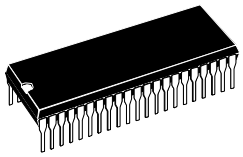
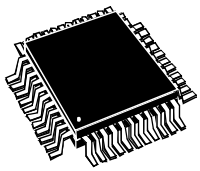
ST72774/ST72754/ST72734

8-BIT USB MCU FOR MONITORS, WITH UP TO 60K OTP, 1K RAM, ADC, TIMER, SYNC, TMU, PWM/BRM, H/W DDC & I^2C

- User ROM/OTP/EPROM: up to 60 Kbytes
- Data RAM: up to 1 Kbytes (256 bytes stack)
- 8 MHz Internal Clock Frequency in fast mode, 4 MHz in normal mode
- Run and Wait CPU modes
- System protection against illegal address jumps and illegal opcode execution
- Sync Processor for Mode Recognition, power management and composite video blanking, clamping and free-running frequency generation
  - Corrector mode
  - Analyzer mode
- USB (Universal Serial Bus) for monitor function [1]
  - Three endpoints
  - Integrated 3.3V voltage regulator
  - Transceiver
  - Suspend and Resume operations
- Timing Measurement Unit (TMU) for autoposition and autosize [1]
- Fast I^2C Single Master Interface
- DDC Bus Interface with:
  - DDC1/2B protocol implemented in hardware
  - Programmable DDC CI modes
  - Enhanced DDC (EDDC) address decoding
- 31 I/O lines



PSDIP42



TQFP44
10 x 10

- 2 lines programmable as interrupt inputs
- 16-bit timer with 2 input captures and 2 output compare functions
- 8-bit Analog to Digital Converter with 4 channels on port B
- 8 10-bit PWM/BRM Digital to Analog outputs
- Master Reset and Low Voltage Detector (LVD) reset
- Programmable Watchdog for system reliability
- Fully static operation
- 63 basic instructions / 17 main addressing modes
- 8x8 unsigned multiply instruction
- True bit manipulation
- Complete development support on PC/DOS-Windows: Real-Time Emulator, EPROM Programming Board and Gang Programmer
- Full software package (assembler, linker, C-compiler, source level debugger)

Device Summary

Features	ST72(T/E)774(J/S)9	ST72(T)754(J/S)9	ST72774(J/S)7	ST72754(J/S)7	ST72(T/E)734J6
Program Memory - Bytes	60K		48K		32K
RAM (stack) - Bytes	1K (256)				512 (256)
Peripherals	USB	No USB	USB	No USB	No USB
	ADC [3], 16-bit timer, I^2C, DDC, TMU,Sync, PWM, LVD, Watchdog				ADC[4], I^2C,LVD, DDC,Sync, 16-bit timer, PWM, Watchdog
Operating Supply	4.0V to 5.5V supply operating range				
Oscillator Frequency	12 or 24 MHz				
Operating Temperature	0 to +70°C				
Package	CSDIP42 or PSDIP42 or TQFP44				PSDIP42 CSDIP42

(1) On some devices only, refer to Device Summary; (2) Contact Sales office for availability
(3) 8-bit ±2 LSB A/D converter ; (4) 8-bit ±4 LSB A/D converter.

Revision follow-up

Changes applied since version 4.0

Version 4.0	March 2001
	Page 1:Addition of 72T774 (32KOTP).
	Addition of 60K/48K ROM for ST72754
	Deletion of table " device summary", replaced with cross reference to table 36 on page 147.
	page 13 - addition of section 1.4. external connections
Version 4.1	July 2001
	Initial format reapplied, text and related figures in the same page.
	Table "Device summary" reinserted in cover page and updated.
	Update of table 36: ordering information (p143)
Version 4.2	July 2001
	Cover - addition of feature about system protection added,
	table for device summary: addition of stack values
	page 9 - figure 3: replaced 1KByte with 512 Bytes + notes about opcode fetch and HALT mode
	page 10 - table: CR replaced by WDGCR
	TIM replaced with Timer and WDG replaced with Watchdog
	page 115 - EDF register: addition of "read from RAM", EDE: few changes
	page 135 - Note 1 replaced, note 2 added SUSpend mode limitation..
	Whole document: all mentions of HALT mode either deleted or rewritten.
Version 4.3	October 2001
	p140, chapter 8, section 8.1-
	code for unused bytes (FFh) replaced with 9Dh (opcode for NOP)
	page 141- update of table 36 "Ordering information"
	page 142 - list of available devices updated
	page 114 - DDC DCR register: bit 5 = 1, text "or read from RAM" deleted
Version 4.3	November 2001
	page 10, one adddress corrected in the figure 3 "memory map": 0400h
	page 14: addition of mandatory 1K resistor (text and figure)

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

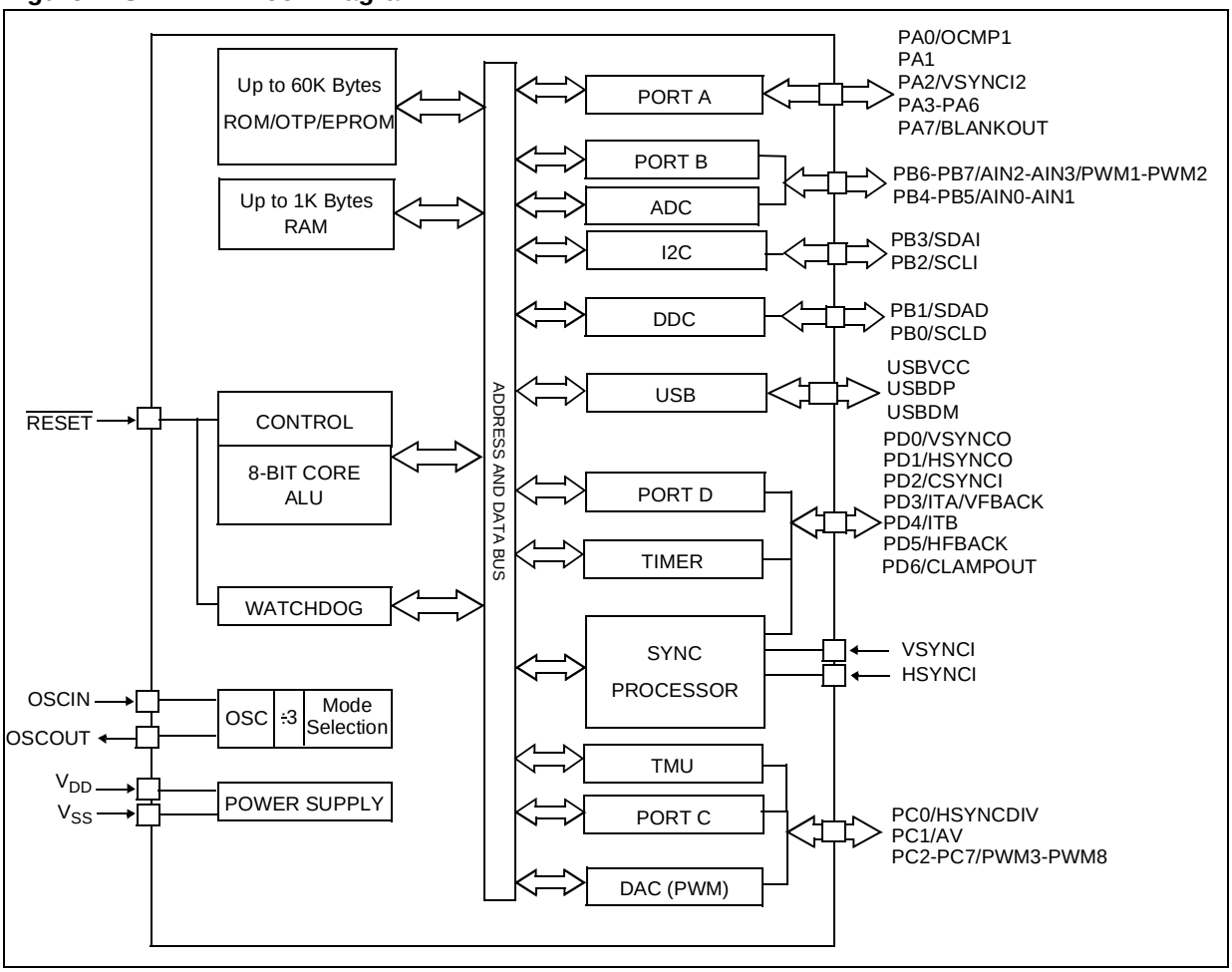
The ST72774, ST72754 and ST72734 are HCMOS microcontroller units (MCU) from the ST727x4 family with dedicated peripherals for Monitor applications.
They are based around an industry standard 8-bit core and offer an enhanced instruction set. The processor runs with an external clock at 12 or 24 MHz with a 5V supply. Due to the fully static design of this device, operation down to DC is possible. Under software control the ST727x4 can be placed in WAIT mode thus reducing power consumption. The HALT mode is no longer available.

The enhanced instruction set and addressing modes afford real programming potential. Illegal opcodes are patched and lead to a reset.

In addition to standard 8-bit data management the ST7 features true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

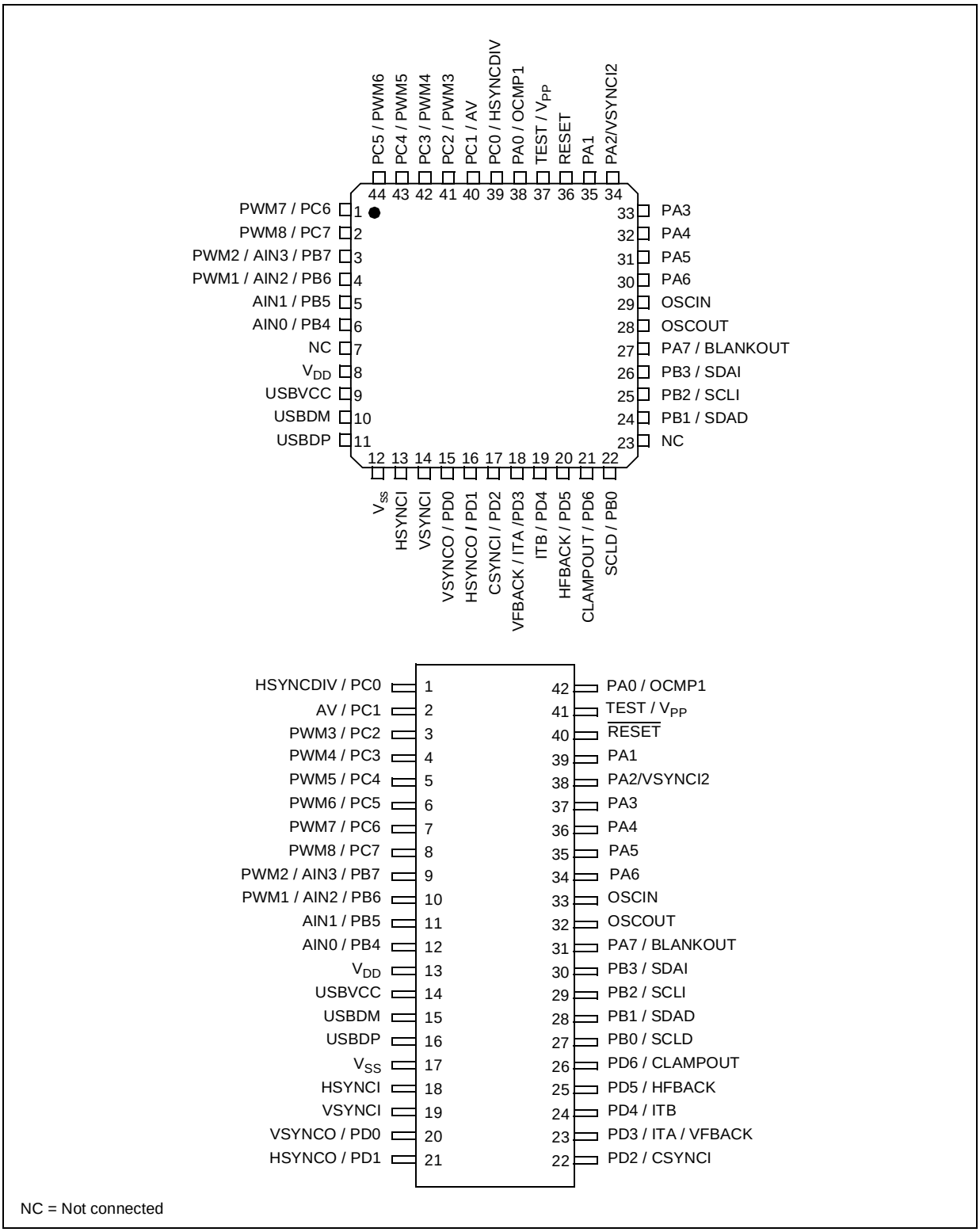
The device includes an on-chip oscillator, CPU, System protection against illegal address jumps, Sync Processor for video timing & Vfback analysis, up to 60K Program Memory, up to 1K RAM, USB/ DMA, a Timing Measurement Unit, I/O, a timer with 2 input captures and 2 output compares, a 4-channel Analog to Digital Converter, DDC, I^2C Single Master, Watchdog Reset, and eight 10-bit PWM/BRM outputs for analog DC control of external functions.

Figure 1. ST727x4 Block Diagram



1.2 PIN DESCRIPTION

Figure 2. 44-pin TQFP and 42-Pin SDIP Package Pinouts



NC = Not connected

PIN DESCRIPTION (Cont'd)

$\overline{\text{RESET}}$: Bidirectional. This active low signal forces the initialization of the MCU. This event is the top priority non maskable interrupt. This pin is switched low when the Watchdog has triggered or V_{DD} is low. It can be used to reset external peripherals.

OSCIN/OSCOUT: Input/Output Oscillator pin. These pins connect a parallel-resonant crystal, or an external source to the on-chip oscillator.

TEST/V_{PP}: Input. EPROM programming voltage. This pin must be held low during normal operating modes.

V_{DD}: Power supply voltage (4.0V-5.5V)

V_{SS}: Digital Ground.

Alternate Functions: several pins of the I/O ports assume software programmable alternate functions as shown in the pin description

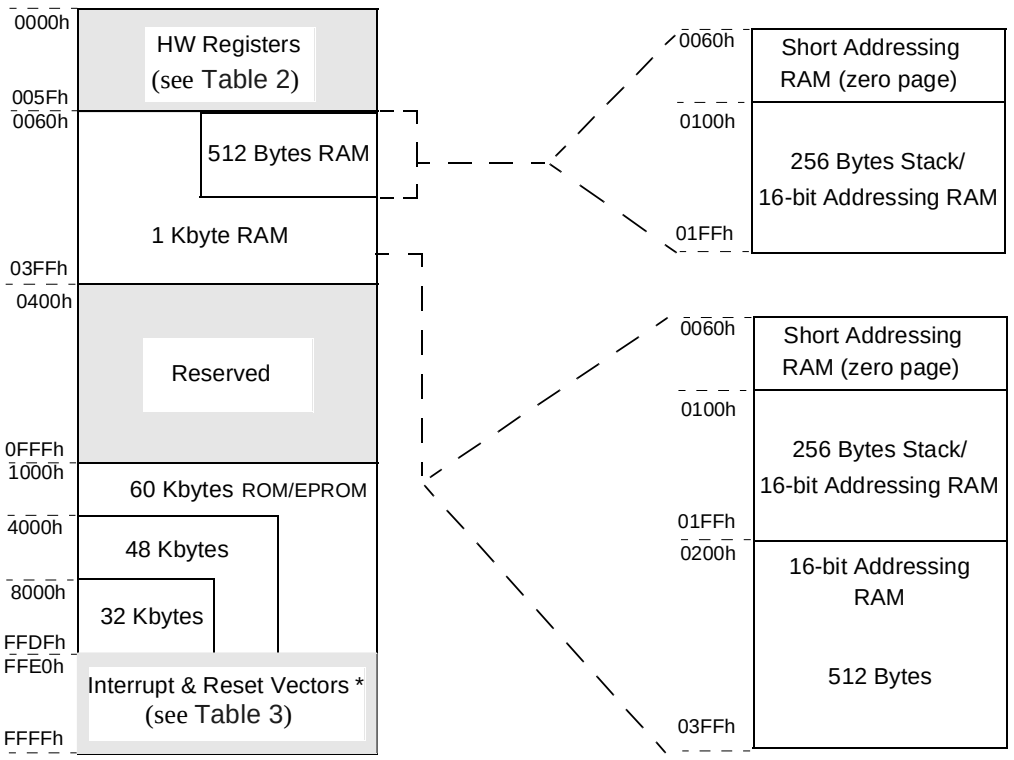
Table 1. ST727x4 Pin Description

Pin No.		Pin Name	Type	Description	Remarks
TQFP44	SDIP42				
39	1	PC0/HSYNCDIV	I/O	Port C0 or HSYNCDIV output (HSYNCO divided by 2)	
40	2	PC1/AV	I/O	Port C1 or "Active Video" input	
41	3	PC2/PWM3	I/O	Port C2 or 10-bit PWM/BRM output 3	For analog controls, after external filtering
42	4	PC3/PWM4	I/O	Port C3 or 10-bit PWM/BRM output 4	
43	5	PC4/PWM5	I/O	Port C4 or 10-bit PWM/BRM output5	
44	6	PC5/PWM6	I/O	Port C5 or 10-bit PWM/BRM output 6	
1	7	PC6/PWM7	I/O	Port C6 or 10-bit PWM/BRM output 7	
2	8	PC7/PWM8	I/O	Port C7 or 10-bit PWM/BRM output 8	
3	9	PB7/AIN3/PWM2	I/O	Port B7 or ADC analog input 3 or 10-bit PWM/BRM output 2	
4	10	PB6/AIN2/PWM1	I/O	Port B6 or ADC analog input 2 or 10-bit PWM/BRM output 1	
5	11	PB5/AIN1	I/O	Port B5 or ADC analog input 1	
6	12	PB4/AIN0	I/O	Port B4 or ADC analog input 0	
8	13	V_{DD}	S	Supply (4.0V - 5.5V)	
9	14	USBVCC	S	USB power supply (output 3.3V +/- 10%)	
10	15	USBDM	I/O	USB bidirectional data	Must be tied to ground for devices without USB peripheral
11	16	USBDP	I/O	USB bidirectional data	
12	17	V_{SS}	S	Ground 0V	
13	18	HSYNCI	I	SYNC horizontal synchronisation input	TTL levels Refer to Figure 16
14	19	VSYNCI	I	SYNC vertical synchronisation input	
15	20	PD0/VSYNCO	I/O	Port D0 or SYNC vertical synchronisation output	
16	21	PD1/HSYNCO	I/O	Port D1 or SYNC horizontal synchronisation output	
17	22	PD2/CSYNCI	I/O	Port D2 or SYNC composite synchronisation input	TTL levels with pull-up (SYNC input)

Pin No.		Pin Name	Type	Description	Remarks
TQFP44	SDIP42				
18	23	PD3/VFBACK/ITA	I/O	Port D3 or SYNC Vertical flyback input or interrupt falling edge detector input A	Refer to Figure 16 and Table 11 Port D Description
19	24	PD4/ITB	I/O	Port D4 or Interrupt falling edge detector input B	Refer to Table 11 Port D Description
20	25	PD5/HFBACK	I/O	Port D5 or SYNC horizontal flyback input	TTL levels with pull-up (SYNC input)
21	26	PD6/CLAMPOUT	I/O	Port D6 or SYNC clamping/ MOIRE output	
22	27	PB0/SCLD	I/O	Port B0 or DDC serial clock	
24	28	PB1/SDAD	I/O	Port B1 or DDC serial data	
25	29	PB2/SCLI	I/O	Port B2 or I2C serial clock	
26	30	PB3/SDAI	I/O	Port B3 or I2C serial data	
27	31	PA7/BLANKOUT	I/O	Port A7 or SYNC blanking output	
28	32	OSCOUT	O	Oscillator output	
29	33	OSCIN	I	Oscillator input	
30	34	PA6	I/O	Port A6	
31	35	PA5	I/O	Port A5	
32	36	PA4	I/O	Port A4	
33	37	PA3	I/O	Port A3	
34	38	PA2/VSYNCI2	I/O	Port A2 or SYNC vertical synchronisation input 2	DDC1 only
35	39	PA1	I/O	Port A1	
36	40	$\overline{\text{RESET}}$	I/O	Reset pin	Active low
37	41	TEST/V_{PP}	S	Test mode pin or EPROM programming voltage. This pin should be tied low in user mode.	
38	42	PA0/OCMP1	I/O	Port A0 or TIMER output compare 1	

1.3 MEMORY MAP

Figure 3. Memory Map



any opcode fetch in those areas is considered as illegal and generates a reset

(*) this block only contains addresses of interrupts and reset routines, no opcode is run from this block

MEMORY MAP (Cont'd)

Table 2. Hardware Register Memory Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h		PADR	Port A Data Register	00h	R/W
0001h		PADDR	Port A Data Direction Register	00h	R/W
0002h		PBDR	Port B Data Register	00h	R/W
0003h		PBDDR	Port B Data Direction Register	00h	R/W
0004h		PCDR	Port C Data Register	00h	R/W
0005h		PCDDR	Port C Data Direction Register	00h	R/W
0006h		PDDR	Port D Data Register	00h	R/W
0007h		PDDDR	Port D Data Direction Register	00h	R/W
0008h	Watchdog	WDGCR	Watchdog Control Register	7Fh	R/W
0009h		MISCR	Miscellaneous Register	10h	R/W
000Ah	ADC	ADCDR	ADC Data Register	00h	Read only
000Bh		ADCCSR	ADC Control Status register	00h	R/W
000Ch	DDC1/2B	DDCDCR	DDC1/2B Control Register	00h	R/W
000Dh		DDCAHR	DDC1/2B Address Pointer High Register	xxh	R/W
000Eh		TMUCSR	TMU control status register	FCh	R/W
000Fh	TMU	TMUT1CR	TMU T1 counter register	FFh	Read only
00010h		TMUT2CR	TMU T2 counter register	FFh	Read only
0011h		TIMCR2	Timer Control Register 2	00h	R/W
0012h		TIMCR1	Timer Control Register 1	00h	R/W
0013h		TIMSR	Timer Status Register	00h	Read only
0014h		TIMIC1HR	Timer Input Capture 1 High Register	xxh	Read only
0015h		TIMIC1LR	Timer Input Capture 1 Low Register	xxh	Read only
0016h		TIMOC1HR	Timer Output Compare 1 High Register	80h	R/W
0017h		TIMOC1LR	Timer Output Compare 1 Low Register	00h	R/W
0018h	Timer	TIMCHR	Timer Counter High Register	FFh	Read only
0019h		TIMCLR	Timer Counter Low Register	FCh	R/W
001Ah		TIMACHR	Timer Alternate Counter High Register	FFh	Read only
001Bh		TIMACLR	Timer Alternate Counter Low Register	FCh	R/W
001Ch		TIMIC2HR	Timer Input Capture 2 High Register	xxh	Read only
001Dh		TIMIC2LR	Timer Input Capture 2 Low Register	xxh	Read only
001Eh		TIMOC2HR	Timer Output Compare 2 High Register	80h	R/W
001Fh		TIMOC2LR	Timer Output Compare 2 Low Register	00h	R/W
0020h to 0024h	Reserved Area (5 bytes)				

