLFRQ2	SCLFRQ1	SCLFRQ0	ENABLE

A.3.12 Analog Comparators Monitor (ACM)

(Section 4.14.5 on page 212)

	7	6	5	4	3	2	1	0
ACMCTS	Reserved				OVUNTHEV	EOCEV	EOMEV	START
ACMCNF	Reserved		OVUNSEL	INTOUEN	INTEMEN	Reserved	ACMMOD	
ACMTIM	Reserved		T0DIV		Reserved	SMPDLY		
THRDAT	Reserved		THRSHD5	THRSHD4	THRSHD3	THRSHD2	THRSHD1	THRSHD0
CMPRES	CMPIN7	CMPIN6	CMPIN5	CMPIN4	CMPIN3	CMPIN2	CMPIN1	CMPIN0
VOLDAT0-7	Reserved		Voltage Level Data 5-0					

A.3.13 Power Management (PM)

(Section 4.17.4 on page 237)

	7	6	5	4	3	2	1	0
PMCSR	OLFC	OHFC	WBPSM	Reserved		IDLE	DHF	Reserved

A.3.14 High-Frequency Clock Generator (HFCG)

(Section 4.18.7 on page 243)

	7	6	5	4	3	2	1	0
HFCGCTRL1	Reserved	FAST96	LOAD96	IVLID	OHFC	PENABLE	FAST	LOAD
HFCGML	HFCGM7-0							
HFCGMH	HFCGM							
HFCGN	Reserved			HFCGN4-0				
HFCGIL	HFCGI7-0							
HFCGIH	Reserved		HFCGI13-8					
HFCGP	Reserved			HFCGP4-0				
HFCGCTRL2	Reserved			96MON	MONERR	SCESTP	SCESTR	SENABLE

A.3.15 Development System Support

(Section 4.20.8 on page 275)

	7	6	5	4	3	2	1	0
DBGCFG	Reserved					BRKLE	FREEZE	ON
DBGFRZEN	Reserved		HIFEN	USARTFEN	ACB2FEN	ACB1FEN	MFT2FEN	MFT1FEN

A.3.16 Multi-Input Wake-Up (MIWU)

(Section 4.4.3 on page 109)

7	6	5	4	3	2	1	0

Software for Hardware Interface (Continued)

WKEDG1	WKED17	WKED16	WKED15	WKED14	WKED03	WKED12	WKED11	WKED10
WKEDG2	WKED27	WKED26	WKED25	WKED24	WKED23	WKED22	WKED21	WKED20
WKEDG3	WKED37	WKED36	WKED35	WKED34	WKED33	WKED32	WKED31	WKED30
WKEDG4	WKED47	WKED46	WKED45	WKED44	WKED43	WKED42	WKED41	WKED40
WKPND1	WKPD17	WKPD16	WKPD15	WKPD14	WKPD03	WKPD12	WKPD11	WKPD10
WKCL1	WKCL17	WKCL16	WKCL15	WKCL14	WKCL03	WKCL12	WKCL11	WKCL10
WKPND2	WKPD27	WKPD26	WKPD25	WKPD24	WKPD23	WKPD22	WKPD21	WKPD20
WKCL2	WKCL27	WKCL26	WKCL25	WKCL24	WKCL23	WKCL22	WKCL21	WKCL20
WKPND3	WKPD37	WKPD36	WKPD35	WKPD34	WKPD33	WKPD32	WKPD31	WKPD30
WKCL3	WKCL37	WKCL36	WKCL35	WKCL34	WKCL33	WKCL32	WKCL31	WKCL30
WKPND4	WKPD47	WKPD46	WKPD45	WKPD44	WKPD43	WKPD42	WKPD41	WKPD40
WKCL4	WKCL47	WKCL46	WKCL45	WKCL44	WKCL43	WKCL42	WKCL41	WKCL40
WKEN1	WKEN17	WKEN16	WKEN15	WKEN14	WKEN13	WKEN12	WKEN11	WKEN10
WKEN2	WKEN27	WKEN26	WKEN25	WKEN24	WKEN23	WKEN22	WKEN21	WKEN20
WKEN3	WKEN37	WKEN36	WKEN35	WKEN34	WKEN33	WKEN32	WKEN31	WKEN30
WKEN4	WKEN47	WKEN46	WKEN45	WKEN44	WKEN43	WKEN42	WKEN41	WKEN40