Address	Block	Register Label	Register Name	Reset Status	Remarks
0025h	USB	USBPIDR	USB PID Register	XXh	Read only
0026h		USBDMAR	USB DMA address Register	XXh	R/W
0027h		USBIDR	USB Interrupt/DMA Register	XXh	R/W
0028h		USBISTR	USB Interrupt Status Register	00h	R/W
0029h		USBIMR	USB Interrupt Mask Register	00h	R/W
002Ah		USBCTLR	USB Control Register	xxxx 0110	R/W
002Bh		USBDADDR	USB Device Address Register	00h	R/W
002Ch		USBEP0RA	USB Endpoint 0 Register A	0000 xxxx	R/W
002Dh		USBEP0RB	USB Endpoint 0 Register B	80h	R/W
002Eh		USBEP1RA	USB Endpoint 1 Register A	0000 xxxx	R/W
002Fh		USBEP1RB	USB Endpoint 1 Register B	0000 xxxx0000	R/W
0030h		USBEP2RA	USB Endpoint 2 Register A	0000	R/W
0031h		USBEP2RB	USB Endpoint2 Register B	xxxx0000	R/W
0032h	PWM	PWM1	10 BIT PWM / BRM	80h	R/W
0033h		BRM21		00h	R/W
0034h		PWM2		80h	R/W
0035h		PWM3		80h	R/W
0036h		BRM43		00h	R/W
0037h		PWM4		80h	R/W
0038h		PWM5		80h	R/W
0039h		BRM65		00h	R/W
003Ah		PWM6		80h	R/W
003Bh		PWM7		80h	R/W
003Ch		BRM87		00h	R/W
003Dh		PWM8		80h	R/W
003Eh		PWMCR	PWM output enable register	00h	R/W
003Fh	Reserved Area (1 byte)				
0040h	SYNC	SYNCCFGR	SYNC Configuration Register	00h	R/W
0041h		SYNCMCR	SYNC Multiplexer Register	20h	R/W
0042h		SYNCCCR	SYNC Counter Register	00h	R/W
0043h		SYNCPOLR	SYNC Polarity Register	08h	R/W
0044h		SYNCLATR	SYNC Latch Register	00h	R/W
0045h		SYNCHGENR	SYNC H Sync Generator Register	00h	R/W
0046h		SYNCVGENR	SYNC V Sync Generator Register	00h	R/W
0047h		SYNCENR	SYNC Processor Enable Register	C3h	R/W
0048h to 004Fh	Reserved Area (8 bytes)				

Address	Block	Register Label	Register Name	Reset Status	Remarks
0050h	DDC/CI	DDCCR	DDC/CI Control Register	00h	R/W
0051h		DDCSR1	DDC/CI Status Register 1	00h	Read only
0052h		DDCSR2	DDC/CI Status Register 2	00h	Read only
0053h			Reserved		
0054h		DDCOAR	DDC/CI (7 Bits) Slave address Register	00h	R/W
0055h			Reserved		
0056h		DDCDR	DDC/CI Data Register	00h	R/W
0057h 0058h	Reserved Area (2 bytes)				
0059h	I2C	I2CDR	I2C Data Register	00h	R/W
005Ah			Reserved		
005Bh			Reserved		
005Ch		I2CCCR	I^2C Clock Control Register	00h	R/W
005Dh		I2CSR2	I^2C Status Register 2	00h	Read only
005Eh		I2CSR1	I^2C Status Register 1	00h	Read only
005Fh		I2CCR	I^2C Control Register	00h	R/W

Table 3. Interrupt Vector Map

Vector Address	Description	Remarks
FFE0-FFE1h	Not used	
FFE2-FFE3h	Not used	
FFE4-FFE5h	Not used	
FFE6-FFE7h	USB interrupt vector	Internal Interrupts
FFE8-FFE9h	Not used	
FFEA-FFEBh	I2C interrupt vector	
FFEC-FFEDh	Timer Overflow interrupt vector	
FFEE-FFEFh	Timer Output Compare interrupt vector	
FFF0-FFF1h	Timer Input Capture interrupt vector	
FFF2-FFF3h	ITA falling edge interrupt vector	External Interrupts
FFF4-FFF5h	ITB falling edge interrupt vector	
FFF6-FFF7h	DDC1/2B interrupt vector	Internal Interrupt
FFF8-FFF9h	DDC/CI interrupt vector	
FFFA-FFFBh	USB End Suspend interrupt vector	
FFFC-FFFDh	TRAP (software) interrupt vector	CPU Interrupt
FFFE-FFFFh	RESET vector	

1.4 External connections

The following figure shows the recommended external connections for the device.

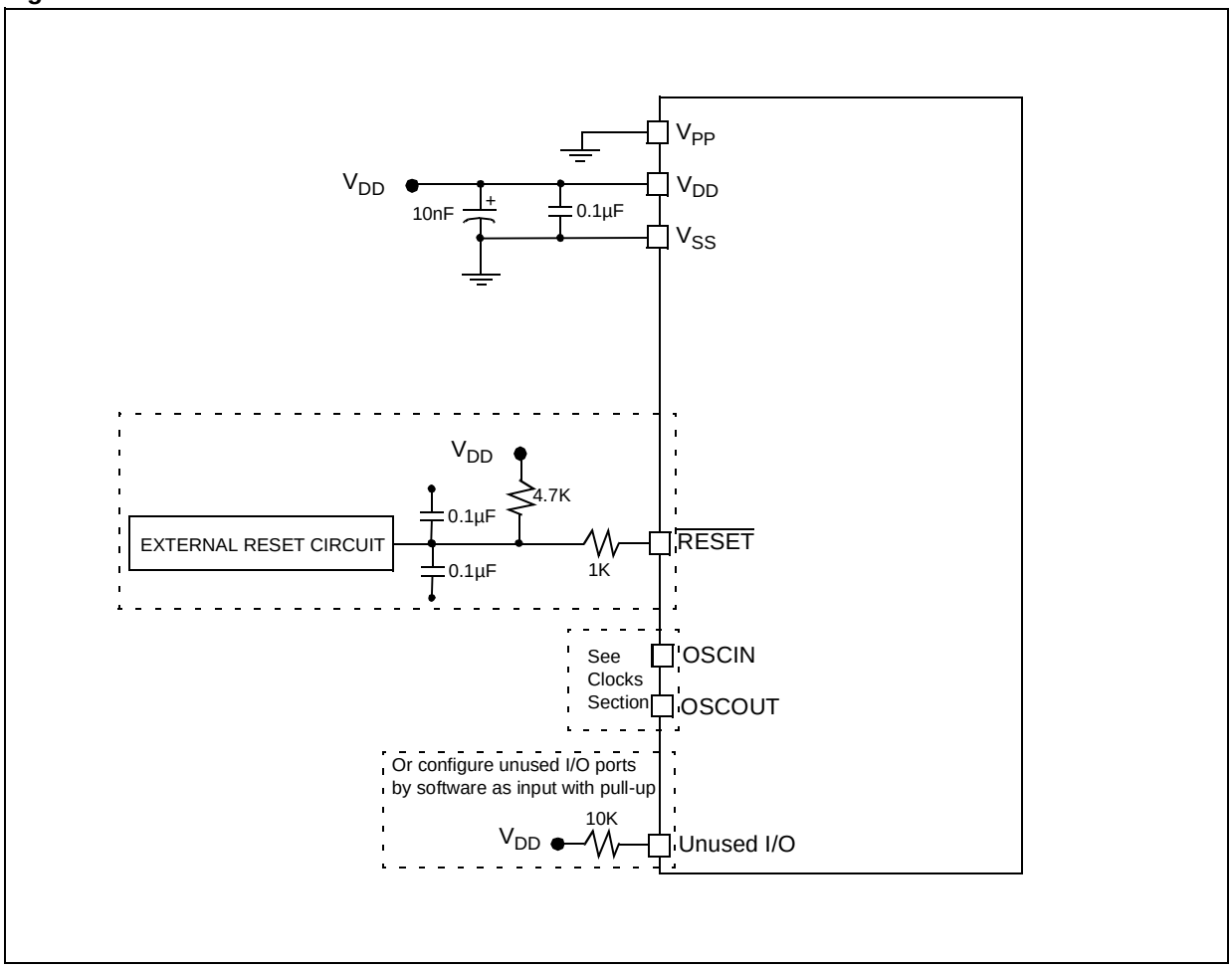
The V_{PP} pin is only used for programming OTP and EPROM devices and must be tied to ground in user mode.

The 10 nF and 0.1 µF decoupling capacitors on the power supply lines are a suggested EMC performance/cost tradeoff.

The external reset network (including the mandatory 1K serial resistor) is intended to protect the device against parasitic resets, especially in noisy environments.

Unused I/Os should be tied high to avoid any unnecessary power consumption on floating lines. An alternative solution is to program the unused ports as inputs with pull-up.

Figure 4. Recommended External Connections



2 CENTRAL PROCESSING UNIT

2.1 INTRODUCTION

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

2.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

2.3 CPU REGISTERS

The 6 CPU registers shown in Figure 5 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

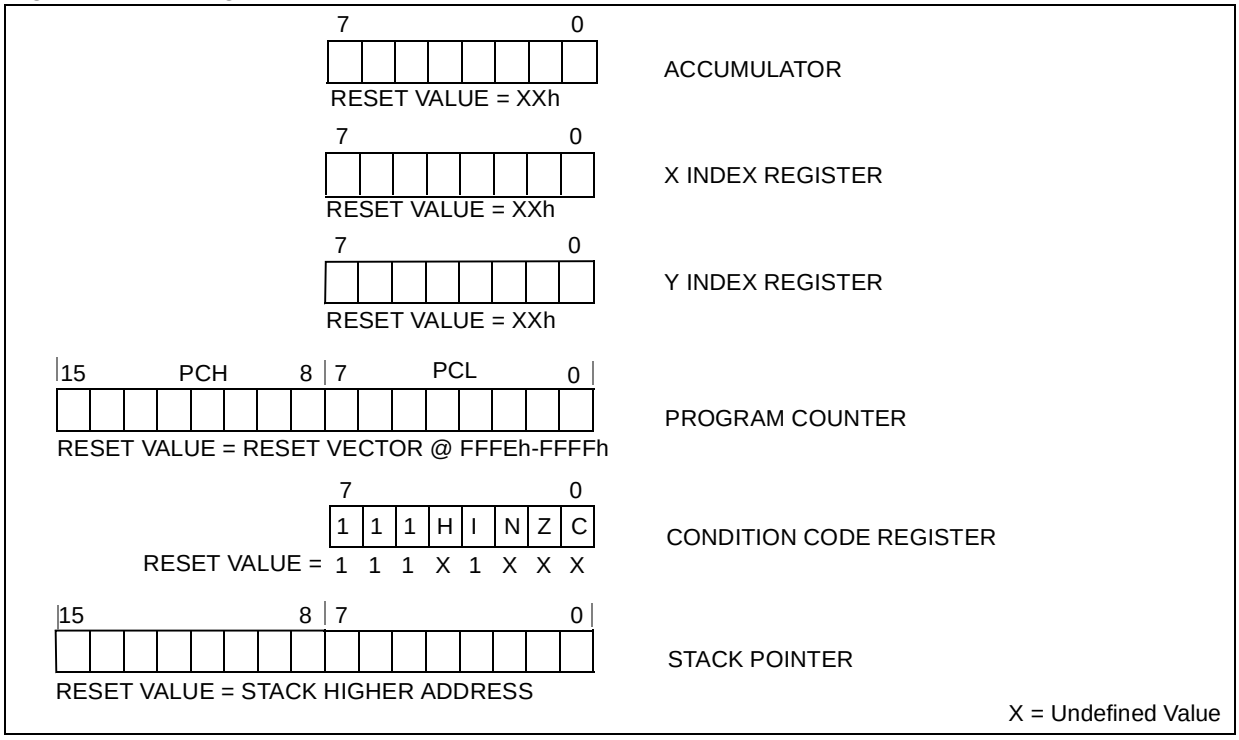
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 5. CPU Registers



CPU REGISTERS (Cont'd)
CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

The 8-bit Condition Code register contains the interrupt mask and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Bit 4 = **H** *Half carry*.

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.

0: No half carry has occurred.
1: A half carry has occurred.
This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 3 = **I** *Interrupt mask*.

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.

0: Interrupts are enabled.
1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptable because the I bit is set by hardware when you enter it and reset by the IRET instruction at the end of the interrupt routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = **N** *Negative*.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7^{th} bit of the result.

0: The result of the last operation is positive or null.
1: The result of the last operation is negative (i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero*.

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

0: The result of the last operation is different from zero.
1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = **C** *Carry/borrow.*

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.
1: An overflow or underflow has occurred.
This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

CPU REGISTERS (Cont'd)
Stack Pointer (SP)

Read/Write

Reset Value: 01 FFh

15							8
0	0	0	0	0	0	0	1

7							0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 6).

Since the stack is 256 bytes deep, the most significant byte is forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

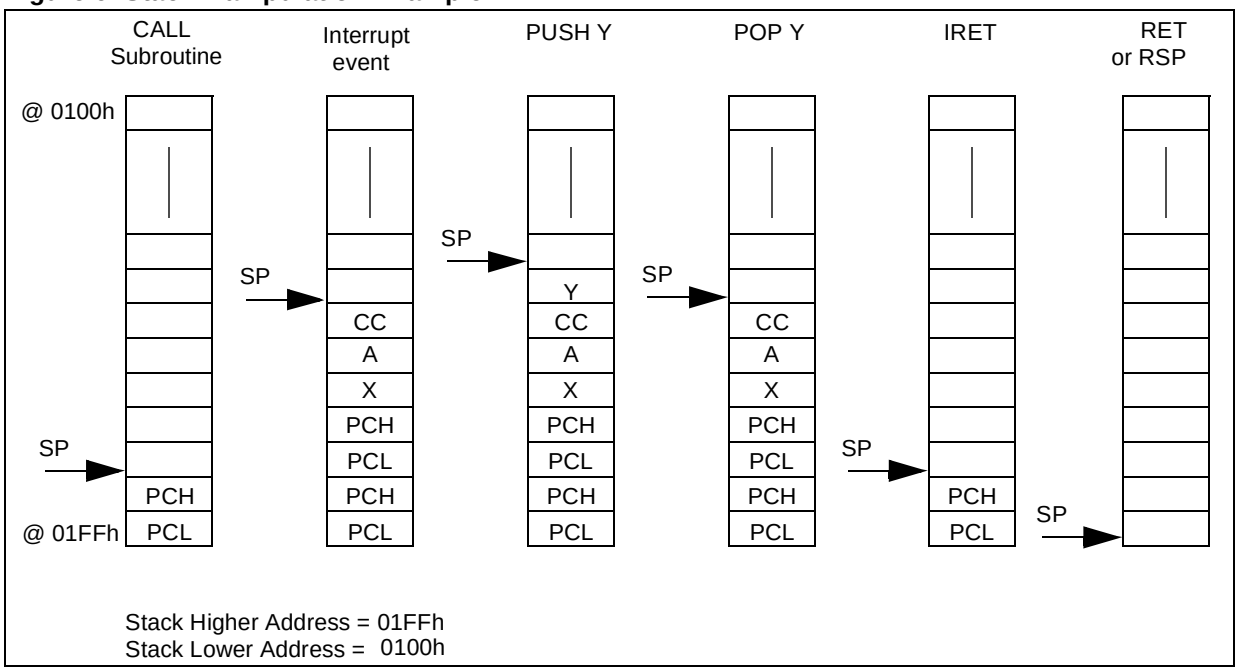
Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 6.

– When an interrupt is received, the SP is decremented and the context is pushed on the stack.
– On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 6. Stack Manipulation Example



3 CLOCKS, RESET, INTERRUPTS & LOW POWER MODES

3.1 CLOCK SYSTEM

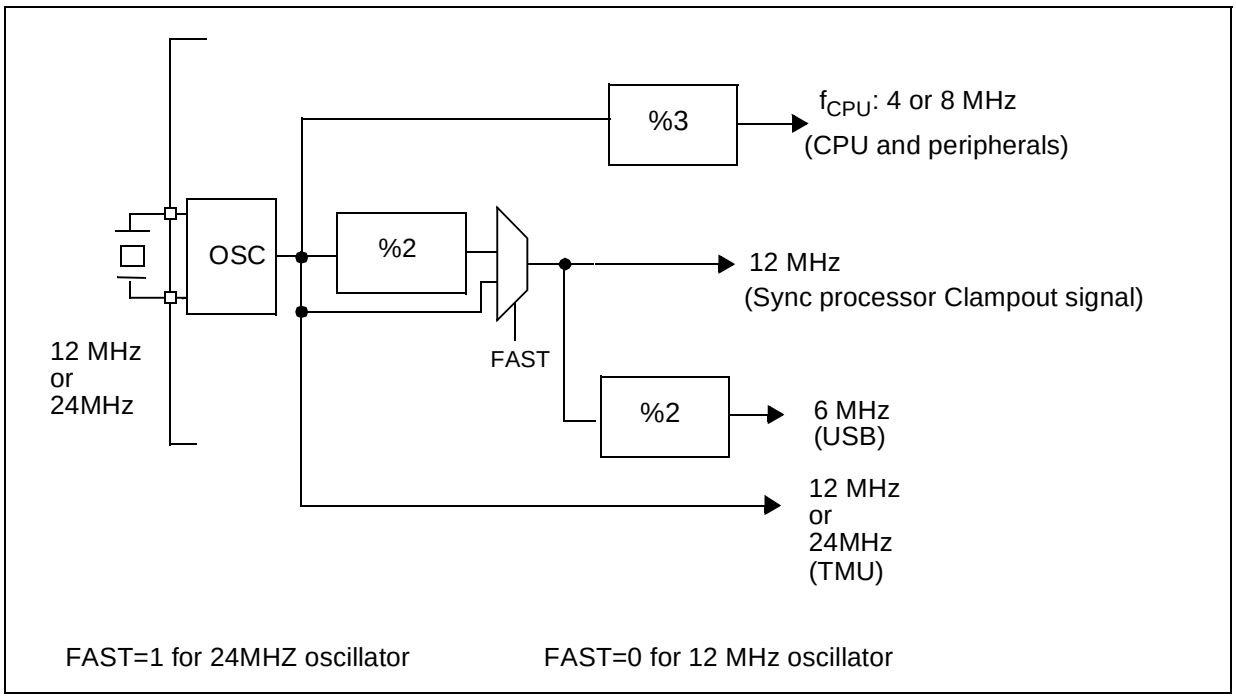
3.1.1 General Description

The MCU accepts either a crystal or an external clock signal to drive the internal oscillator. The internal clock (CPU CLK running at f_{CPU}) is derived from the external oscillator frequency (f_{OSC}), which is divided by 3. Depending on the external quartz or clock frequency, a division factor of 2 is optionally added to generate the 12 MHz clock for the Sync Processor (clamp function) as shown in Figure 7 and a second divider by 2 for the 6MHz USB clock.

The CPU clock is used also as clock for the ST727x4 peripherals.

Note: In the Sync processor, an additional divider by two is added in fast mode (same external timing for this peripheral).

Figure 7. Clock divider chain



CLOCK SYSTEM (Cont'd)

3.1.2 Crystal Resonator

The internal oscillator is designed to operate with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{osc}. The circuit shown in Figure 8 is recommended when using a crystal, and Table 4, ". Recommended Crystal Values," on page 19 lists the recommended capacitance and feedback resistance values. The crystal and associated components should be mounted as close as possible to the input pins in order to minimize output distortion and start-up stabilization time.

Figure 8. Crystal/Ceramic Resonator

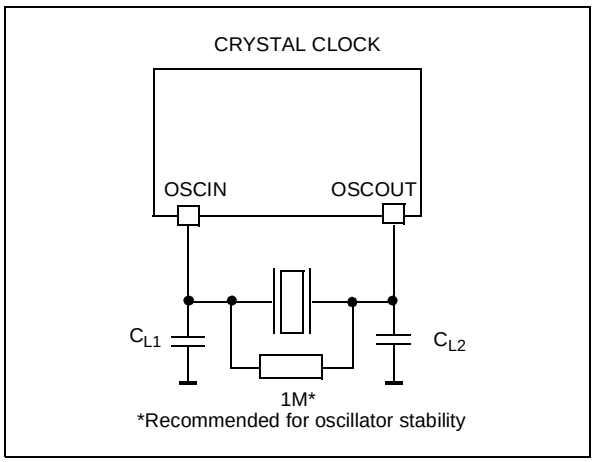


Table 4. Recommended Crystal Values

24 Mhz				Unit
R_{SMAX}	70	25	20	Ohms
C_{L1}	22	47	56	pf
C_{L2}	22	47	56	pf

Legend:

C_{L1}, C_{L2} = Maximum total capacitance on pins OSCIN and OSCOUT (the value includes the external capacitance tied to the pin plus the parasitic capacitance of the board and of the device).

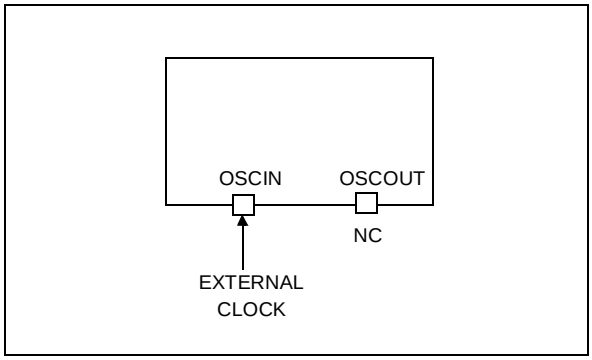
R_{SMAX} = Maximum series parasitic resistance of the quartz allowed.

Note: The tables are relative to the quartz crystal only (not ceramic resonator).

3.1.3 External Clock

An external clock should be applied to the OSCIN input with the OSCOUT pin not connected as shown in Figure 9. The Crystal clock specifications do not apply when using an external clock input. The equivalent specification of the external clock source should be used.

Figure 9. External Clock Source Connections



3.2 RESET

The Reset procedure is used to provide an orderly software start-up or to quit low power modes.

Five conditions generate a reset:

- LVD,
- watchdog,
- external pulse at the $\overline{\text{RESET}}$ pin,
- illegal address,
- illegal opcode.

A reset causes the reset vector to be fetched from addresses FFFEh and FFFFh in order to be loaded into the PC and with program execution starting from this point.

An internal circuitry provides a 4096 CPU clock cycle delay from the time that the oscillator becomes active.

3.2.1 LVD and Watchdog Reset

The Low Voltage Detector (LVD) generates a reset when V_{DD} is below V_{TRH} when V_{DD} is rising or V_{TRL} when $_{VDD}$ is falling (refer to Figure 11). This circuitry is active only when V_{DD} is above V_{TRM}.

During LVD Reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

When a watchdog reset occurs, the $\overline{\text{RESET}}$ pin is pulled low permitting the MCU to reset other devices as when Power on/off (Figure 10).

3.2.2 External Reset

The external reset is an active low input signal applied to the RESET pin of the MCU.
As shown in Figure 12, the $\overline{\text{RESET}}$ signal must remain low for 1000ns.

An internal Schmitt trigger at the $\overline{\text{RESET}}$ pin is provided to improve noise immunity.

3.2.3 Illegal Address Detection

An opcode fetch from an illegal address (refer to Figure 3) generates an illegal address reset. Program execution at those addresses is forbidden (especially to protect page 0 registers against spurious accesses).

3.2.4 Illegal Opcode Detection

Illegal instructions corresponding to no valid opcode generate a reset. Refer to ST7 Programming Manual.

Table 5. List of sections affected by RESET and WAIT (Refer to 3.6 for Wait Mode)

Section	RESET	WAIT
Fast bit of the miscellaneous register set to one (24 MHz as external clock)	X	
Timer Prescaler reset to zero	X	
Timer Counter set to FFFCh	X	
All Timer enable bits set to 0 (disabled)	X	
Data Direction Registers set to 0 (as Inputs)	X	
Set Stack Pointer to 01FFh	X	
Force Internal Address Bus to restart vector FFFEh, FFFFh	X	
Set Interrupt Mask Bit (I-Bit, CC) to 1 (Interrupt disable)	X	
Set Interrupt Mask Bit (I-Bit, CC) to 0 (Interrupt enable)		X
Reset WAIT latch	X	
Disable Oscillator (for 4096 cycles)	X	
Set Timer Clock to 0	X	
Watchdog counter reset	X	
Watchdog register reset	X	
Port data registers reset	X	
Other on-chip peripherals: registers reset	X	

RESET (Cont'd)

Figure 10. Low Voltage Detector Functional Diagram

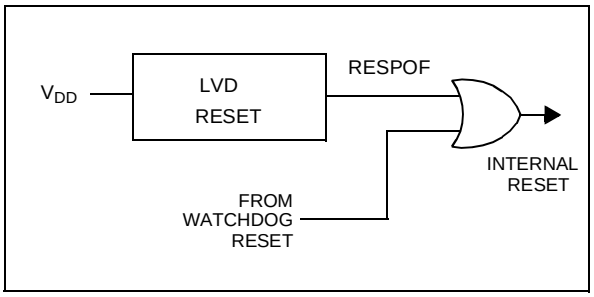
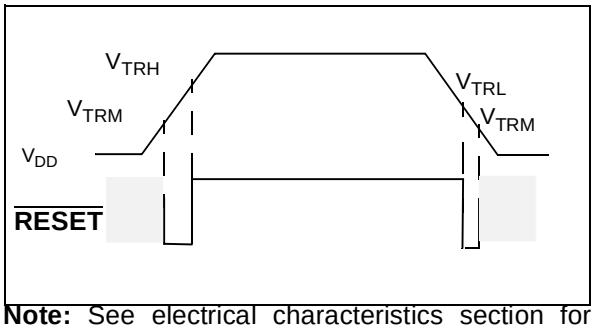
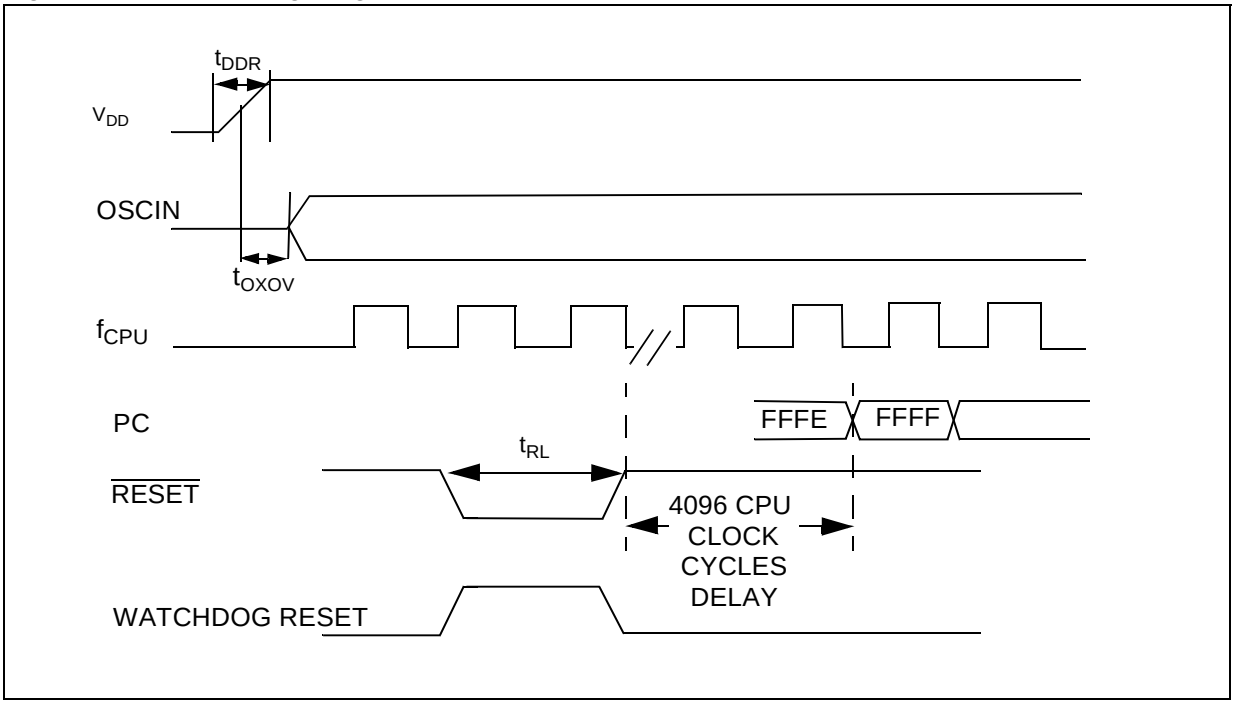


Figure 11. LVD Reset Signal Output



Note: See electrical characteristics section for values of V_{TRH}, V_{TRL} and V_{TRM}

Figure 12. Reset Timing Diagram



Note: Refer to Electrical Characteristics for values of t_{DDR}, t_{OXOV} and t_{RL}

3.3 INTERRUPTS

The ST727x4 may be interrupted by one of two different methods: maskable hardware interrupts as listed in Table 6 and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 13.

The maskable interrupts must be enabled in order to be serviced. However, disabled interrupts can be latched and processed when they are enabled. When an interrupt has to be serviced, the PC, X, A and CC registers are saved onto the stack and the interrupt mask (I bit of the Condition Code Register) is set to prevent additional interrupts. The Y register is not automatically saved.

The PC is then loaded with the interrupt vector of the interrupt to service and the interrupt service routine runs (refer to Table 6, “Interrupt Mapping,” on page 24 for vector addresses). The interrupt service routine should finish with the IRET instruction which causes the contents of the registers to be recovered from the stack and normal processing to resume. Note that the I bit is then cleared if and only if the corresponding bit stored in the stack is zero.

Though many interrupts can be simultaneously pending, a priority order is defined (see Table 6, “Interrupt Mapping,” on page 24). The $\overline{\text{RESET}}$ pin has the highest priority.

If the I bit is set, only the TRAP interrupt is enabled.

All interrupts allow the processor to leave the WAIT low power mode.

Software Interrupt. The software interrupt is the executable instruction TRAP. The interrupt is recognized when the TRAP instruction is executed, regardless of the state of the I bit. When the interrupt is recognized, it is serviced according to the flowchart on Figure 13.

ITA, ITB interrupts. The ITA (PD3), ITB (PD4), pins can generate an interrupt when a falling edge occurs on these pins, if these interrupts are enabled with the ITAITE, ITBITE bits respectively in the miscellaneous register and the I bit of the CC register is reset. When an enabled interrupt occurs, normal processing is suspended at the end of the current instruction execution. It is then serviced according to the flowchart on Figure 13. Software in the ITA or ITB service routine must reset the cause of this interrupt by clearing the ITALAT, ITBLAT or ITAITE, ITBITE bits in the miscellaneous register.

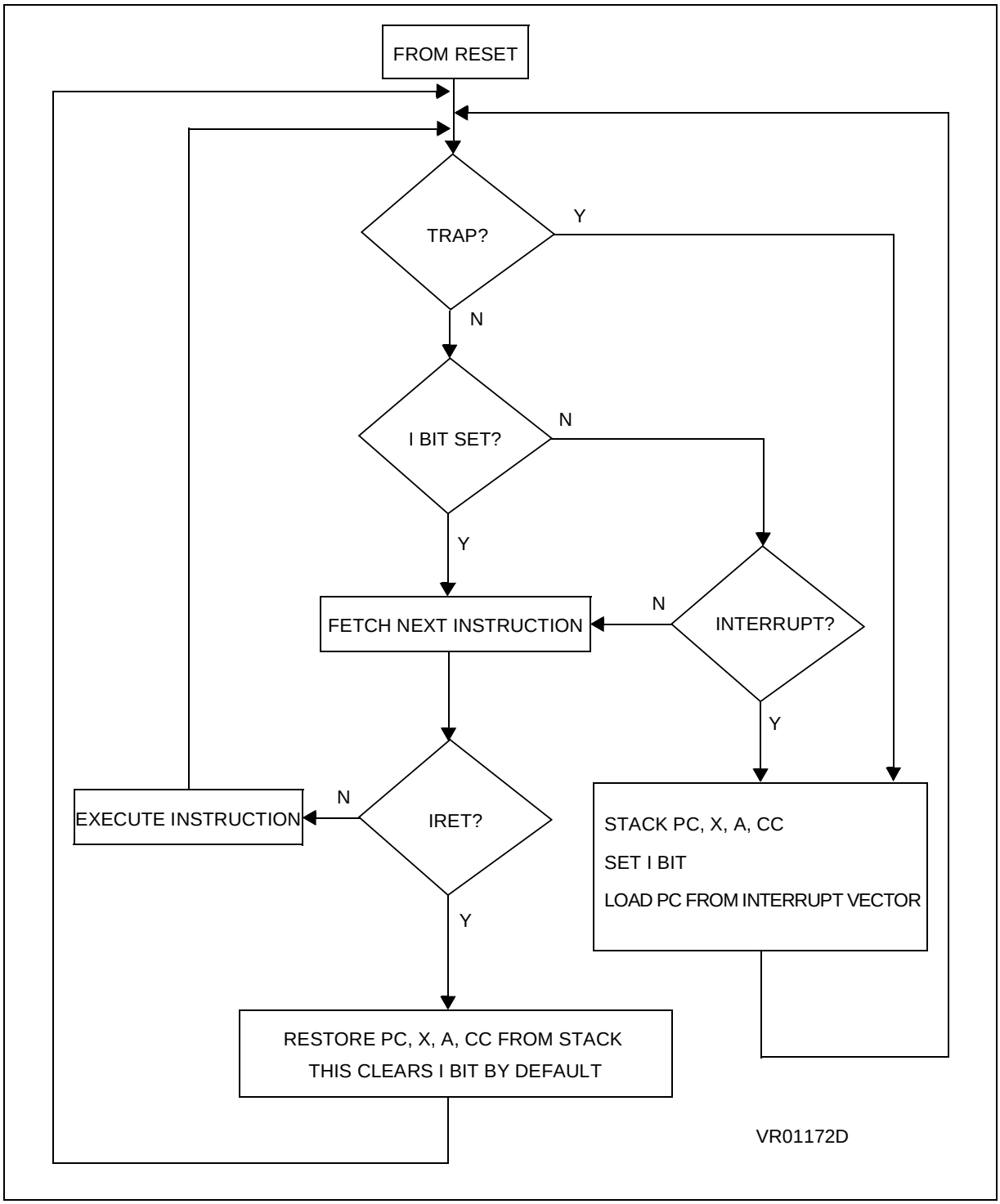
Peripheral Interrupts. Different peripheral interrupt flags are able to cause an interrupt when they are active if both the I bit of the CC register is reset and if the corresponding enable bit is set. If either of these conditions is false, the interrupt is latched and thus remains pending.

The interrupt flags are located in the status register. The Enable bits are in the control register. When an enabled interrupt occurs, normal processing is suspended at the end of the current instruction execution. It is then serviced according to the flowchart on Figure 13.

The general sequence for clearing an interrupt is an access to the status register while the flag is set followed by a read or write of an associated register. Note that the clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being enabled) will therefore be lost if the clear sequence is executed.

INTERRUPTS (Cont'd)

Figure 13. Interrupt Processing Flowchart



INTERRUPTS (Cont'd)

Table 6. Interrupt Mapping

Source Block	Description	Register Label	Flag	Maskable by I-bit	Vector Address	Priority Order
RESET	Reset	N/A	N/A	no	FFFEh-FFFFh	Highest Priority
TRAP	Software	N/A	N/A	no	FFFCh-FFFDh	
USB	End Suspend Interrupt	USBISTR	ESUSP	yes	FFFAh-FFFBh	
DDC/CI	DDC/CI Interrupt	DDCSR1 DDCSR2	**	yes	FFF8h-FFF9h	
DDC1/2B	DDC1/2B Interrupt	DDCDCR	EDF	yes	FFF6h-FFF7h	
Port D bit 4	External Interrupt ITB	MISCR	ITBLAT	yes	FFF4h-FFF5h	
Port D bit 3	External Interrupt ITA		ITALAT	yes	FFF2h-FFF3h	
TIM	Input Capture 1	TIMSR	ICF1	yes	FFF0h-FFF1h	
	Input Capture 2		ICF2			
	Output Compare 1		OCF1	yes	FFEEh-FFEFh	
	Output Compare 2		OCF2			
	Timer Overflow		TOF	yes	FFECh-FFEDh	
I2C	I2C Peripheral Interrupts	I2CSR1 I2CSR2	**	yes	FFEAh-FFEBh	
USB	USB Interrupt	USBISTR	**	yes	FFE6h-FFE7h	Lowest Priority

** Many flags can cause an interrupt, see peripheral interrupt status register description.

3.4 POWER SAVING MODES

3.4.1 WAIT Mode

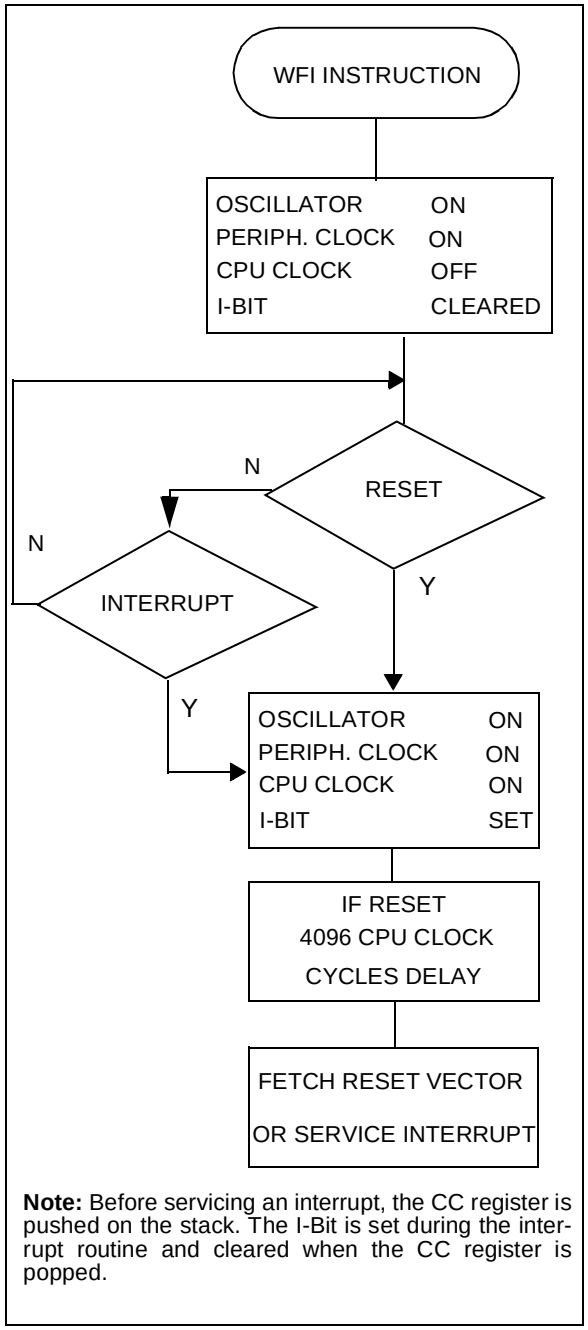
This mode is a low power consumption mode. The WFI instruction places the MCU in WAIT mode: The internal clock remains active but all CPU processing is stopped; however, all other peripherals are still running.

Note: In WAIT mode, DMA accesses (DDC, USB) are possible.

During WAIT mode, the I bit in the condition code register is cleared to enable all interrupts, which causes the MCU to exit WAIT mode, causes the corresponding interrupt vector to be fetched, the interrupt routine to be executed and normal processing to resume. A reset causes the program counter to fetch the reset vector and processing starts as for a normal reset.

Table 5 gives a list of the different sections affected by the low power modes. For detailed information on a particular device, please refer to the corresponding part.

Figure 14. WAIT Flow Chart



Note: Before servicing an interrupt, the CC register is pushed on the stack. The I-Bit is set during the interrupt routine and cleared when the CC register is popped.

3.4.2 HALT Mode

The HALT mode is the MCU lowest power consumption mode. Meanwhile, the HALT mode also stops the oscillator stage completely which is the most critical condition in CRT monitors.

For this reason, the HALT mode has been disabled and its associated HALT instruction is now considered as illegal and will generate a reset.

3.5 MISCELLANEOUS REGISTER

MISCELLANEOUS REGISTER (MISCR)
Address: 0009h — Read/Write
Reset Value: 0001 0000 (10h)

7							0
VSYNCSEL	FLY_SYN	HSYNCDIVEN	FAST	ITBLAT	ITALAT	ITBITE	ITAITE

Bit 7= **VSYNCSEL** *DDC1 VSYNC Selection.*

This bit is set and cleared by software. It is used to choose the VSYNC signal in DDC1 mode.

0: VSYNCI selected
1: VSYNCI2 selected

Note: VSYNCI 2 is only available for the DDC cell, not for the SYNC processor cell.

Bit 6= **FLY_SYN** *Flyback or Synchro Switch.*

This bit is set and cleared by software. It is used to choose the signals the Timing Measurement Unit (TMU) will analyse.

0: horizontal and vertical synchro outputs analysis
1: horizontal and vertical Flyback inputs analysis

Bit 5= **HSYNCDIVEN** *HSYNCDIV Enable.*

This bit is set and cleared by software. It is used to enable the output of the HSYNCO output on PC0.

0: HSYNCDIV disabled
1:HSYNCDIV enabled

Bit 4= **FAST** *Fast Mode.*

This bit is set and cleared by software. It is used to select the external clock frequency. If the external clock frequency is 12 MHz, this bit must be at 0, else if the external frequency is 24 MHz, this bit must be at 1.

Bit 3= **ITBLAT** *Falling Edge Detector Latch.*
This bit is set by hardware when a falling edge occurs on pin ITB/PD4 in Port D. An interrupt is generated if ITBITE=1and the I bit in the CC register = 0. It is cleared by software.

0: No falling edge detected on ITB
1: Falling edge detected on ITB

Bit 2= **ITALAT** *Falling Edge Detector Latch.*
This bit is set by hardware when a falling edge occurs on pin ITA/PD3 in Port D. An interrupt is generated if ITAITE=1and the I bit in the CC register = 0. It is cleared by software.

0: No falling edge detected on ITA
1: Falling edge detected on ITA

Bit 1= **ITBITE** *ITB Interrupt Enable*.
This bit is set and cleared by software.

0: ITB interrupt disabled
1: ITB interrupt enabled

Bit 0= **ITAITE** *ITA Interrupt Enable*.
This bit is set and cleared by software.

0: ITA interrupt disabled
1: ITA interrupt enabled

4 ON-CHIP PERIPHERALS

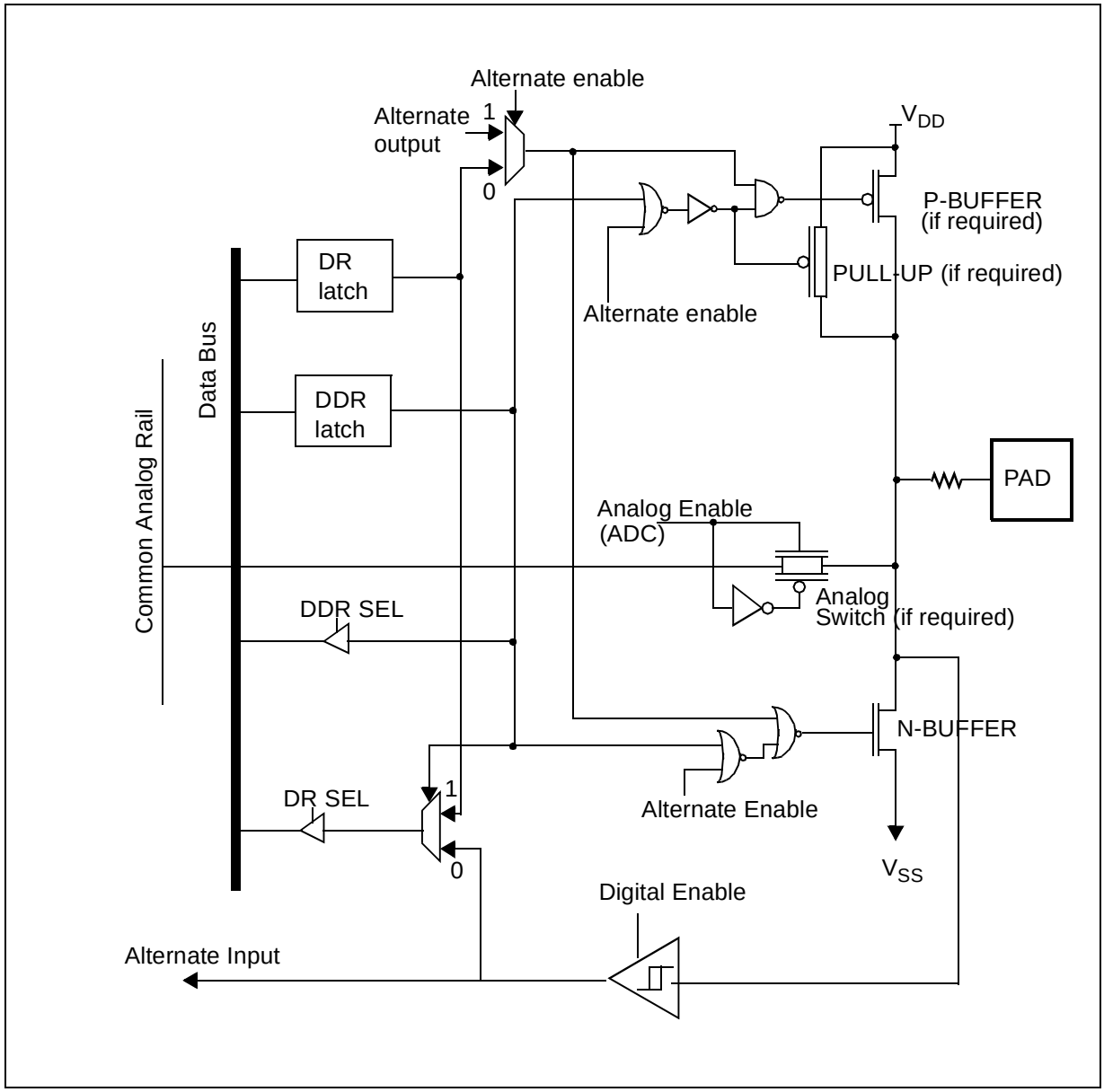
4.1 I/O PORTS

4.1.1 Introduction

The I/O ports allow the transfer of data through digital inputs and outputs, and, for specific pins, the input of analog signals or the Input/Output of alternate signals for on-chip peripherals (DDC, TIMER...).