A.3.17 Interrupt Control Unit (ICU)

(Section 4.3.4 on page 101)

	15 12	**11 8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IVCT	N/A		0	0	INTVECT					
NMISTAT	N/A		Reserved							EXT
PFAIL	N/A		Reserved					ENLCK	PIN	EN
ISTAT0	Interrupt Status Register (15-0)									
ISTAT1	Interrupt Status Register (31-16)									
IENAM0	Interrupt Enable (15-0)									
IENAM1	Interrupt Enable (31-16)									

A.3.18 Debugger Interface

(Section 4.19.9 on page 267)

	15 8	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBGRXD0	RX_Data0								
DBGRXD2	RX_Data2								
DBGRXD4	RX_Data4								
DBGRXD6	RX_Data6								
DBGRXD8	RX_Data8								
DBGRXD10	RX_Data10								
DBGRXD12	RX_Data12								
DBGRXD14	RX_Data14								
DBGTXD0	TX_Data0								
DBGTXD2	TX_Data2								
DBGTXD4	TX_Data4								
DBGTXD6	TX_Data6								
DBGTXD8	TX_Data8								
DBGTXD10	TX_Data10								
DBGTXD12	TX_Data12								
DBGTXD14	TX_Data14								
DBGRXST	N/A	MSG_LEN			PID				RX_BUSY
DBGTXST	N/A	Reserved					MSG_LEN		
DBGTXLOC	N/A	Reserved				PID			
DBGTINT	N/A	Reserved							ASSERT
DBGABORT	Reserved								P_0
DBGISESRCA	Reserved							ABORT_0	RX_0

Software for Hardware Interface (Continued)

A.3.19 Pulse with Modulator (PWM)

(Section 4.8.5 on page 148)

	7	6	5	4	3	2	1	0
PRSC	PRSC(7-0)							
CTR	CTR(7-0)							
PWMPOL	INVP(7-0)							
PWMCNT	Reserved							PWR
DCRi	DCRi(7-0)							

A.3.20 Universal Synchronous/Asynchronous Receiver Transmitter (USART)

(Section 4.9.4 on page 158)

	7	6	5	4	3	2	1	0
UTBUF	UTBUF							
URBUF	URBUF							
UICTRL	EEI	ERI	ETI	Reserved	Reserved	Reserved	RBF	TBE
USTAT	Reserved	XMIP	RB9	BKD	ERR	DOE	FE	PE
UFRS	Reserved	PEN	PSEL		XB9	STP	CHAR	
UMDSL	Reserved	Reserved	ERD	ETD	CKS	BRK	UnATN	MOD
UBAUD	UDIV(7-0)							
UPSR	UPSC					UDIV(10-8)		

A.3.21 Flash Interface

(Section 4.16.7 on page 228)

	15 9	8	7	6	5	4	3	2	1	0
IBAI	Reserved		Indirect Memory Address (7-0)							
IBD	Information Block Data									
FCWP0	FC0WP(15-0)									
FCWP1	FC1WP(15-0)									
FCWP2	FC2WP(15-0)									
FLCR	Res	DIS-VRF	SMER	PMER	PE	IENPROG	Reserved		LOWP-WRS	LOWP-WRF
FLSR	N/A		Reserved			DERR	FMLFULL	FMBUSY	PERR	EERR
FLPSLR	N/A		Reserved				FTDIV(3-0)			
FLSTART	N/A		FTSTART(7-0)							
FLTRAN	N/A		FTTRAN(7-0)							
FLPROG	N/A		FTPROG(7-0)							
FLPERASE	N/A		FTPER(7-0)							
FLSERASEi	N/A		FTSER(7-0)							
FLEND	N/A		FTEND(7-0)							
FLSEND	N/A		FTSEND(7-0)							
FLRCV	N/A		FTRCV(7-0)							

A.3.22 Shared Memory Core

(Section 5.3.9 on page 307)

	15 8	7	6	5	4	3	2	1	0
SMCCST	N/A	HSEMIE	HSEMW	HLOCK	HERES		HERRIEN	HWERR	HRERR
SMCTA	N/A	Reserved				MBSD			
SMHSEM	N/A	CSEM3	CSEM2	CSEM1	CSEM0	HSEM3	HSEM2	HSEM1	HSEM0
SMCOHRP0	ORPLA(15-0)								
SMCOHRP1	ORP(15-2)							Reserved	
SMCOHRP2	ORP(31-16)								
SMCOHWP0	OWPLA(15-0)								
SMCOHWP1	OWP(15-2)							Reserved	
SMCOHWP2	OWP(31-16)								