Each pin can be programmed independently as digital input or digital output. Each pin can be an analog input when an analog switch is connected to the Analog to Digital Converter (ADC).

Figure 15. I/O Pin Typical Circuit



Note: This is the typical I/O pin configuration. For cost optimization, each port is customized with a specific configuration.

I/O PORTS (Cont'd)

Table 7. I/O Pin Functions

DDR	MODE
0	Input
1	Output

4.1.2 Common Functional Description

Each port pin of the I/O Ports can be individually configured under software control as either input or output.

Each bit of a Data Direction Register (DDR) corresponds to an I/O pin of the associated port. This corresponding bit must be set to configure its associated pin as output and must be cleared to configure its associated pin as input (Table 7, ". I/O Pin Functions," on page 28). The Data Direction Registers can be read and written.

The typical I/O circuit is shown on Figure 15. Any write to an I/O port updates the port data register even if it is configured as input. Any read of an I/O port returns either the data latched in the port data register (pins configured as output) or the value of the I/O pins (pins configured as input).

Remark: when an I/O pin does not exist inside an I/O port, the returned value is a logic one (pin configured as input).

At reset, all DDR registers are cleared, which configures all port's I/Os as inputs with or without pull-ups (see Table 8 to Table 12 I/O Ports Register Map). The Data Registers (DR) are also initialized at reset.

4.1.2.1 Input mode

When DDR=0, the corresponding I/O is configured in Input mode.
In this case, the output buffer is switched off, the state of the I/O is readable through the Data Register address, but the I/O state comes directly from the CMOS Schmitt Trigger output and not from the Data Register output.

4.1.2.2 Output mode

When DDR=1, the corresponding I/O is configured in Output mode.
In this case, the output buffer is activated according to the Data Register's content.
A read operation is directly performed from the Data Register output.

4.1.2.3 Analog input

Each I/O can be used as analog input by adding an analog switch driven by the ADC.
The I/O must be configured in Input before using it as analog input.
The CMOS Schmitt trigger is OFF and the analog value directly input through an analog switch to the Analog to Digital Converter, when the analog channel is selected by the ADC.

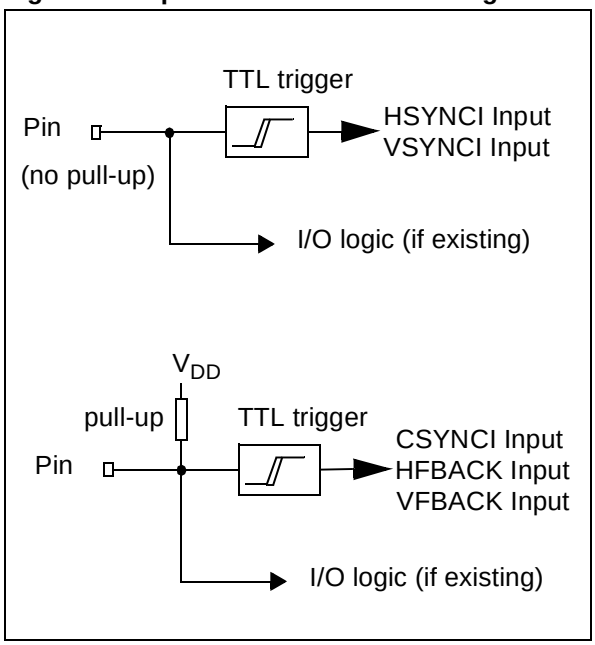
4.1.2.4 Alternate mode

A signal coming from a on-chip peripheral can be output on the I/O.
In this case, the I/O is automatically configured in output mode.
This must be controlled directly by the peripheral with a signal coming from the peripheral which enables the alternate signal to be output.

A signal coming from an I/O can be input in a on-chip peripheral.
Before using an I/O as Alternate Input, it must be configured in Input mode (DDR=0). So both Alternate Input configuration and I/O Input configuration are the same (with or without pull-up). The signal to be input in the peripheral is taken after the CMOS Schmitt trigger or TTL Schmitt trigger for SYNC.

The I/O state is readable as in Input mode by addressing the corresponding I/O Data Register.

Figure 16. Input Structure for SYNC signals



4.1.3 Port A

PA7 and PA[2:0] can be defined as Input lines (with pull-up) or as Push-pull Outputs.

PA [6:3] can be defined as Input lines (without pull-up) or as Output Open drain lines.

PA7 and PA[2:0] can be defined as Input lines (with pull-up) or as Push-pull Outputs.

PA [6:3] can be defined as Input lines (without pull-up) or as Output Open drain lines.

Table 8. Port A Description

PORT A	I / O		Alternate Function	
	Input*	Output	Signal	Condition
PA0	With pull-up	push-pull	OCMP1	OC1E =1 (CR2[TIMER])
PA1	With pull-up	push-pull	-	-
PA2	With pull-up	push-pull	VSYNCI2	VSYNCSEL=1 (MISCR)
PA3	Without pull-up	open-drain	-	-
PA4	Without pull-up	open-drain	-	-
PA5	Without pull-up	open-drain	-	-
PA6	Without pull-up	open-drain	-	-
PA7	With pull-up	push-pull	BLANKOUT	BLKEN = 1 (ENR[SYNC])

*Reset State

I/O PORTS (Cont'd)

Figure 17. PA0 to PA2, PA7

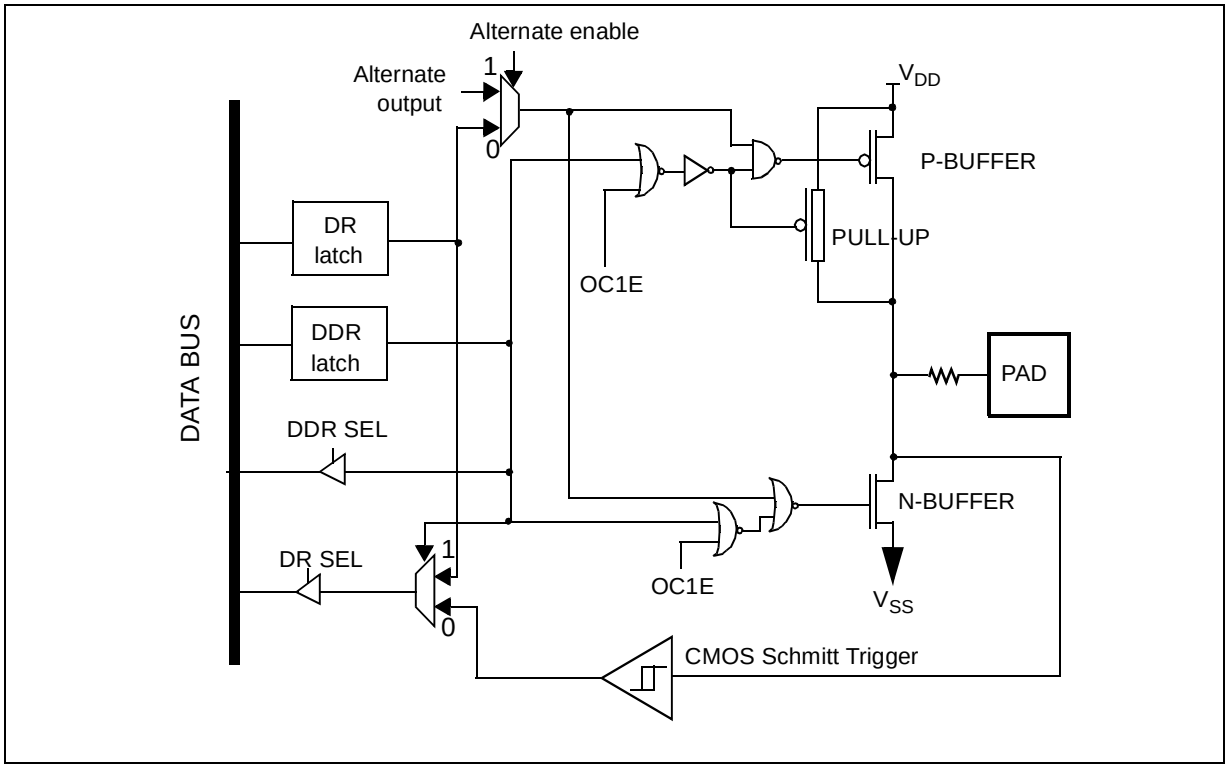
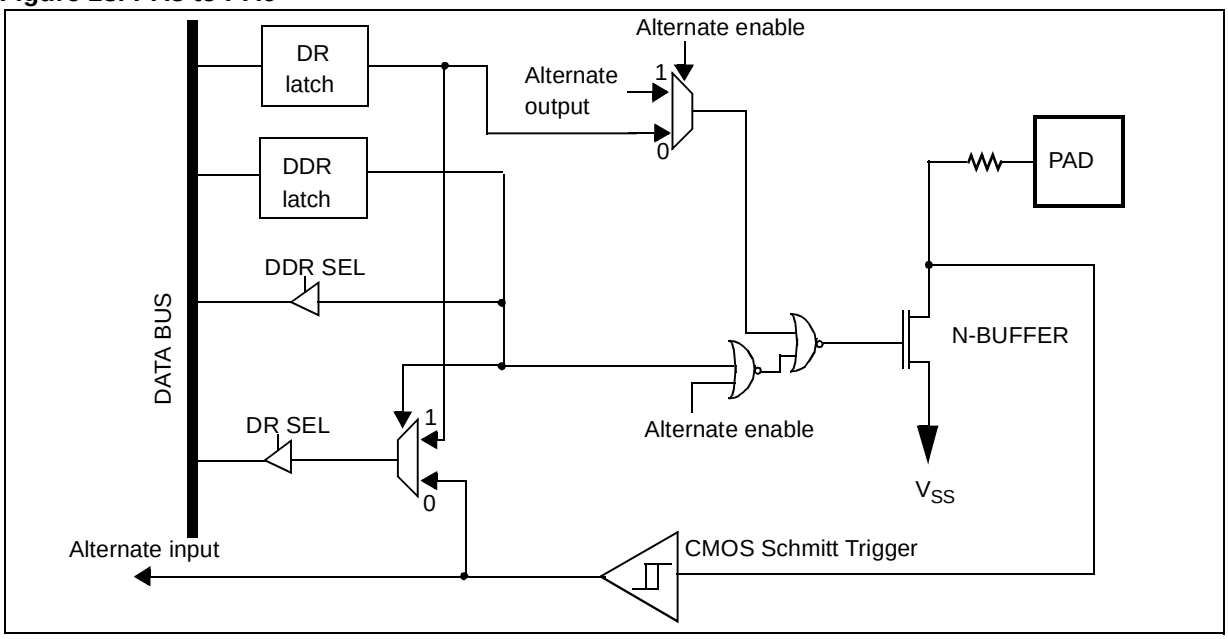


Figure 18. PA3 to PA6



I/O PORTS (Cont'd)

4.1.4 Port B

The alternate functions are the I/O pins of the on-chip DDC SCLD & SCDAD for PB0:1, the I/O pins of the on-chip I2C SCLI & SCDAI for PB2:3, and 4 bits of port B bit can be used as the Analog source to the Analog to Digital Converter.

Only one I/O line must be configured as an analog input at any time. The user must avoid any situation in which more than one I/O pin is selected as an analog input simultaneously to avoid device malfunction.

When the analog function is selected for an I/O pin, the pull-up of the respective pin of Port B is disconnected and the digital input is off.

All unused I/O lines should be tied to an appropriate logic level (either V_{DD} or V_{SS})

Since the ADC is on the same chip as the microprocessor, the user should not switch heavily loaded signals during conversion, if high precision is required. Such switching will affect the supply voltages used as analog references. the accuracy of the conversion depends on the quality of the power supplies (V_{DD} and V_{SS}). The user must take special care to ensure that a well regulated reference voltage is present on the V_{DD} and V_{SS} pins (power supply variations must be less than 5V/ms). This implies, in particular, that a suitable decoupling capacitor is used at the V_{DD} pin.

Table 9. Port B Description

PORT B	I/O		Alternate Function	
	Input*	Output	Signal	Condition
PB0	Without pull-up	Open-drain	SCLD (input with CMOS schmitt trigger or open drain output)	DDC enable
PB1	Without pull-up	Open-drain	SDAD (input with CMOS schmitt trigger or open drain output)	DDC enable
PB2	Without pull-up	Open-drain	SCLI (input with CMOS schmitt trigger or open drain output)	I2C enable
PB3	Without pull-up	Open-drain	SDAI (input with CMOS schmitt trigger or open drain output)	I2C enable
PB4	With pull-up	Push-pull	Analog input (ADC) (without pull-up)	CH[2:0]=000 (ADCCSR)
PB5	With pull-up	Push-pull	Analog input (ADC) (without pull-up)	CH[2:0]=001 (ADCCSR)
PB6	With pull-up	Push-pull	Analog input (ADC) (without pull-up)	CH[2:0]=010 (ADCCSR)
			10-bit output 1 (PWM)	OE0=1 (PWMOE)
PB7	With pull-up	Push-pull	Analog input (ADC) (without pull-up)	CH[2:0]=011 (ADCCSR)
			10-bit output 2 (PWM)	OE1=1 (PWMOE)

*Reset state

I/O PORTS (Cont'd)

Figure 19. PB0 to PB3

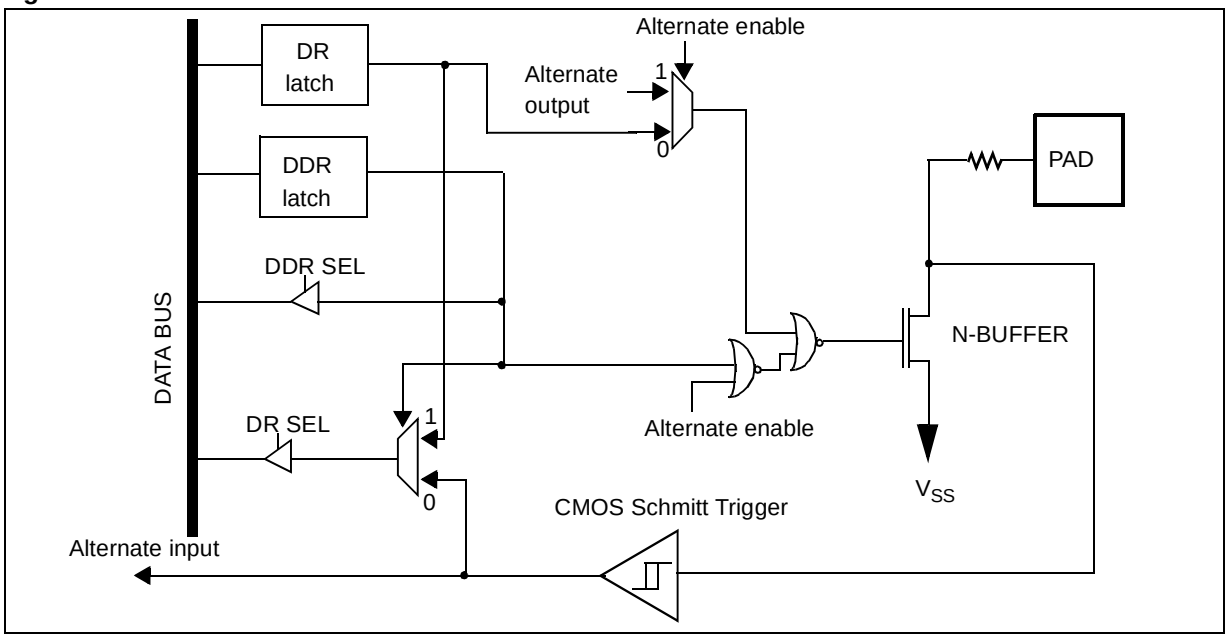
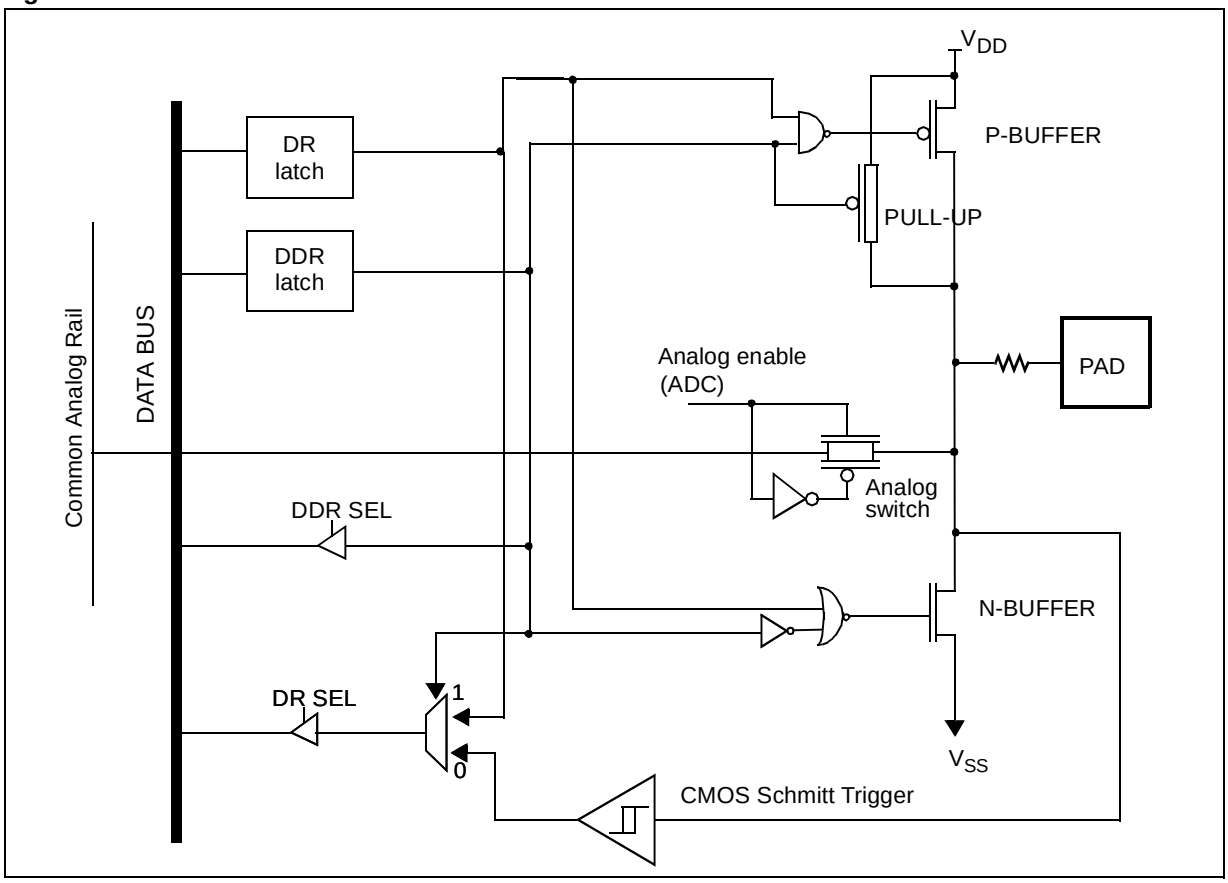


Figure 20. PB4 to PB7



I/O PORTS (Cont'd)

4.1.5 Port C

The available port pins of port C may be used as general purpose I/O.

The alternate functions are the PWM outputs for PC2:7, HSYNCDIV (HSYNCO divided by 2) for PC0 and the TMU input for PC1.

Table 10. Port C Description

PORT C	I / O		Alternate Function	
	Input*	Output	Signal	Condition
PC0	With pull-up	Push-pull	HSYNCDIV (push-pull)	HSYNCDIVEN =1 (MISCR)
PC1	With pull-up	Push-pull	AV (active video) input (TMU)	-
PC2	With pull-up	Push-pull	10-bit output 3 (PWM)	OE2=1 (PWMOE)
PC3	With pull-up	Push-pull	10-bit output 4 (PWM)	OE3=1 (PWMOE)
PC4	With pull-up	Push-pull	10-bit output 5 (PWM)	OE4=1 (PWMOE)
PC5	With pull-up	Push-pull	10-bit output 6 (PWM)	OE5=1 (PWMOE)
PC6	With pull-up	Push-pull	10-bit output 7 (PWM)	OE6=1 (PWMOE)
PC7	With pull-up	Push-pull	10-bit output 8 (PWM)	OE7=1 (PWMOE)

* Reset State

I/O PORTS (Cont'd)

Figure 21. PC0, PC2 to PC7

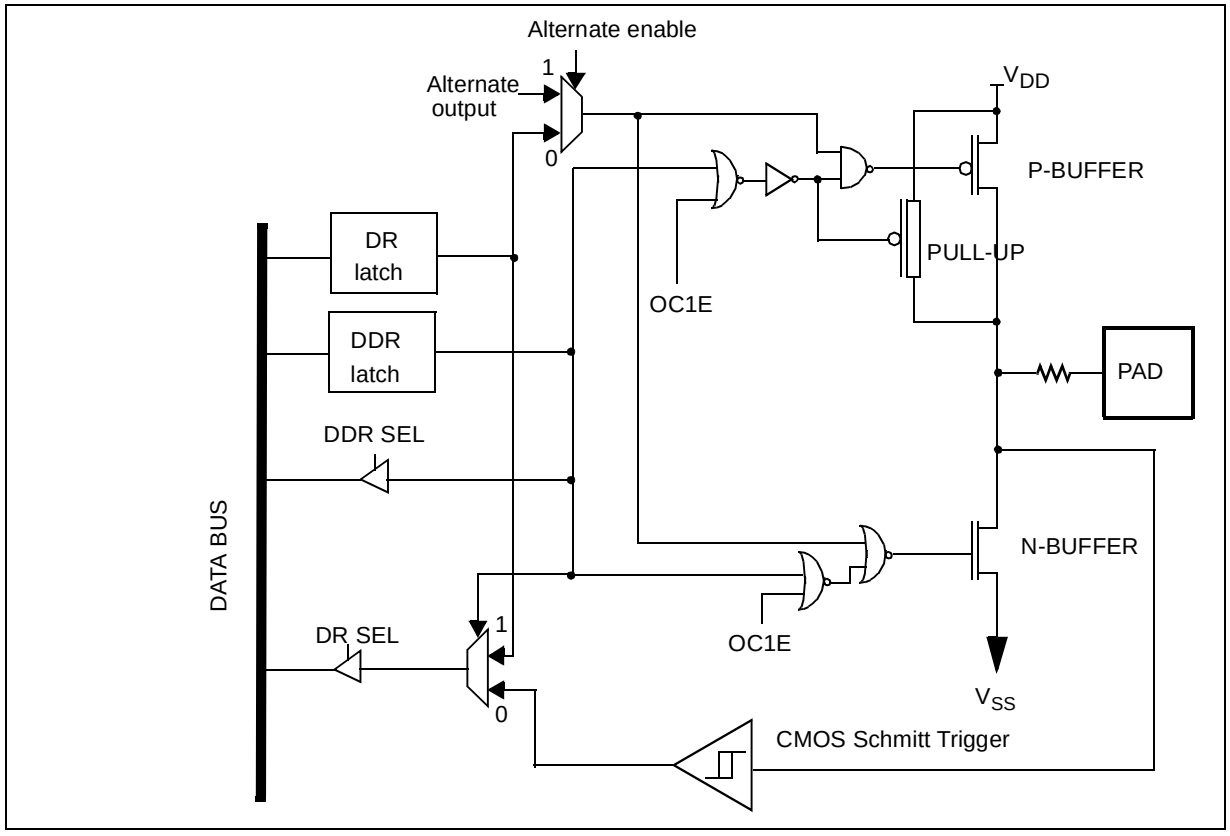
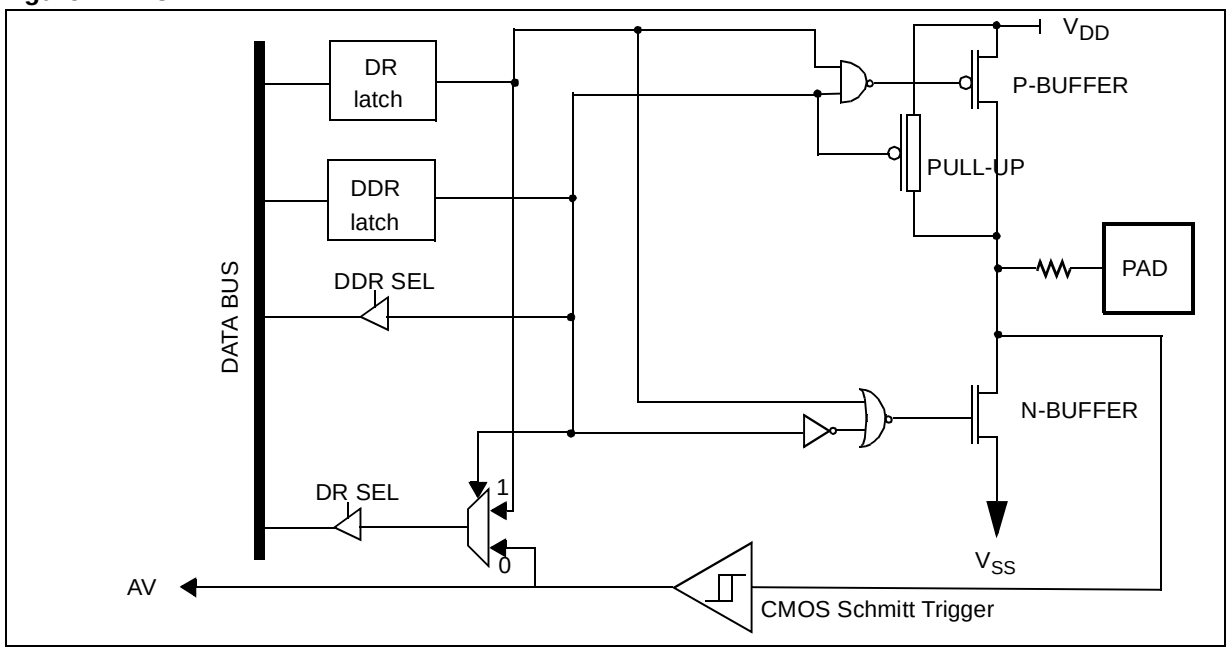


Figure 22. PC1



I/O PORTS (Cont'd)

4.1.6 Port D

The Port D I/O pins are normally used for the input and output of video synchronization signals of the Sync Processor, but are set to I/O Input with pull-up upon reset. The I/O mode can be set individually for each port bit to Input with pull-up and output push-pull through the Port D DDR.

The configuration to support the Sync Processor requires that the SYNOP (bit7) and CLMPEN (bit6) of the ENR (Enable Register of SYNC) is reset. SYNOP enables port D bits 0,1 and CLMPEN enables Port D bit 6 to the sync outputs.

Port D, bit 4:3 are the alternate inputs ITA, ITB, (for the interrupt falling edge detector).
When a falling edge occurs on these inputs, an interrupt will be generated depending on the status of the INTX (ITAITE & ITBITE) bits in the MISCR Register.

Port D, bit 6 is switched to the alternate (CLAMPOUT) by resetting the CLMPEN bit of the ENR Register inside SYNC block.

If the SYNC function is selected, Port D bit 5 and 3 MUST be set as input to enable the HFBACK or VFBACK timing inputs.

Note: As these inputs are switched from normal I/O functionality, the video synchronization signals may also be monitored directly through the Port D Data Register for such tasks as checking for the presence of video signals or checking the polarity of Horizontal and Vertical synchronization signals (when the Sync Inputs are switched directly to the outputs using the multiplexers of the Sync Processor).

Table 11. Port D Description

PORT D	I / O		Alternate Function	
	Input*	Output	Signal	Condition
PD0	With pull-up	Push-pull	VSYNCO (push pull output)	SYNOP=0 (ENR [SYNC])
PD1	With pull-up	Push-pull	HSYNCO (push pull output)	SYNOP=0 (ENR [SYNC])
PD2	With pull-up	Push-pull	CSYNCI (input with TTL Schmitt trigger & pull-up)	-
PD3	With pull-up	Push-pull	ITA (input with CMOS Schmitt trigger & pull-up)	-
			VFBACK (input with TTL Schmitt trigger & pull-up)	-
PD4	With pull-up	Push-pull	ITB (input with CMOS Schmitt trigger & pull-up)	-
PD5	With pull-up	Push-pull	HFBACK (input with TTL Schmitt trigger & pull-up)	-
PD6	With pull-up	Push-pull	CLAMPOUT (push pull output)	CLMPEN=0 (ENR [SYNC])

* Reset state

I/O PORTS (Cont'd)

Figure 23. PD2 to PD5

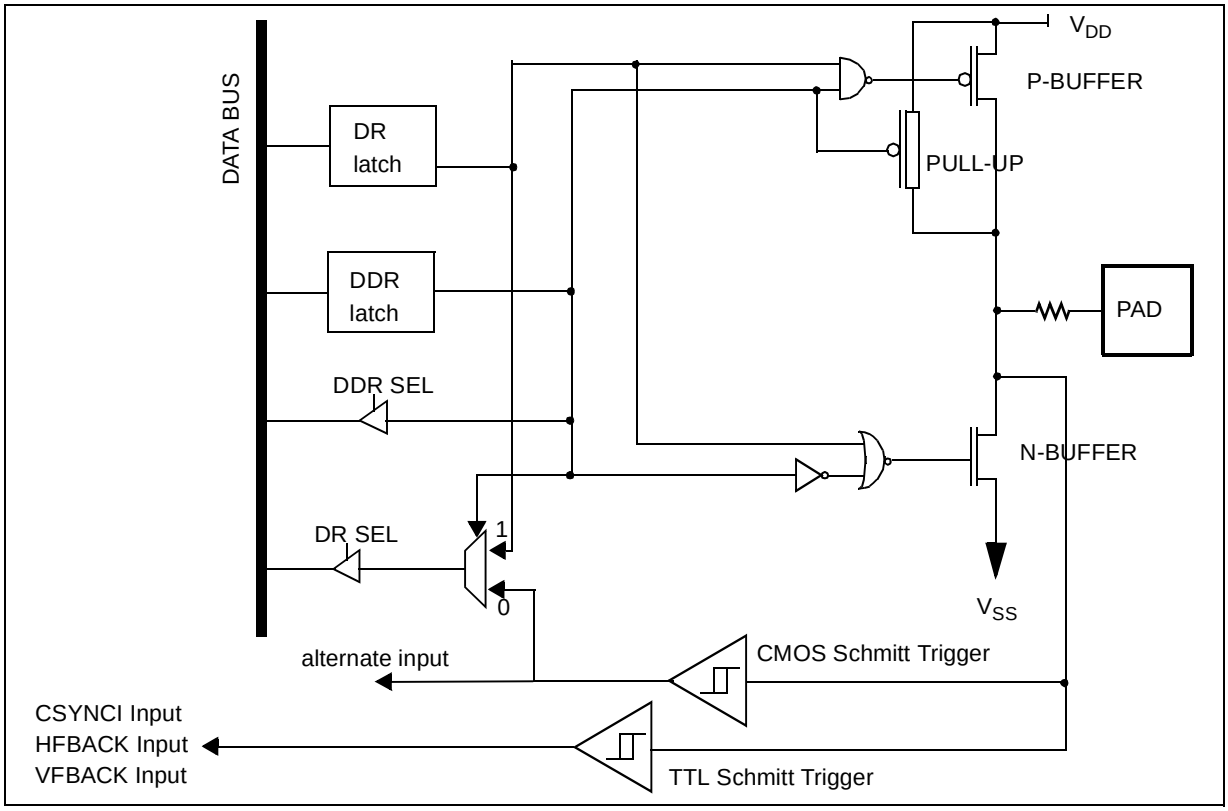
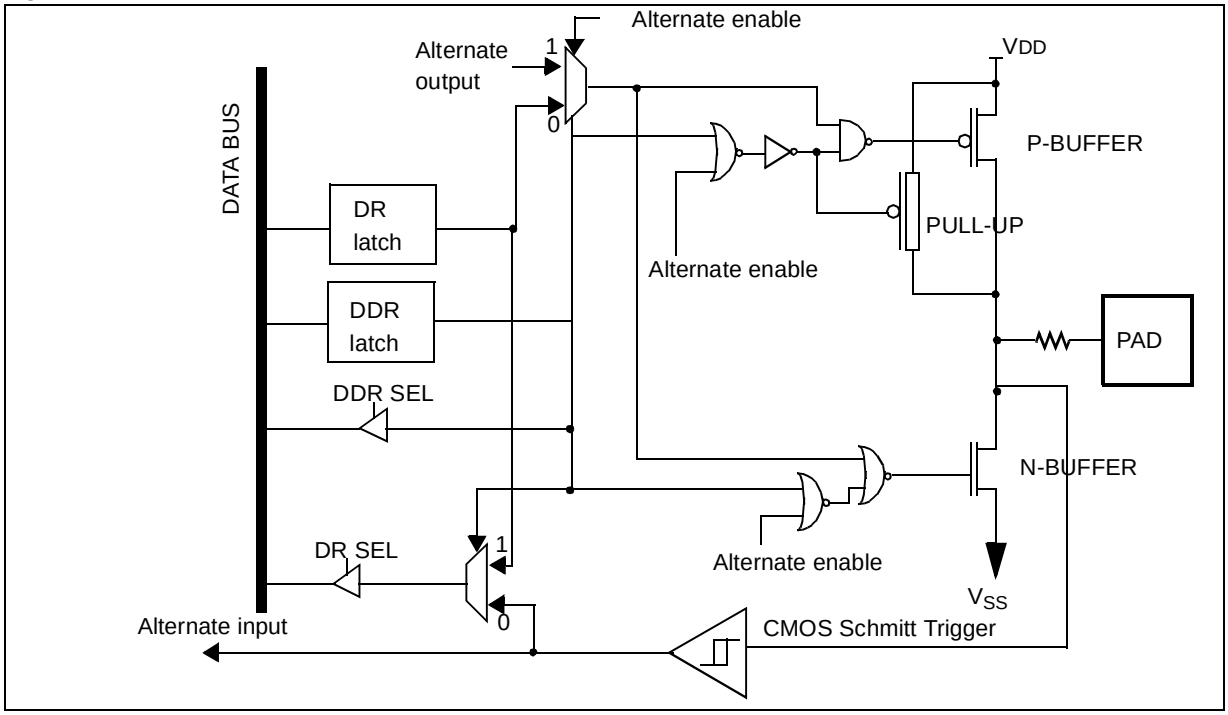
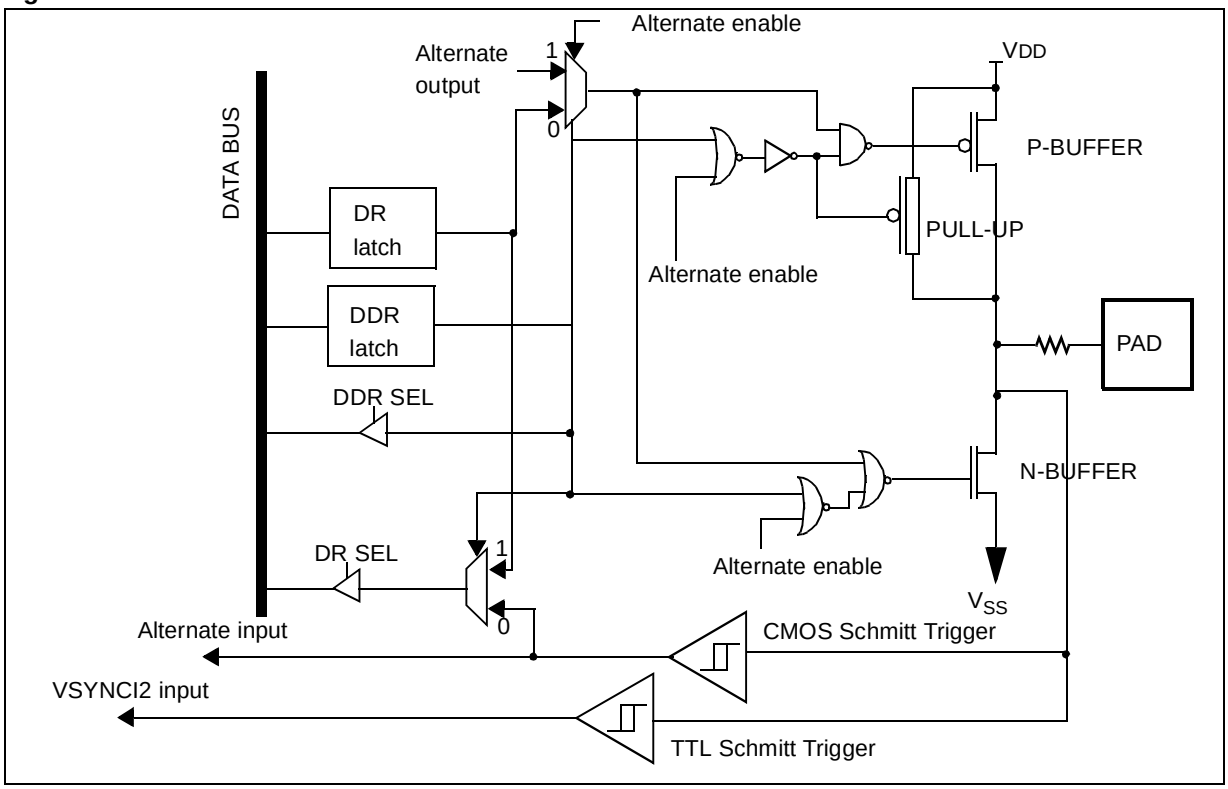


Figure 24. PD0 to PD1



I/O PORTS (Cont'd)

Figure 25. PD6



I/O PORTS (Cont'd)

4.1.7 Register Description

Data Registers (PxDR)

Read/Write
Reset Value: 0000 0000 (00h)

7							0
MSB							LSB

Data Direction Registers (PxDDR)

Read/Write
Reset Value: 0000 0000 (00h) (as inputs)

7							0
MSB							LSB

Table 12. I/O Ports Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
00	**PADR**	MSB							LSB
01	**PADDR**	MSB							LSB
02	**PBDR**	MSB							LSB
03	**PBDDR**	MSB							LSB
04	**PCDR**	MSB							LSB
05	**PCDDR**	MSB							LSB
06	**PDDR**	MSB							LSB
07	**PDDDR**	MSB							LSB

4.2 WATCHDOG TIMER (WDG)

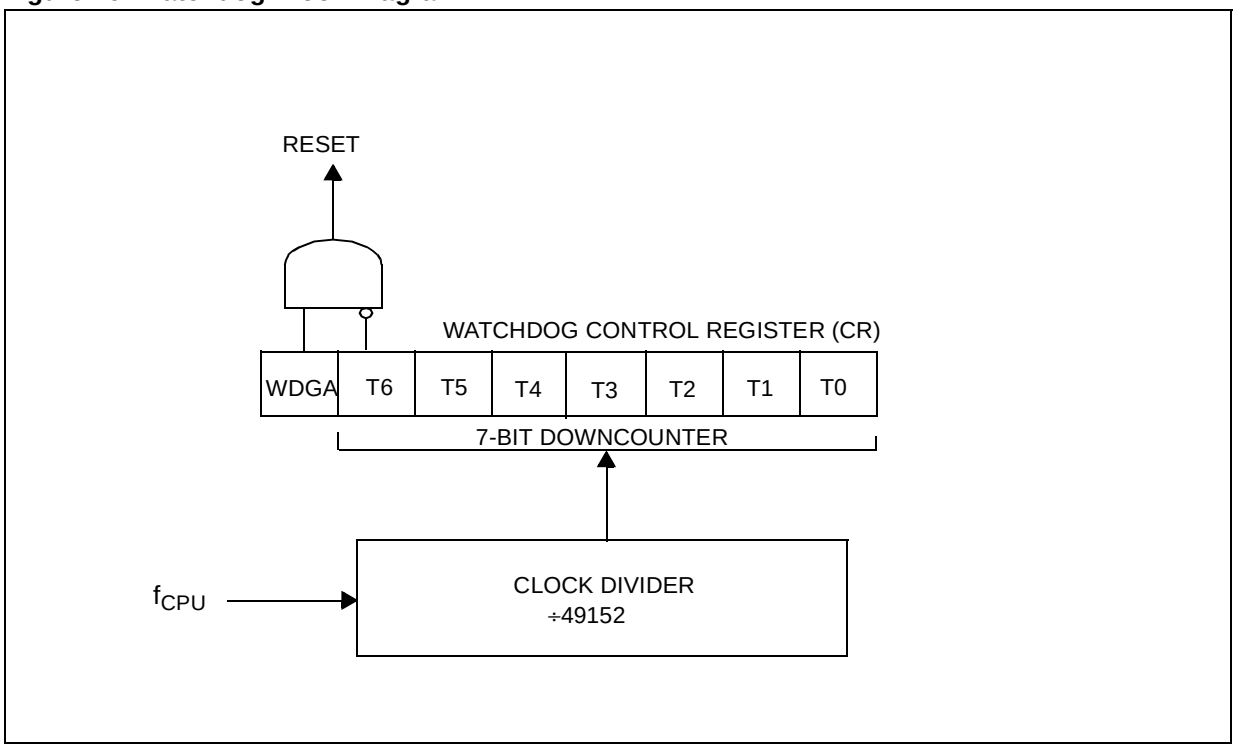
4.2.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

4.2.2 Main Features

- Programmable timer (64 increments of 49152 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero

Figure 26. Watchdog Block Diagram



4.2.3 Functional Description

The counter value stored in the CR register (bits T6:T0), is decremented every 49,152 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T6:T0) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the CR register must be between FFh and C0h (see Table 13 . Watchdog Timing (fCPU = 8 MHz)):

– The WDGA bit is set (watchdog enabled)
– The T6 bit is set to prevent generating an immediate reset
– The T5:T0 bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Table 13. Watchdog Timing (f_{CPU} = 8 MHz)

	CR Register initial value	WDG timeout period (ms)
Max	FFh	393.216
Min	C0h	6.144

Notes: Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

4.2.4 Interrupts

None.

4.2.5 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7 = **WDGA** *Activation bit.*
This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled
1: Watchdog enabled

Bit 6:0 = **T[6:0]** *7-bit timer (MSB to LSB).*
These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

Table 14. Watchdog Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
08	**WDGCR**	WDGA	T6	T5	T4	T3	T2	T1	T0
	Reset Value	0	1	1	1	1	1	1	1

4.3 16-BIT TIMER (TIM)

4.3.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

4.3.2 Main Features

- Programmable prescaler: f_{cpu} divided by 2, 4 or 8.
- Overflow status flag and maskable interrupt
- External clock input (must be at least 4 times slower than the CPU clock speed) with the choice of active edge
- Output compare functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Input capture functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One pulse mode
- 5 alternate functions on I/O ports*

The Block Diagram is shown in Figure 27.

Note: Some external pins are not available on all devices. Refer to the device pin out description.

4.3.3 Functional Description

4.3.3.1 Counter

The principal block of the Programmable Timer is a 16-bit free running counter and its associated 16-bit registers:

Counter Registers

- Counter High Register (CHR) is the most significant byte (MSB).
- Counter Low Register (CLR) is the least significant byte (LSB).

Alternate Counter Registers

- Alternate Counter High Register (ACHR) is the most significant byte (MSB).
- Alternate Counter Low Register (ACLR) is the least significant byte (LSB).

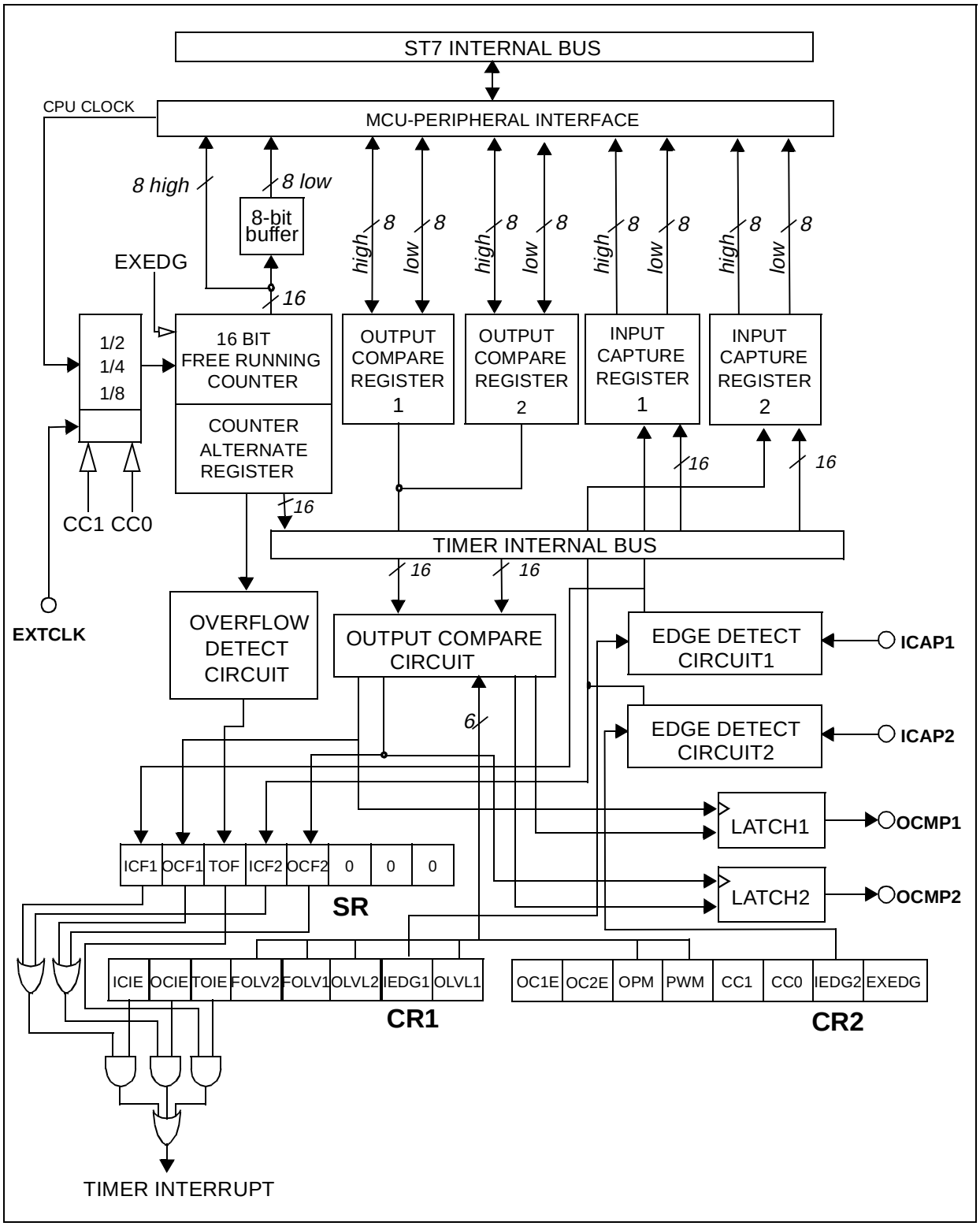
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (overflow flag), (see note page 43).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in Table 15 Clock Control Bits. The value in the counter register repeats every 131.072, 262.144 or 524.288 internal processor clock cycles depending on the CC1 and CC0 bits.