Software for Hardware Interface (Continued)

	15 8	7	6	5	4	3	2	1	0
RNGCS	N/A	Reserved			CLKP			DVALID	RNGE
RNGD	N/A	RNGD(0-7)							

A.3.23 Core Access to SuperI/O

(Section 5.4.1 on page 314)

	15 8	7	6	5	4	3	2	1	0
IHIOA	Reserved	Indirect Host I/O Offset							
IHD	N/A	Indirect Host Data							
LKSIOHA	Reserved							LKRTCHA	LKCFG
SIOLV	Reserved							RTCLV	CFGLV
CRSMAE	Reserved							RTCAE	CFGAE
SIBCTRL	N/A	Reserved				RTCMR	CSWR	CSRD	CSAE

A.3.24 MSWC

(Section 5.5.6 on page 328)

	7	6	5	4	3	2	1	0
MSWCTL1	Reserved		HSECM	VHCFGLK	VHCFGA	LPCRSTA	HPWRON	HRSTOB
MSWCTL2	CFGPSO	CFGPBM	ACPIS5	ACPIS4	ACPIS3	ACPIS2	ACPIS1	ACPIS0
MSWCTL3	Reserved					RTCAL	LPFTO	HRAPU
HCFGBAL	Host Configuration Registers Base Address Low							
HCFGBAH	Host Configuration Registers Base Address High							
MSIEN2	EICFGPSO	EICFG-PBM	EIACPIS5	EIACPIS4	EIACPIS3	EIACPIS2	EIACPIS1	EIRTCAL
MSHES0	Module IRQ Event Status	Software Event Status	Reserved		RINGEvent Status	Reserved	RI2 Event Status	RI1 Event Status
MSHEIE0	Module IRQ Event Enable	Software Event Enable	Reserved		RINGEvent Enable	Reserved	RI2 Event Enable	RI1 Event Enable

A.4 HOST DOMAIN REGISTER LAYOUT

Host Configuration Registers

A.4.1 SuperI/O Configuration

(Section 6.1.8 on page 343)

	7	6	5	4	3	2	1	0
SID	Family ID							
SIOCF1	Reserved		Number of DMA Wait States		Number of I/O Wait States		Software Reset	SuperI/O Devices Enable
SIOCF5	Reserved			$\overline{\text{SMI}}$ to IRQ2 Enable	Reserved			
SIOCF6	SCIOF6 Software Lock	General-Purpose Scratch		RTC Disabled	Reserved			
SRID	Chip Revision ID							
SIOCF8	Reserved				LPC Bus Master Num.		Reserved	
SIOCF9	Reserved			Module Enable Status	Valid Multiplier Clock Status	Clock Enable	SuperI/O Clock Domain Source	
SIOCFD	Reserved						Power Supply Off	Power Button Mode

A.4.2 Shared Memory Configuration

(Section 6.1.11 on page 349)

	7	6	5	4	3	2	1	0
Shared Memory Configuration	BIOS FWH ID				BIOS FWH Enable	User-Defined Memory Space Enable	BIOS Extended Space Enable	BIOS LPC Enable
Shared Memory Base Address High Byte	User-Defined Memory Zone Address High							
Shared Memory Base Address Low Byte	User-Defined Memory Zone Address Low							
Shared Memory Size Configuration	Reserved				User-Defined Memory Zone Size			

A.4.3 RTC Configuration

(Section 6.1.12 on page 353)

	7	6	5	4	3	2	1	0
RLR	Block Standard RAM	Block RAM Write	Block Extended RAM Write	Block Extended RAM Read	Block Extended RAM	Reserved		
DOMAO	Reserved	Date of Month Alarm Register Offset Value						
MONAO	Reserved	Month Alarm Register Offset Value						
CENO	Reserved	Century Register Offset Value						

Host Runtime Registers

A.4.4 Shared Memory Host

(Section 5.3.8 on page 303)

	7	6	5	4	3	2	1	0
SMIMA0	Indirect Memory Address (7-0)							
SMIMA1	Indirect Memory Address (15-8)							
SMIMA2	Indirect Memory Address (23-16)							

Software for Hardware Interface (Continued)