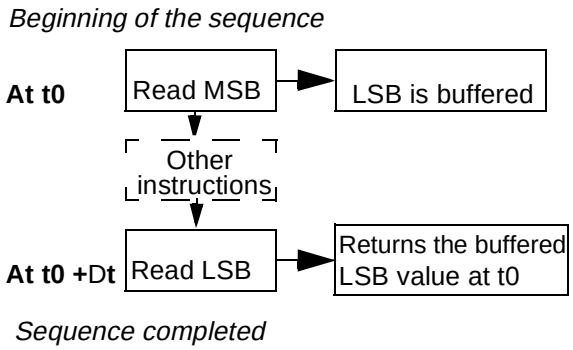
16-BIT TIMER (Cont'd)

Figure 27. Timer Block Diagram



16-BIT TIMER (Cont'd)

16-bit read sequence: (from either the Counter Register or the Alternate Counter Register).



The user must read the MSB first, then the LSB value is buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LSB of the count value at the time of the read.

An overflow occurs when the counter rolls over from FFFFh to 0000h then:

– The TOF bit of the SR register is set.

– A timer interrupt is generated if:

  – TOIE bit of the CR1register is set and

  – I bit of the CC register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done by:

1. Reading the SR register while the TOF bit is set.

2. An access (read or write) to the CLR register.

Notes: The TOF bit is not cleared by accesses to ACLR register. This feature allows simultaneous use of the overflow function and reads of the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by WAIT mode.

4.3.3.2 External Clock

The external clock (where available) is selected if CC0=1 and CC1=1 in CR2 register.

The status of the EXEDG bit determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronised with the falling edge of the internal CPU clock.

At least four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

16-BIT TIMER (Cont'd)

Figure 28. Counter Timing Diagram, internal clock divided by 2

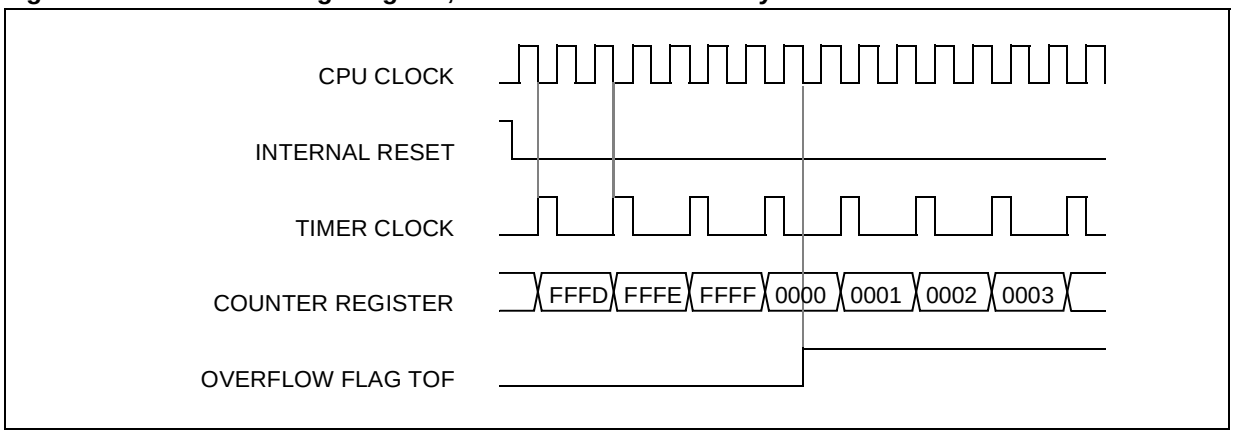


Figure 29. Counter Timing Diagram, internal clock divided by 4

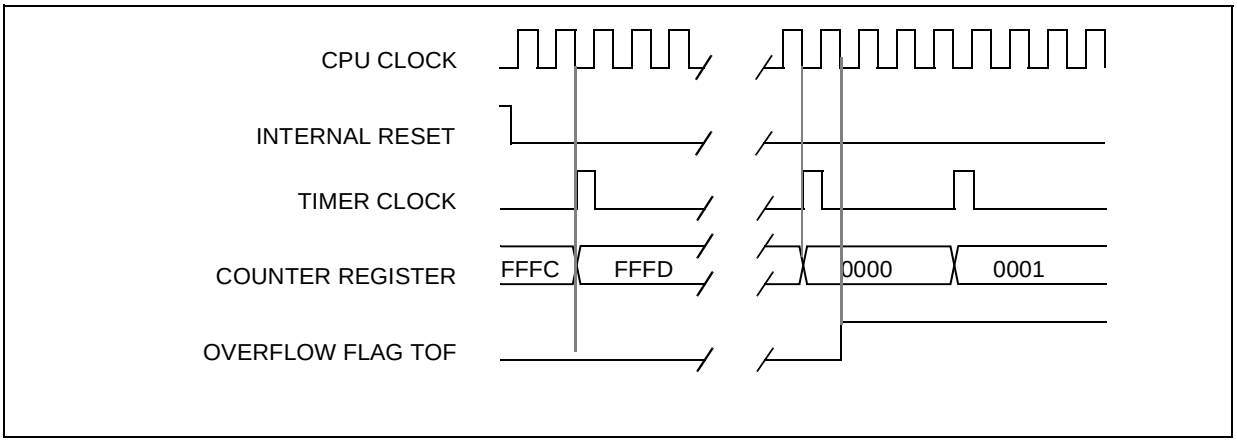
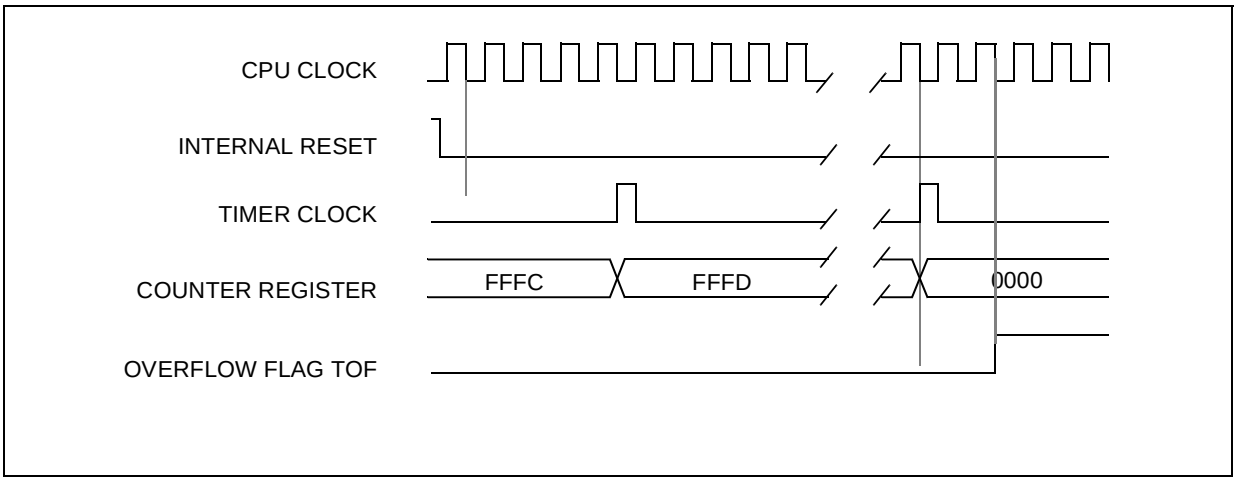


Figure 30. Counter Timing Diagram, internal clock divided by 8



16-BIT TIMER (Cont'd)

4.3.3.3 Input Capture

In this section, the index, *i*, may be 1 or 2.

The two input capture 16-bit registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition detected by the ICAP*i* pin (see figure 5).

	MS Byte	LS Byte
IC*i*R	**IC*i*HR**	**IC*i*LR**

IC*i* Rregister is a read-only register.

The active transition is software programmable through the IEDG*i* bit of the Control Register (CR*i*).

Timing resolution is one count of the free running counter: ($f_{CPU/(CC1.CC0)}$).

Procedure

To use the input capture function select the following in the CR2 register:

– Select the timer clock (CC1-CC0) (see Table 15 Clock Control Bits).

– Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit.

And select the following in the CR1 register:

– Set the ICIE bit to generate an interrupt after an input capture.

– Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit.

When an input capture occurs:

– ICF*i* bit is set.

– The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see Figure 32).

– A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request is done by:

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.

After reading the IC*i*HR register, transfer of input capture data is inhibited until the IC*i*LR register is also read.

The IC*i*R register always contains the free running counter value which corresponds to the most recent input capture.

16-BIT TIMER (Cont'd)

Figure 31. Input Capture Block Diagram

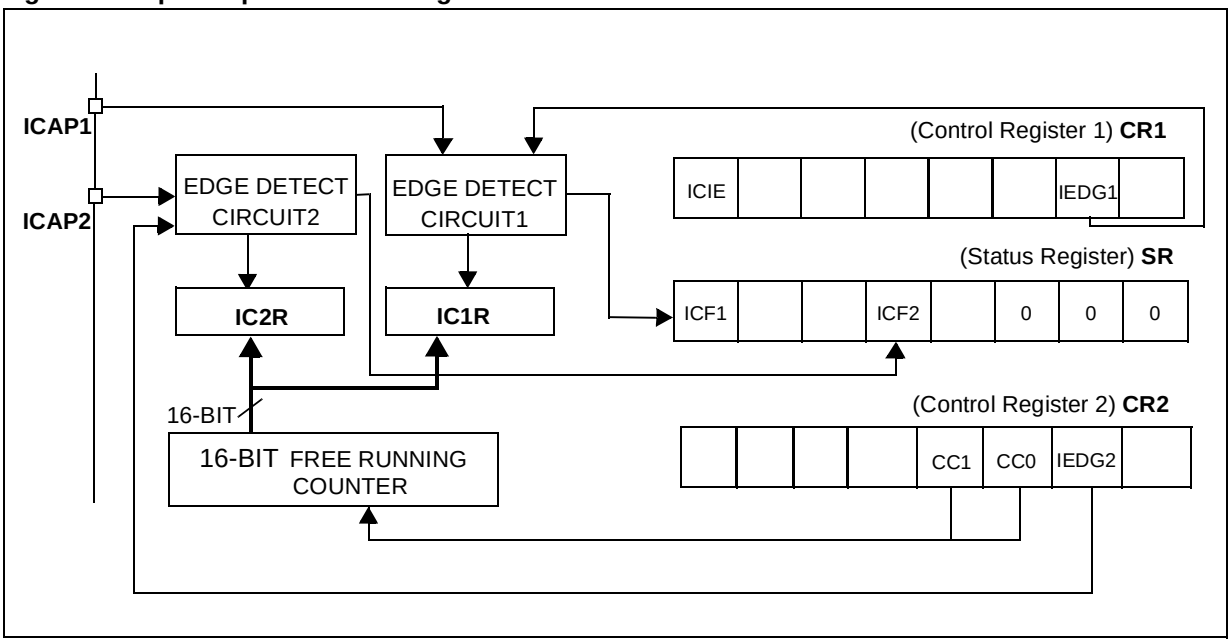
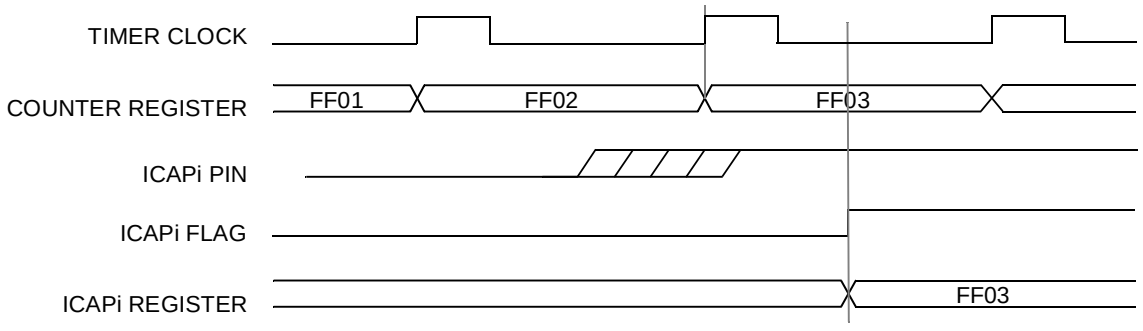


Figure 32. Input Capture Timing Diagram



Note: Active edge is rising edge.

16-BIT TIMER (Cont'd)

4.3.3.4 Output Compare

In this section, the index, *i*, may be 1 or 2.

This function can be used to control an output waveform or indicating when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OCIE bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare Register 1 (OC1R) and Output Compare Register 2 (OC2R) contain the value to be compared to the free running counter each timer clock cycle.

	MS Byte	LS Byte
OC*i*R	OC*i*HR	OC*i*LR

These registers are readable and writable and are not affected by the timer hardware. A reset event changes the OC*i*R value to 8000h.

Timing resolution is one count of the free running counter: ($f_{CPU/(CC1.CC0)}$).

Procedure

To use the output compare function, select the following in the CR2 register:

- Set the OC*i*E bit if an output is needed then the OCMP*i* pin is dedicated to the output compare *i* function.
- Select the timer clock (CC1-CC0) (see Table 15 Clock Control Bits).

And select the following in the CR1 register:

- Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When match is found:

- OCF*i* bit is set.
- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset and stays low until valid compares change it to a high level).
- A timer interrupt is generated if the OCIE bit is set in the CR2 register and the I bit is cleared in the CC register (CC).

Clearing the output compare interrupt request is done by:

3. Reading the SR register while the OCF*i* bit is set.
4. An access (read or write) to the OC*i*LR register.

Note: After a processor write cycle to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.

If the OC*i*E bit is not set, the OCMP*i* pin is a general I/O port and the OLVL*i* bit will not appear when match is found but an interrupt could be generated if the OCIE bit is set.

The value in the 16-bit OC*i*R register and the OLV*i* bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\Delta \text{ OC}i\text{R} = \frac{\Delta t * f_{CPU}}{(\text{CC1.CC0})}$$

Where:

Δt = Desired output compare period (in seconds)

f_{CPU} = Internal clock frequency

CC1-CC0 = Timer clock prescaler

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).

16-BIT TIMER (Cont'd)

Figure 33. Output Compare Block Diagram

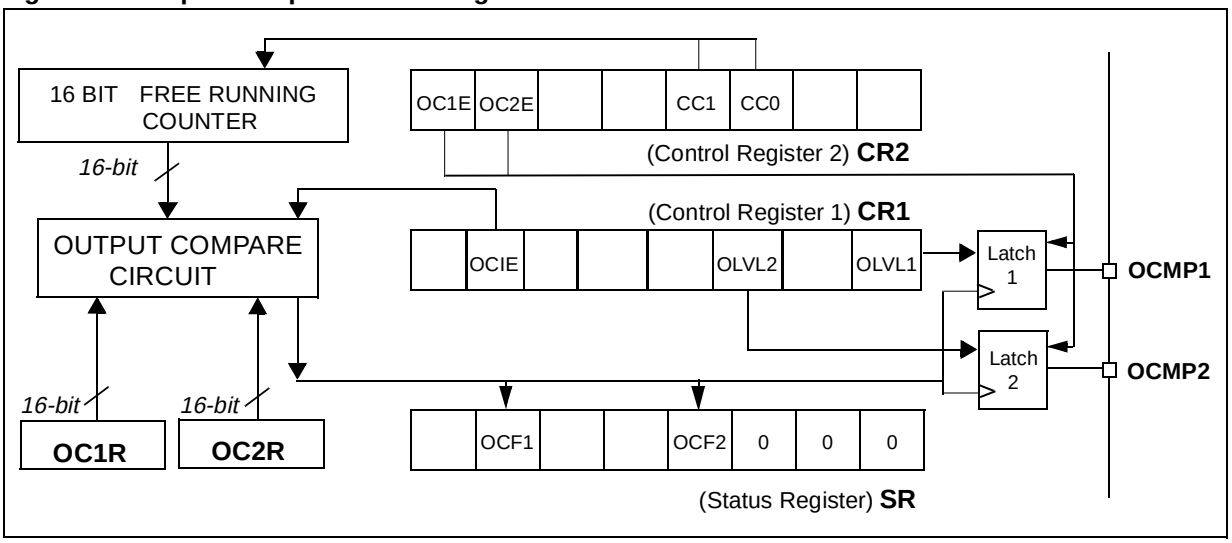
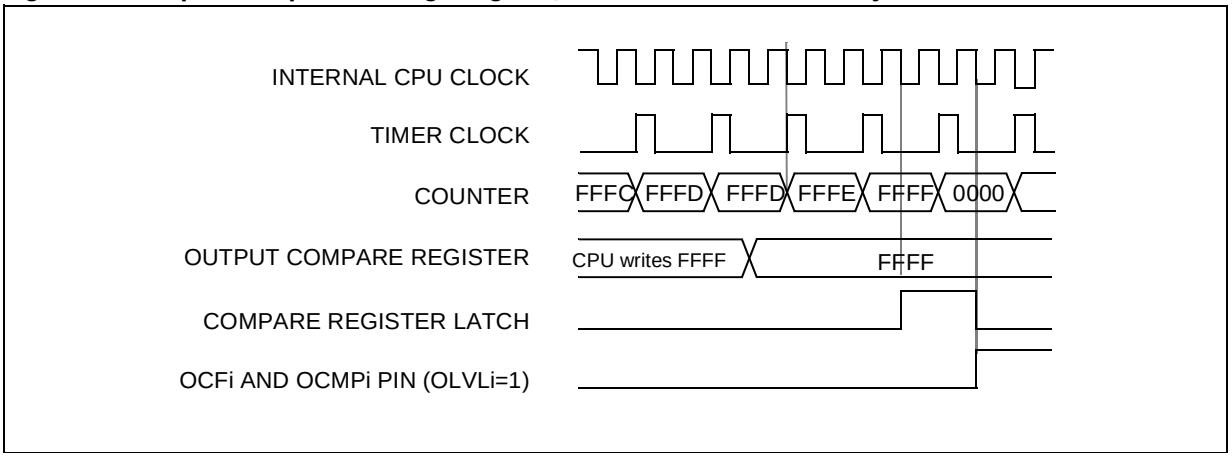


Figure 34. Output Compare Timing Diagram, Internal Clock Divided by 2



16-BIT TIMER (Cont'd)

4.3.3.5 Forced Compare Mode

In this section *i* may represent 1 or 2.

The following bits of the CR1 register are used:

			FOLV2	FOLV1	OLVL2		OLVL1

When the FOLV*i* bit is set, the OLVL*i* bit is copied to the OCMP*i* pin. The FOLV*i* bit is not cleared by software, only by a chip reset. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit=1).

The OCF*i* bit is not set, and thus no interrupt request is generated.

4.3.3.6 One Pulse Mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure

To use one pulse mode, select the following in the the CR1 register:

– Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.

– Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.

– Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit.

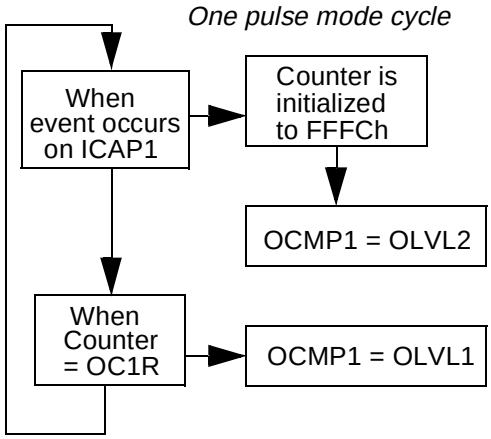
And select the following in the CR2 register:

– Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.

– Set the OPM bit.

– Select the timer clock CC1-CC0 (see Table 15 Clock Control Bits).

Load the OC1R register with the value corresponding to the length of the pulse (see the formula in Section 4.3.3.7).



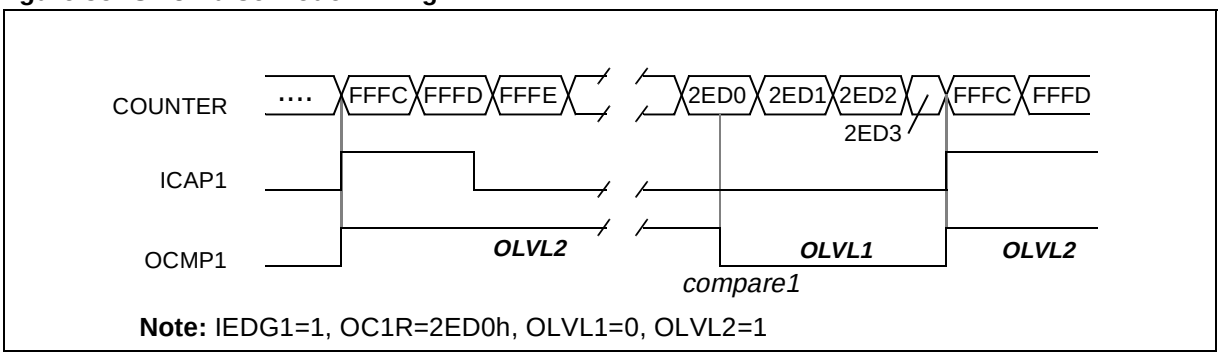
Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin. When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin, (See Figure 35).

Note: The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.

The ICF1 bit is set when an active edge occurs and can generate an interrupt if the ICIE bit is set.

When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

Figure 35. One Pulse Mode Timing



Note: IEDG1=1, OC1R=2ED0h, OLVL1=0, OLVL2=1

16-BIT TIMER (Cont'd)

4.3.3.7 Pulse Width Modulation Mode

Pulse Width Modulation mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

The pulse width modulation mode uses the complete Output Compare 1 function plus the OC2R register.

Procedure

To use pulse width modulation mode select the following in the CR1 register:

– Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC1R register.

– Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC2R register.

And select the following in the CR2 register:

– Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.

– Set the PWM bit.

– Select the timer clock (CC1-CC0) (see Table 15 Clock Control Bits).

Load the OC2R register with the value corresponding to the period of the signal.

Load the OC1R register with the value corresponding to the length of the pulse if (OLVL1=0 and OLVL2=1).

If OLVL1=1 and OLVL2=0 the length of the pulse is the difference between the OC2R and OC1R registers.

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\text{OC}i\text{R Value} = \frac{t * f_{CPU}}{(\text{CC1.CC0})} - 5$$

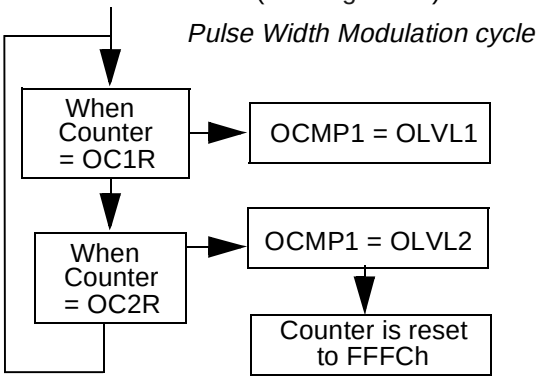
Where:

– t = Desired output compare period (seconds)

– f_{CPU} = Internal clock frequency (see Miscellaneous register)

– CC1-CC0 = Timer clock prescaler

The Output Compare 2 event causes the counter to be initialized to FFFCh (See Figure 36).

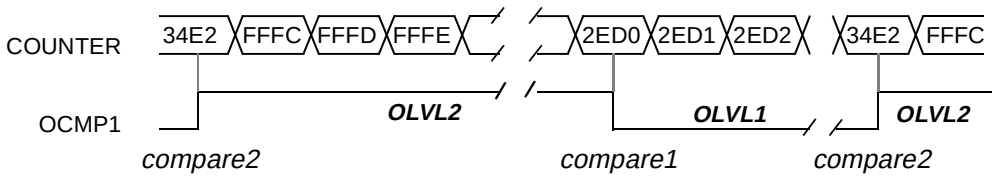


Note: After a write instruction to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.

The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited. The Input Capture interrupts are available.

When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

Figure 36. Pulse Width Modulation Mode Timing



Note: OC1R=2ED0h, OC2R=34E2, OLVL1=0, OLVL2= 1

4.3.4 Register Description

Each Timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 7 = **ICIE** *Input Capture Interrupt Enable.*

0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.

Bit 6 = **OCIE** *Output Compare Interrupt Enable.*

0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.

Bit 5 = **TOIE** *Timer Overflow Interrupt Enable.*

0: Interrupt is inhibited.
1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Bit 4 = **FOLV2** *Forced Output Compare 2.*
This bit is not cleared by software, only by a chip reset.

0: No effect.
1:Forces the OLVL2 bit to be copied to the OCMP2 pin.

Bit 3 = **FOLV1** *Forced Output Compare 1.*
This bit is not cleared by software, only by a chip reset.

0: No effect.
1: Forces OLVL1 to be copied to the OCMP1 pin.

Bit 2 = **OLVL2** *Output Level 2.*
This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse Mode and Pulse Width Modulation mode.

Bit 1 = **IEDG1** *Input Edge 1.*
This bit determines which type of level transition on the ICAP1 pin will trigger the capture.

0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **OLVL1** *Output Level 1.*
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

16-BIT TIMER (Cont'd)
CONTROL REGISTER 2 (CR2)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *Output Compare 1 Enable.*

0: Output Compare 1 function is enabled, but the OCMP1 pin is a general I/O.
1: Output Compare 1 function is enabled, the OCMP1 pin is dedicated to the Output Compare 1 capability of the timer.

Bit 6 = **OC2E** *Output Compare 2 Enable.*

0: Output Compare 2 function is enabled, but the OCMP2 pin is a general I/O.
1: Output Compare 2 function is enabled, the OCMP2 pin is dedicated to the Output Compare 2 capability of the timer.

Bit 5 = **OPM** *One Pulse Mode.*

0: One Pulse Mode is not active.
1: One Pulse Mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.

Bit 4 = **PWM** *Pulse Width Modulation.*

0: PWM mode is not active.
1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.

Bit 3, 2 = **CC1-CC0** *Clock Control.*

The value of the timer clock depends on these bits:

Table 15. Clock Control Bits

CC1	CC0	Timer Clock
0	0	f_{CPU} / 4
0	1	f_{CPU} / 2
1	0	f_{CPU} / 8
1	1	External Clock (where available)

Bit 1 = **IEDG2** *Input Edge 2.*
This bit determines which type of level transition on the ICAP2 pin will trigger the capture.

0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **EXEDG** *External Clock Edge.*
This bit determines which type of level transition on the external clock pin EXTCLK will trigger the free running counter.

0: A falling edge triggers the free running counter.
1: A rising edge triggers the free running counter.

16-BIT TIMER (Cont'd)

STATUS REGISTER (SR)
Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7							0
ICF1	OCF1	TOF	ICF2	OCF2			

Bit 7 = **ICF1** *Input Capture Flag 1.*

0: No input capture (reset value).
1: An input capture has occurred. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Bit 6 = **OCF1** *Output Compare Flag 1.*

0: No match (reset value).
1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.

Bit 5 = **TOF** *Timer Overflow.*

0: No timer overflow (reset value).
1:The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

Note: Reading or writing the ACLR register does not clear TOF.

Bit 4 = **ICF2** *Input Capture Flag 2.*

0: No input capture (reset value).
1: An input capture has occurred.To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 = **OCF2** *Output Compare Flag 2.*

0: No match (reset value).
1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2-0 = Unused.

INPUT CAPTURE 1 HIGH REGISTER (IC1HR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

INPUT CAPTURE 1 LOW REGISTER (IC1LR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

OUTPUT COMPARE 1 HIGH REGISTER (OC1HR)

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 1 LOW REGISTER (OC1LR)

Read/Write
Reset Value: 0000 0000 (00h)

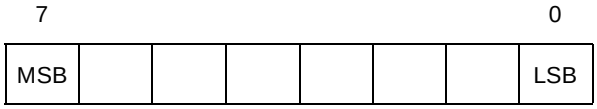
This is an 8-bit register that contains the low part of

16-BIT TIMER (Cont'd)

OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)

Read/Write
Reset Value: 1000 0000 (80h)

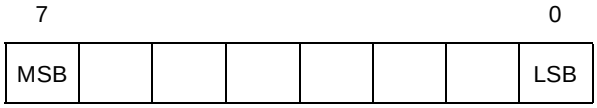
This is an 8-bit register that contains the high part of the value to be compared to the CHR register.



OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

Read/Write
Reset Value: 0000 0000 (00h)

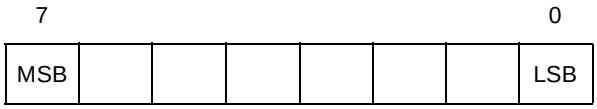
This is an 8-bit register that contains the low part of the value to be compared to the CLR register.



COUNTER HIGH REGISTER (CHR)

Read Only
Reset Value: 1111 1111 (FFh)

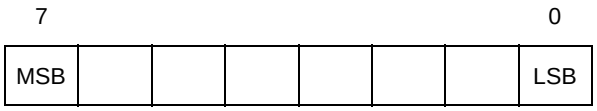
This is an 8-bit register that contains the high part of the counter value.



COUNTER LOW REGISTER (CLR)

Read/Write
Reset Value: 1111 1100 (FCh)

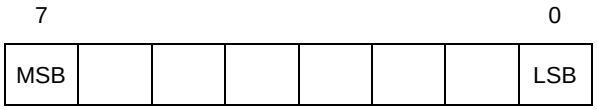
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the SR register clears the TOF bit.



ALTERNATE COUNTER HIGH REGISTER (ACHR)

Read Only
Reset Value: 1111 1111 (FFh)

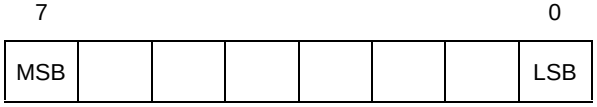
This is an 8-bit register that contains the high part of the counter value.



ALTERNATE COUNTER LOW REGISTER (ACLR)

Read/Write
Reset Value: 1111 1100 (FCh)

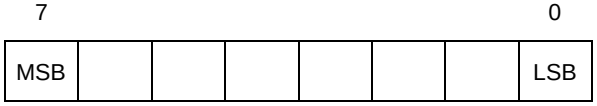
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to SR register does not clear the TOF bit in SR register.



INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).



INPUT CAPTURE 2 LOW REGISTER (IC2LR)

Read Only
Reset Value: Undefined

16-BIT TIMER (Cont'd)

Table 16. 16-Bit Timer Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
11	**CR2**	OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG
12	**CR1**	ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
13	**SR**	ICF1	OCF1	TOF	ICF2	OCF2	0	0	0
14	**IC1HR**	MSB							LSB
15	**IC1LR**	MSB							LSB
16	**OC1HR**	MSB							LSB
17	**OC1LR**	MSB							LSB
18	**CHR**	MSB							LSB
19	**CLR**	MSB							LSB
1A	**ACHR**	MSB							LSB
1B	**ACLR**	MSB							LSB
1C	**IC2HR**	MSB							LSB
1D	**IC2LR**	MSB							LSB
1E	**OC2HR**	MSB							LSB
1F	**OC2LR**	MSB							LSB

4.4 SYNC PROCESSOR (SYNC)

4.4.1 Introduction

The Sync processor handles all the management tasks of the video synchronization signals, and is used with the timer and software to provide information and status on the video standard and timings. This block supports multiple video standards such as: Separate Sync, Composite Sync and (via an external extractor) Sync on Green. The internal clock in the Sync processor is 4 MHz.

4.4.2 Main Features

■ Input Processing

– Presence of incoming signals (edge detection)
– Read the HSYNCI / VSYNCI input signal levels
– Measure the signal periods
– Detect the sync polarities
– Detect the composite sync and extract VSYNCO

■ Output Processing

– Control the sync output polarities
– Generate free-running frequencies
– Generate a video blanking signal
– Generate a clamping signal or a Moire signal

■ Analyzer Mode

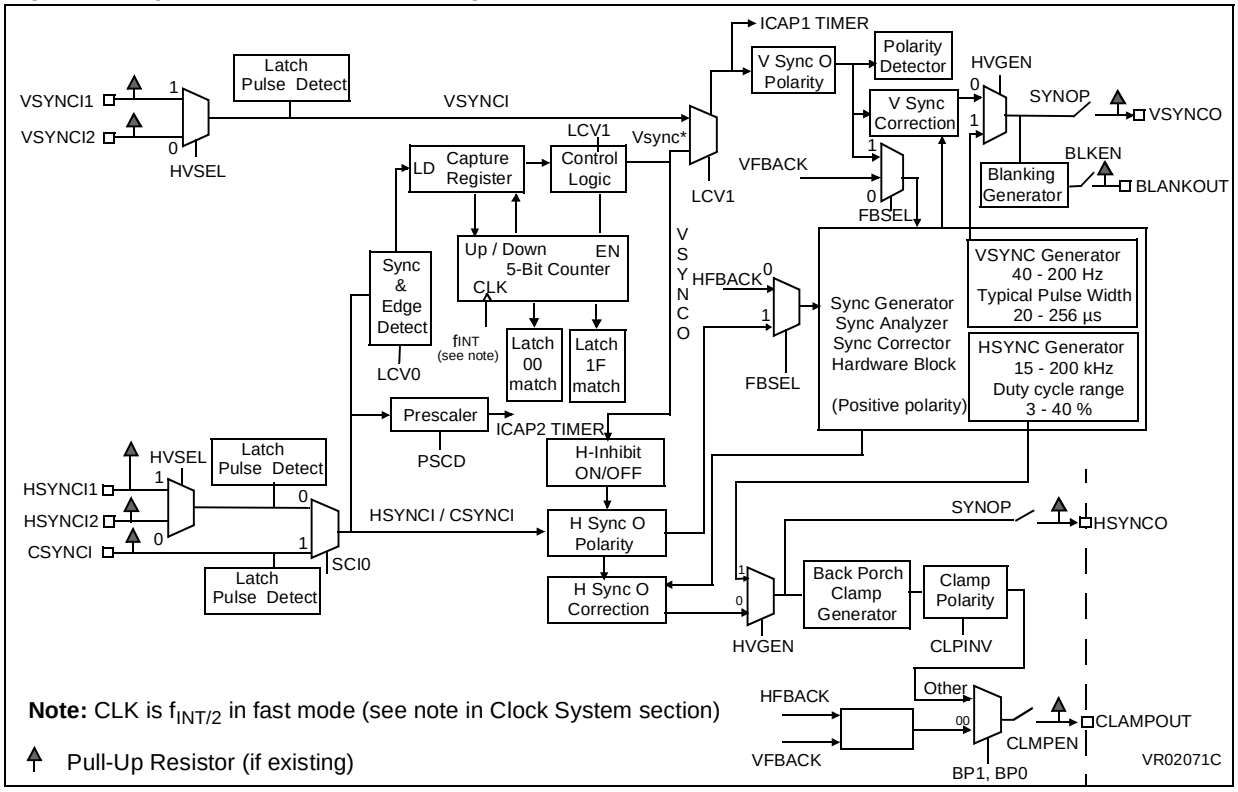
– Measure the number of scan lines per frame to simplify OSD vertical centering
– Detect HSYNCI reaching too high a frequency
– Detect pre/post equalization pulses
– Measure the low level of HSYNCO or HFBACK

■ Corrector Mode

– Inhibit Pre/Post equalization pulses
– Program VSYNCO pulse width extension
– Extend VSYNCO pulse widths during:
post-equalization pulse detection only
pre and post-equalization pulse detection

Note: Some external pins are not available on all devices. Refer to the device pinout description.

Figure 37. Sync Processor Block Diagram



Note: CLK is $f_{INT/2}$ in fast mode (see note in Clock System section)

Pull-Up Resistor (if existing)

SYNC PROCESSOR (SYNC) (Cont'd)

4.4.3 Input Signals

The Sync Processor has the following inputs (TTL level):

– VSYNCI1 Vertical Sync input1

– HSYNCI1 Horizontal Sync input1 or Composite sync

– VSYNCI2 Vertical Sync input2

– HSYNCI2 Horizontal Sync input2 or Composite sync

Note: The above input pairs can be used for DSUB or BNC connectors. To select these inputs use the HVSEL bit in the POLR register.

– CSYNCI Sync on Green (external extractor)

Note: If the CSYNCI pin is needed for another I/O function, the composite sync signal can be connected to HSYNCI using the SCI0 bit in the MCR register.

– HFBACK Horizontal Flyback input

– VFBACK Vertical Flyback input

4.4.4 Input Signal Waveforms

– The input signals must contain only synchronization pulses. In case of serration pulses on CSYNCI/HSYNCI, the pulse width should be less than 8µs.

– The VSYNCI signal is internally connected to Timer Input Capture 1 (ICAP1).

– The HSYNCI or CSYNCI signal, prescaled by 256, is internally connected to Timer Input Capture 2 (ICAP2).

– Typical timing range: See Figure 38 and 39

– If the timer clock is 2 MHz (external oscillator frequency 24 MHz):
PV accuracy = +/- 1 Timer clock (500ns)

PH*256 accuracy = +/- 1 Timer clock (500ns)

(PV= Vertical pulse, PH = Horizontal pulse)

4.4.5 Output Signals

The Sync Processor has the following outputs:

HSYNCO Horizontal Sync Output

Enable: SYNOP bit in ENR register

Programmable polarity:
HS0/HS1 bits in MCR register

In case of composite sync signal, the signal can be blanked by software during the vertical period (HINH bit in ENR register).

In case of separate sync, no blanking is generated.

VSYNCO Vertical Sync Output

Enable: SYNOP bit in ENR register

Programmable polarity:
VOP bit in the MCR register

In case of composite sync the delay of the extracted Vsync signal is:

minimum: 500ns + HSYNCO pulse width

maximum: 8750ns (max. threshold in extraction mode)

SYNC PROCESSOR (SYNC) (Cont'd)

Figure 38. Typical Horizontal Sync Input Timing

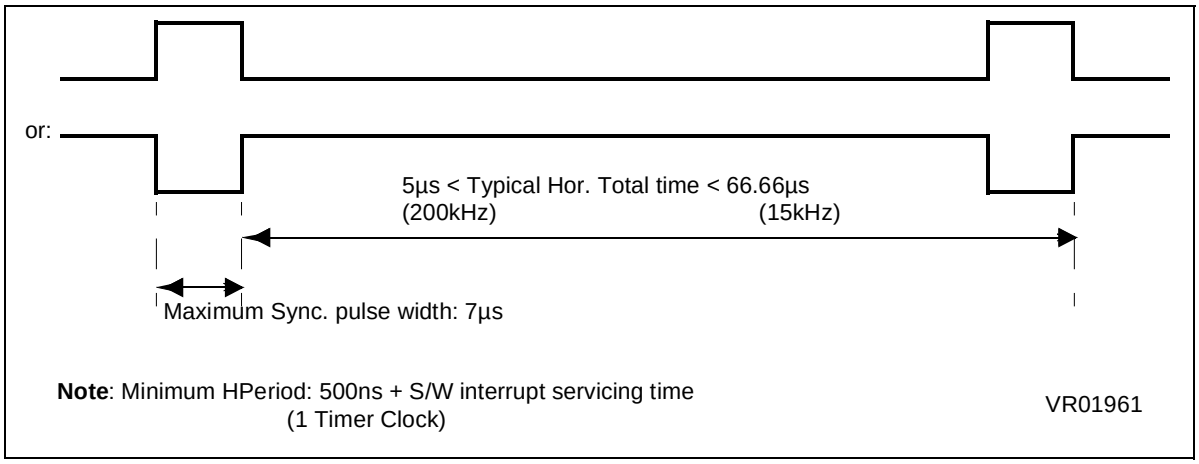
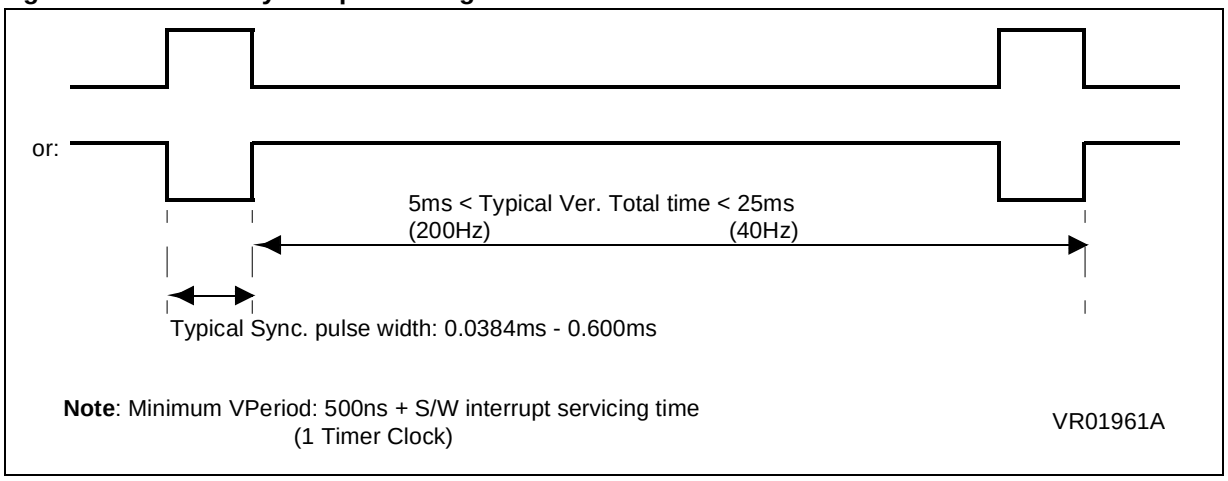


Figure 39. Vertical Sync Input Timing



SYNC PROCESSOR (SYNC) (Cont'd)
ClampOut and Moire Signal

Clamp Output signal

The clamping pulse generator can control the pulse width and polarity signal and can be configured as pseudo-front porch or back porch.

To use the ClampOut signal:

– Select the Clamping Pulse width:
BP0/BP1 bits in MCR register
– Program the Clamp polarity:
CLPINV bit in POLR register
– Select the ClampOut signal as back-porch (after falling edge of HSYNCO) or pseudo-front porch (after the rising edge of HSYNCO):
HS0/HS1 bits in MCR register.
– Enable the CLAMPOUT signal:
CLMPEN bit in ENR register

Moire Signal

The Moire output signal is available (instead of the clamping signal) to reduce the screen Moire effect and improve color transitions.

The CLAMPOUT pin is alternatively used to output a Moire signal.

The output signal toggles at each HFBACK rising edge. After each VFBACK falling edge, the value of the Moire output is the opposite of the previous one, independent of the number of HFBACK pulses during the VFBACK low level.

To use the Moire signal:

– Select the Moire signal:
Reset the BP0/BP1 bits in MCR register
– Enable the output signal:
CLMPEN bit in ENR register

Figure 40. Clamping Pulse (CLAMPOUT) Delay

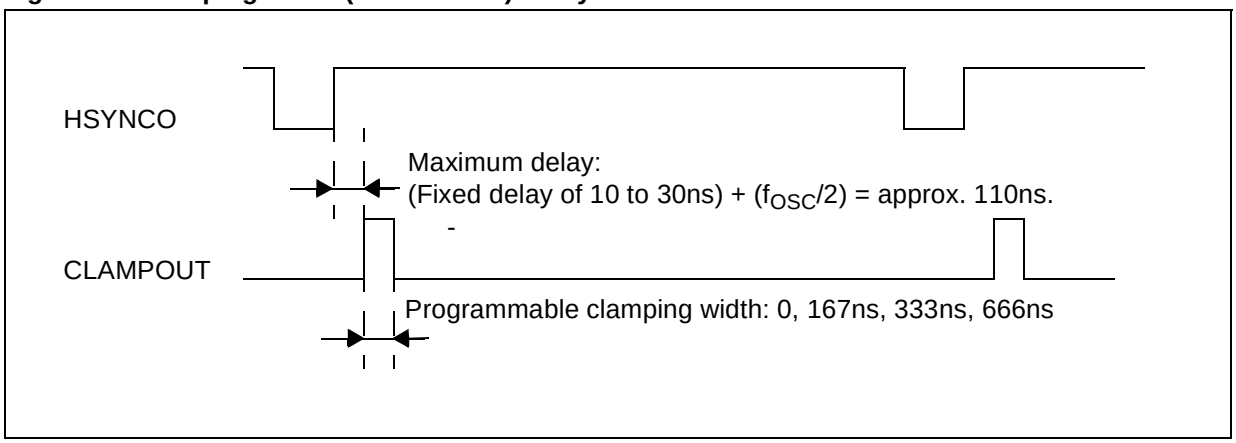
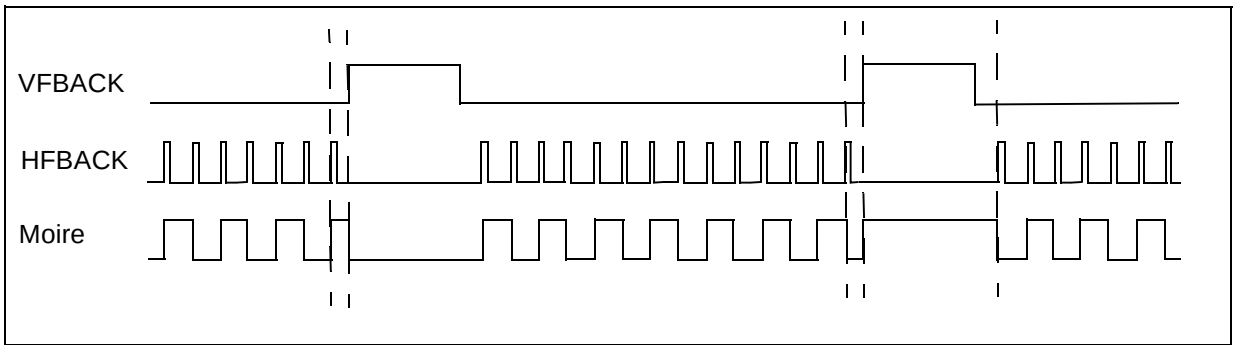


Figure 41. Moire Output (instead of Clamping Output)



4.4.5.1 Blanking output signal

The Video Blanking function uses VSYNCO, HFBACK, VFBACK as input signals and BLANKOUT output as Video Blanking Output. This output pin is a 5V open-drain output and can be AND-wired with any external video blanking signal.