SMIMA3	Indirect Memory Address (31-24)							
SMIMD	Indirect Memory Data (7-0)							
SMHC	Reserved		PMER	Reserved				
SMHST	Reserved			HDERR	FMLFULL	HBUSY	HPERR	HEERR
SMHAP1,2	Host Access Protection Index				Index Write	Host Lock Protection	Host Write Protection	Host Read Protection
SMHSEM	CSEM3	CSEM2	CSEM1	CSEM0	HSEM3	HSEM2	HSEM1	HSEM0

A.4.5 MSWC Host

(Section 5.5.5 on page 323)

	7	6	5	4	3	2	1	0
WK_STS0	Module IRQ Event Status	Software Event Status	Reserved		RING Event Status	Reserved	RI2 Event Status	RI1 Event Status
WK_EN0	Module IRQ Event Enable	Software Event Enable	Reserved		RING Event Enable	Reserved	RI2 Event Enable	RI1 Event Enable
WK_CFG	Reserved						Configuration Bank Select	
WK_SIGV	Reserved		$\overline{\text{PWUREQ}}$ Wakeup or IRQ	$\overline{\text{PWUREQ}}$ Output Value	PM2 $\overline{\text{SMI}}$ Output	PM1 $\overline{\text{SMI}}$ Output	$\overline{\text{SMI}}$ Wake-up Output	$\overline{\text{SMI}}$ Output Value
WK_STATE	Reserved		S5	S4	S3	S2	S1	Reserved
WK_SMIEN0	Reserved	Software Event to $\overline{\text{SMI}}$ Enable	Reserved		Ring Event to $\overline{\text{SMI}}$ Enable	Reserved	RI2 Event to $\overline{\text{SMI}}$ Enable	RI1 Event to $\overline{\text{SMI}}$ Enable
WK_IRQENO	Reserved	Software Event to IRQ Enable	Reserved		Ring Event to IRQ Enable	Reserved	RI2 Event to IRQ Enable	RI1 Event to IRQ Enable

A.4.6 Host Interface (HI) Registers

(“Host Addresses” on page 277)

	7	6	5	4	3	2	1	0
DBBOUT	Keyboard/Mouse DBBOUT Data							
STATUS	ST3	ST2	ST1	ST0	A2	F0	IBF	OBF
DBBIN	Keyboard/Mouse DBBIN Data							
COMAND	Keyboard/Mouse DBBIN Data							

A.4.7 RTC Registers

(Section 6.2.15 on page 363)

	7	6	5	4	3	2	1	0
SEC	Seconds Data							
SECA	Seconds Alarm Data							
MIN	Minutes Data							
MINA	Minutes Alarm Data							
HOR	Hours Data							
HORA	Hours Alarm Data							
DOW	Day of Week Data							
DOM	Date of Month Data							
MON	Month Data							
YER	Year Data							
CRA	Update in Progress	Divider Chain Control 2-0			Periodic Interrupt Rate Select 3-0			

Software for Hardware Interface (Continued)

CRB	Set Mode	Periodic Interrupt Enable	Alarm Interrupt Enable	Update Ended Interrupt Enable	Reserved	Data Mode	Hour Mode	Daylight Savings
CRC	IRQ Flag	Periodic Interrupt Flag	Alarm Interrupt Flag	Update Ended Interrupt Flag	Reserved			
CRD	Valid RAM and Time	Reserved						
DOMA	Date of Month Alarm Data							
MONA	Month Alarm Data							
CEN	Century Data							

B. Software for Hardware Interface

B.1 FACTORY PARAMETERS

The following table is stored in the factory parameters section of the flash.

Information Block Address	Data [Word]
0000_{16}	unsigned int RevisionCode
0002_{16}	unsigned int count
$007C_{16}$	Copy of SRID register indication for the device revision number. For PC87591E: $FF0n_{16}$ For PC87591S: $FF2n_{16}$

RevisionCode Interpretation

- $00xx_{16}$ - Pre-production versions; contact NSC for usage guidelines.
- $01xx_{16}$ - Production version xx of PC87591x devices (xx is a version number starting with 0x00).
- $02xx_{16}$ - Production version xx of PC87591x devices with special handling noted in user information.
- $FFxx_{16}$ - A non-tested device; contact National Semiconductor for usage guidelines.
- Other - Reserved for future use.

The ADC should be initialized after reset with the factory parameters setting that are stored in flash memory. The code below is an example of how the data may be loaded into the ADC.

```
#define IBAI            (*((volatile unsigned short *)0xF880))
#define IBD             (*((volatile unsigned short *)0xF882))
#define ADCPINX          (*((volatile unsigned char*)0xFF2A))
#define ADCPD           (*((volatile unsigned int *)0xFF2C))

void CalibrateADC()
{
  unsigned int  i, count;

 IBAI = 0x0000;                         // Set address to RevisionCode
  if ( (IBD & 0xff00) == 0x0100)        // Check that RevisionCode is 0x01yy;
                                        // refer to user information on
                                        // other cases.
  {           // Factory Parameters are programed with the correct version
    IBAI= 0x0002;                        // Set address to ‘count’
    count = IBD;                         // Read the number of elements to copy
    for (i=1; i<= count ; i++)
     {
         ADCPINX = (unsigned char)i;    // Set pointer in ADC register
         IBAI = 2*i+2;                  // Set pointer in Flash
         ADCPD  = IBD;                  // Copy from flash to ADC register
    }
    ADCPINX = 0x00;  // change the index back to the default register.
    ADCPD = 0x0001;   // Lock the ADC calibration registers.}
}
```

B.2 PROTECTION INFORMATION

The flash protection word is used to protect the contents of the on-chip flash from being altered. It also prevents the system from executing from an erased flash and sets various system defaults. The following are examples of common settings for the protection word. The protection word should be written into the flash as the last operation after updating the flash via the JTAG. The protection word may be written by a write operation, using the programing tool to address 00FE$_{16}$ in the information block (block 2). Note that the protection word must be updated in all operating environments (DEV, OBD, PROG and IRE).

Example 1

Core boot block size is 4 Kbytes

  - Bit 3-0 (Core Boot Block): 1110

- Shared BIOS using an expansion memory
  - Bit 4 (Host Boot Block): 1
  - Bit 6 (Force MBTA Zero): 0
- RTC may be accessed by the host (at default)
  - Bit 5 (RTC Lock Default): 0
- Flash is not protected to allow debug
  - Bits 9-7 (Flash Contents Protect): 111
- Use $\overline{\text{RESET2}}$ on pin IOPB7/$\overline{\text{RING}}$/$\overline{\text{PFAIL}}$/$\overline{\text{RESET2}}$
  - Bits 15-14 (RST2EN): 01

The value that should be written to the protection word is 7F9E$_{16}$ (binary 0111 1111 1001 1110).

Example 2

Same as Example 1, but for production version, once all debugging is completed. Change bits 9-7 from 111 to 000. Thus the value that should be written to the protection word is 7C1E$_{16}$ (binary 0111 1100 0001 1110).

Example 3

- Core boot block size is 8 Kbytes
  - Bit 3-0 (Core Boot Block): 1101
- Shared BIOS using an expansion memory
  - Bit 4 (Host Boot Block): 1
  - Bit 6 (Force MBTA Zero): 0
- RTC may be accessed by the host (at default)
  - Bit 5 (RTC Lock Default): 0
- Flash is not protected to allow debug
  - Bits 9-7 (Flash Contents Protect): 111
- Use $\overline{\text{RESET2}}$ on pin IOPD2/EXWINT24/$\overline{\text{RESET2}}$
  - Bits 15-14 (RST2EN): 10

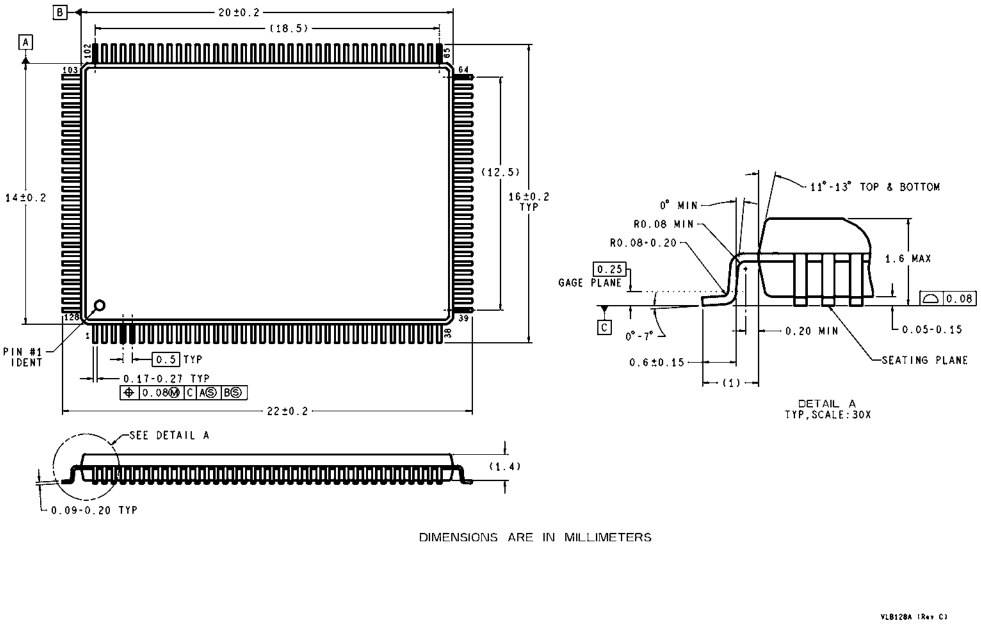
The value that should be written to the protection word is BF9D$_{16}$ (binary 1011 1111 1001 1101).

Example 4

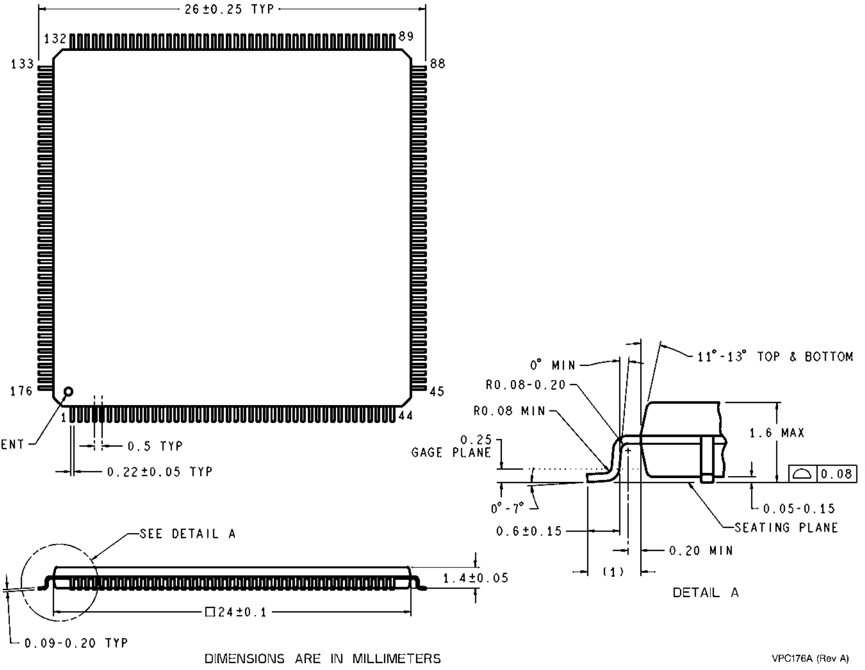
- Core boot block size is 4 Kbytes
  - Bit 3-0 (Core Boot Block): 1110
- Shared BIOS using an expansion memory
  - Bit 4 (Host Boot Block): 1
  - Bit 6 (Force MBTA Zero): 0
- After reset, RTC may be accessed only by the core, and host access is blocked until host access is enabled by the core.
  - Bit 5 (RTC Lock Default): 1
- Flash is not protected to allow debug
  - Bits 9-7 (Flash Contents Protect): 111
- Do not use $\overline{\text{RESET2}}$ (only RESET1 is in use)
  - Bits 15-14 (RST2EN): 11

The value that should be written to the protection word is FFBE$_{16}$ (binary 1111 1111 1011 1110)..

Physical Dimensions inches (millimeters) unless otherwise noted



128-pin LQFP Package
Order Number: PC87591E-VLB/PC87591S-VLB
NS Package Number VLB128



176-Low Profile Plastic Quad Flatpack (LQFP)
Order Number PC87591E-VPC/PC87591S-VPC/ PC87591L-VPC
NS Package Number VPC176

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Email:
new.feedback@nsc.com

National Semiconductor
Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 87 90

National Semiconductor
Asia Pacific Customer
Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

National Semiconductor
Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507
Email: nsj.crc@jksmtp.nsc.com

www.national.com

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.