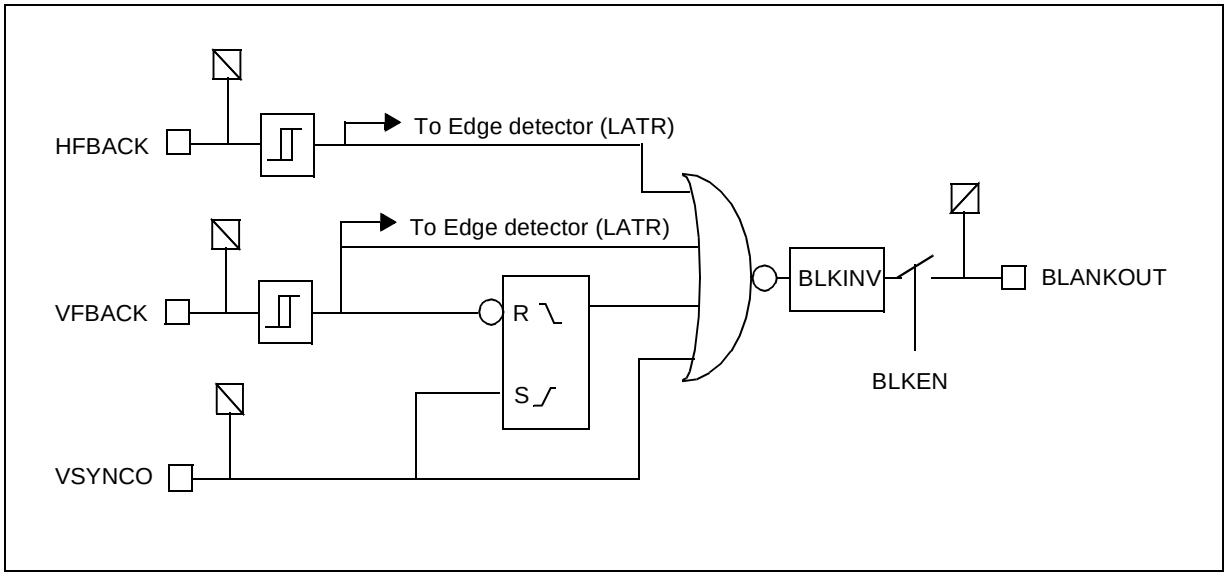
Note: HFBACK, VFBACK, VSYNCO signals must have positive polarity.

To use the video blanking signal:

– Program the polarity:
BLKINV bit in POLR register

– Enable the BLANKOUT output:
BLKEN bit in ENR register

Figure 42. Video Blanking Stage Simplified Schematic



SYNC PROCESSOR (SYNC) (Cont'd)

4.4.6 Input Processing

4.4.6.1 Detecting Signal Presence

The Sync Processor provides two ways of checking input signal presence, by directly polling the LATR Latch Register or using the Timer interrupts.

Polling check

Use the Latch Register (LATR), to detect the presence of HSYNCI, VSYNCI, CSYNCI, HFBACK and VFBACK signals. These latched bits are set when the falling edge of the corresponding signal is detected. They are cleared by software.

Interrupts check

Due to the fact that VSYNCI is connected to Timer Input Capture 1 and HSYNCI or CSYNCI is connected to Timer Input Capture 2, the Timer interrupts can be used to detect the presence of input signals. Refer to the 16-bit Timer chapter for the description of the Timer registers.

To use the interrupt method:

– Select Input Capture1 edge detection:
IEDG1 bit in the Timer CR1 register

– Select Input Capture 2 edge detection (must be falling edge):
IEDG2 bit = 0 in the Timer CR2 register

– Enable Timer Input Capture interrupts:
ICIE bit in the Timer CR1 register.

– Select the Hsync and Vsync input signals:
HVSEL bit in the POLR register

– Enable the prescaler for HSYNCI or CSYNCI signal:
PSCD bit in the CCR register.

– Select the normal mode:
LCV1/LCV0 bits in the CCR register.

Perform any of the following:

– Check for VSYNCI presence by monitoring interrupt requests from Timer ICAP1. When VSYNCI is detected then either detect the VSYNCI polarity or check for HSYNCI presence.

– Check for HSYNCI presence by monitoring interrupt requests from Timer ICAP2. On detecting HSYNCI, either detect its polarity or check if the composite sync on HSYNCI pin is detected or check for CSYNCI presence.

– Check for CSYNCI presence by monitoring interrupt requests from Timer ICAP2.

4.4.6.2 Measuring Sync Period

To measure the sync period, the Sync processor block uses the Timer Input Capture interrupts:

– ICAP1 connected to VSYNCI signal

– ICAP2 connected to HSYNCI/CSYNCI signal with a 256 prescaler

Calculating the difference between two subsequent Input Captures (16-bit value) gives the period for 256xPH (horizontal period) and PV (vertical period).

The period accuracy is one timer clock (500ns at 2 MHz), so that the tolerance is 500ns for PH and 256 * PH (PH accuracy =1.95ns).

Notes:

1) In case of composite sync, the HSYNCI period measurement can be synchronized on the VSYNCI pulse by setting and resetting the prescaler PSCD bit in the CCR register (for this function, the ICAP2 detection must be selected as falling edge).

   This avoids errors in the period measurement due to the Vsync pulse.

2) The Timer Interrupt request should be masked during a write access to any of the Sync processor control registers.

Important Note:

Since the recognition of the video mode relies on the accuracy of the measurements, it is highly recommended to implement a counter-style algorithm which performs several consecutive measurements before switching between modes.
The purpose of this algorithm is to filter out any glitches occurring on the video signals.

SYNC PROCESSOR (SYNC) (Cont'd)

4.4.6.3 Detecting Signal Polarity

The Sync Processor provides two ways for checking input signal polarity by polling the latches or using the 5-bit up/down counter.

Polling check

– HSYNCI polarity detection:
UPLAT/DWNLAT bits in LATR register
These bits are directly connected to the 5-bit Up/Down counter.

UPLAT=1/ DOWNLAT=0 HSYNCI polarity<0
UPLAT=0/ DOWNLAT=1 HSYNCI polarity>0

– VSYNCI Polarity Detection

– VPOL bit (VSYNCO polarity) in POLR and

– VOP bit (VSYNCO polarity control) in MCR
The delay between VSYNCI polarity changes and the VPOL bit typically toggles within 4 msecs. The polarity detector includes an integrator to filter possible incoming VSYNCI glitches.

5-bit Up/Down Counter Check for HSYNCI Polarity

This method involves the internal 5-bit up/down counter.
The counter value (CV4-CV0 bits) is updated with the 5-bit counter value at every detected edge on the signal monitored.
It is incremented when the signal is high, otherwise it is decremented.

– Start the detection phase:
Initialize the 5-bit counter: write '00000' in the CCR register (CV4-CV0 bits).

Select normal mode on falling edge:
LCV1/LCV0 = 0 in the CCR register.

– Software checks the counter value (CV4-CV0) after an interrupt (with the signal internally connected or ICAP2) or by polling (timeout 150µs).
Positive polarity: The counter value < 1Fh.

Negative polarity: The counter value =1Fh on the falling edge.

In case of a composite incoming signal, the software just has to check that the VSYNCO period and polarity are stable.

4.4.6.4 Extracting VSYNCO from CSYNCI

In case of composite sync, the Vertical sync output signal is extracted with the 5-bit up/down counter.

Initially, the width of an Horizontal Sync component pulse is automatically determined by hardware which defines a threshold for the 5-bit counter with a possible user-defined tolerance.

The circuit then monitors for any incoming period greater than this previously captured value. This is then processed as the VSYNCO signal.

To use the Vsync extractor, the following steps are necessary:

– Detection of a composite sync signal:
When the UPLAT and DOWNLAT bits in LATR register are set, a composite sync signal or a HSYNCI polarity change is detected.
If these bits are stable during two subsequent ICAP2 interrupt, the composite sync signal is stable.

– Defining a threshold:
Select the normal mode (LCV1/LCV0=0 in the CCR register).
Initialize the counter capture CV4-CV0 to 0.

This automatically measures the HSYNCI pulse width. It defines a threshold in the CV4-CV0 bits used by the 5-bit up/down counter.
It also allows to check the HSYNCI polarity (refer to the "5-bit Up/Down Counter Check" paragraph.
If a user-defined tolerance is to be added, then an updated value should be written in the CCR register (CV4-CV0 bits).
In a composite sync signal, Hsync and Vsync always have the same polarity.

– Starting the VSYNCO hardware extraction mode:
According to the Composite sync polarity, select the extraction mode (LCV1/LCV0 in CCR register) and rewrite the counter if necessary.

Negative polarity: minimum threshold (00h)

Positive polarity: maximum threshold (1Fh)

Note:

The extracted VSYNCO signal always has negative polarity.

SYNC PROCESSOR (SYNC) (Cont'd)

4.4.6.5 Example of VSYNCO extraction for a negative composite sync with serration pulses

Refer to Figure 43.

In extraction mode, the 5-bit comparator checks the counter value with respect to the threshold.

When the incoming signal is high, the counter is increased, otherwise it is decreased.

When the counter reaches the threshold on its way down, VSYNCO is asserted. During the vertical blanking, the counter value is decreased down to a programmable minimum, i.e. it does not underflow.

When the vertical period is finished, the counter starts counting up and when the maximum is reached, VSYNCO is negated. The extracted signal may be validated by software since it is input to Timer ICAP1.

Serration pulses during vertical blanking can be filtered if the serration pulse widths are less than 8µs.

In the same way, positive composite sync signals can be used by properly selecting the edge sensitivity in HSYNCI width measurement mode (LCV0 bit).

Figure 43. VSYNCO Extraction from a Composite Signal (negative polarity)

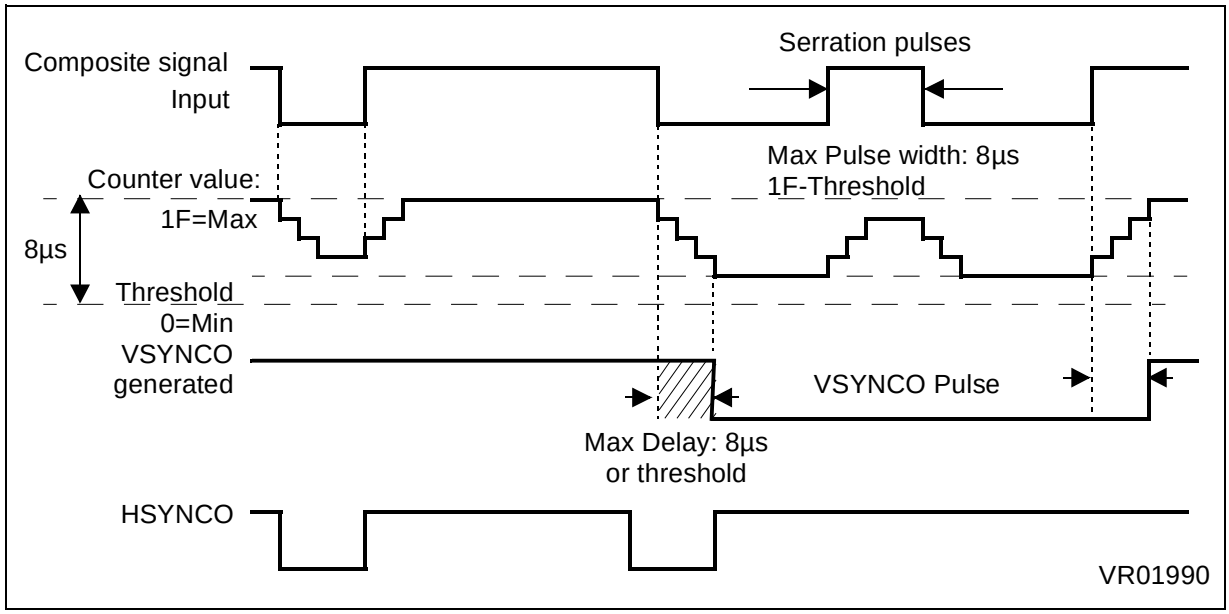
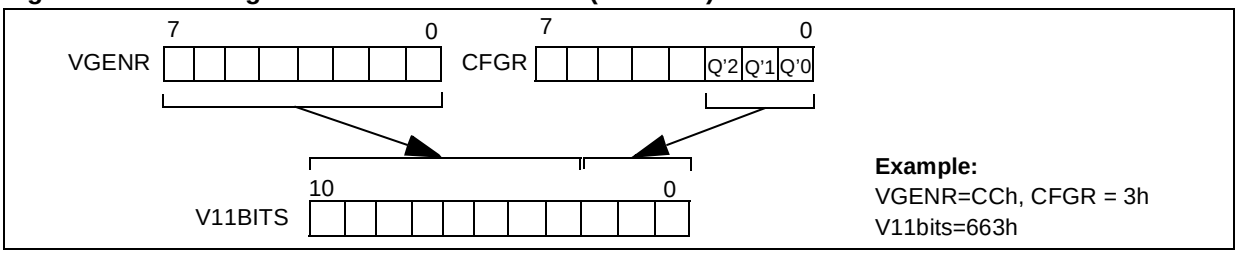


Figure 44. Obtaining the 11-bit Vertical Period (V11BITS)



SYNC PROCESSOR (SYNC) (Cont'd)

4.4.7 Output Processing

4.4.7.1 Generating Free-Running Frequencies

The free-running frequencies function is used to:

– Drive the monitor when no or bad sync signals are received.

– Stabilize the OSD screen when the monitor is unlocked.

– Perform fast alignment for maintenance purposes.

Note: When free-running mode is active, the analyzer and corrector modes must be disabled.

– VCORDIS = 1, VEXT = 0 in CFGR and POLR registers for vertical output measurement

– 2FHINH = 0 in CFGR register for horizontal low level measurment

– VACQ, HACQ = 0, in CFGR register for analyzer mode

The Sync processor can generate any of the following output sync signals HSYNCO, VSYNCO, CLAMPOUT, BLANKOUT.

To select the generation mode:

– Program the horizontal period using the HGENR register.

– Program the vertical period using the VGENR (8 bits) and CFGR (3 bits) registers (2047 scan lines per frame). Refer to Figure 44.

– Configure the following bits:
SYNOP = 0
HVGEN = 1
HACQ = 0
VACQ = 0

Horizontal Period

PH = Horizontal period = ((HGENR+1)/4) µs

Pulse width: 2 µs => HGENR min=8

Polarity: Positive

HGENR range: [8..255]

Vertical Period

PV = Vertical period = (PH * V11bits) µs

V11bits is a concatenation of VGENR and the Q'2 Q'1 Q'0 bits of the CFGR register.
Refer to Figure 44.

Pulse width: 4 * PH => min value= 8µs

Polarity: Positive

VGENR/CFGR range: [5..7FF]

Table 17. Typical values for generated HSYNC signals

HGENR (hex value)	H Period	HFREQ	Pulse Width	Duty Cycle
13	5 µs	200 kHz	2 µs	40%
1F	8 µs	125 kHz	2 µs	25%
3F	16 µs	62.5 kHz	2 µs	12.5%
7F	32 µs	31.25 kHz	2 µs	6.2%
FF	64 µs	15.6 kHz	2 µs	3.1%

Table 18. Typical values for generated VSYNC signals

HGENR (hex value)	H Period	H Freq	V11bits (hex value)	V Period	V Freq	Pulse width
13	5 µs	200 kHz	7FF (2047)	10.2 ms	97.7 Hz	20 µs
13	5 µs	200 kHz	400 (1024)	5.1 ms	195 Hz	20 µs
1F	8 µs	125 kHz	7FF (2047)	16.3 ms	61 Hz	32 µs
1F	8 µs	125 kHz	400 (1024)	8.2 ms	122 Hz	32 µs
3F	16 µs	62.5 kHz	7FF (2047)	32.6 ms	30.6 Hz	64 µs
3F	16 µs	62.5 kHz	400 (1024)	16.4 ms	60.9 Hz	64 µs
7F	32 µs	31.25 kHz	7FF (2047)	65.5 ms	15 Hz	128 µs
7F	32 µs	31.25 kHz	400 (1024)	32.8 ms	30 Hz	128 µs

SYNC PROCESSOR (SYNC) (Cont'd)

4.4.8 Analyzer Mode

The analyzer block is used for all extra measurements on the sync signals to manage the monitor functions:

– Measure the number of scan lines per frame (VSYNCO or VFBACK) to simplify the OSD vertical centering.

– Measure the low level of HSYNCO or HFBACK. This function can be used for VSYNCO pulse extension or for a fast estimation of the incoming Hsync signal period.

– Detection of the pre/post equalization pulses.

Notes:

1. Analyzer mode should be performed before corrector mode.
2. When analyzer mode is active, the free-running frequencies generator and corrector mode must be disabled.
   – HVGEN = 0 in ENR register
   – 2FHINH 0 in CFGR register for Horizontal low level measurement
   – VEXT = 0, VCORDIS = 1 in CFGR, POLR registers for Vertical output measurement
3. If H/VBACK are selected (FBSEL=0) corrector mode must be disabled
4. For all measurements, HSYNCO and VSYNCO must be **POSITIVE**.

4.4.8.1 Horizontal Low Level Measurement

The measurement starts in setting HACQ by software. When this bit is cleared by hardware, the HGENR register returns the result.

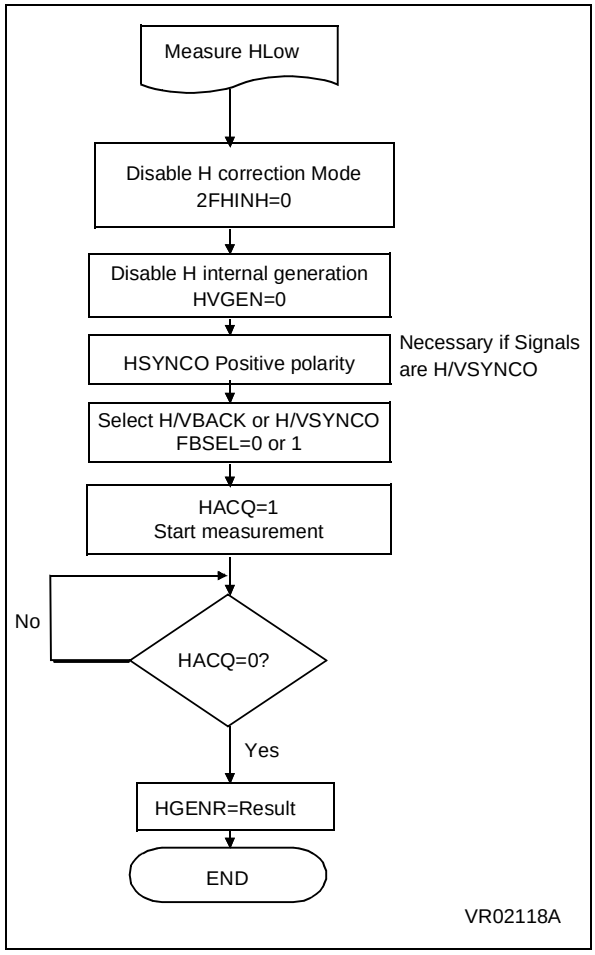
The algorithm is shown in Figure 45.

HLow = ((255-HGENR+1)/4) µs

Note: HLow maximum value = 64µs (even if real value is greater)

For maximum accuracy, it is possible to measure the low level of HFBACK with the same technique (FBSEL bit in the MCR register).

Figure 45. Horizontal Low Level Measurement



SYNC PROCESSOR (SYNC) (Cont'd)

4.4.8.2 Vertical Output Measurement

The function of vertical pulse measurement is to:

– Capture the number of HSYNCO pulses during a Low level of VSYNCO.

– Capture the number of HFBACK pulses during a Low level of VFBACK (maximum accuracy).

Start the measurement by setting VACQ in the CFGR. When the measurement is completed this bit is cleared by hardware. The VGENR and CFGR registers return the result.

The algorithm is shown in Figure 46.

HLine = 2048 - (V11bits)

Hline maximum value = 2048 (even if real value is greater)

V11bits = VGENR(8 MSB) and Q'2,Q'1,Q'0 (3 LSB). Refer to Figure 44.

Note: In case of pre/post equalization pulses, set the 2FHINH bit in the CFGR register.

4.4.8.3 Detection of pre equalization pulses

This function uses two bits:

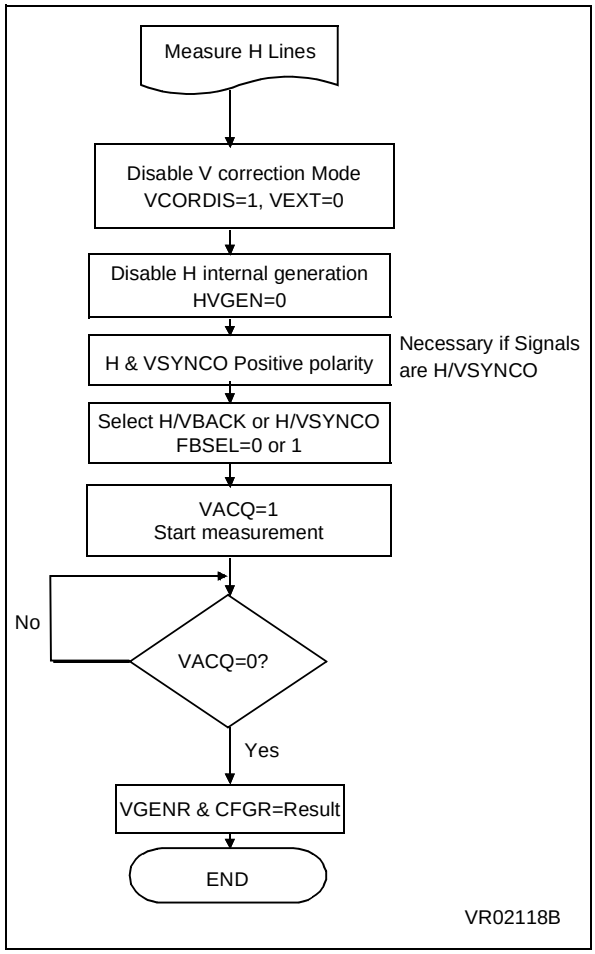
– 2FHDET in POLR register continuously updated by hardware

– 2FHLAT in LATR register set by hardware when a higher frequency is detected and reset by software

A measurement of the low level of HSYNCO is necessary before reading this information.

Note: Reset the 2FHLAT bit in the LATR register on the third Hsync pulse after the Vsync pulse.

Figure 46. Vertical Output Measurement



SYNC PROCESSOR (SYNC) (Cont'd)

4.4.9 Corrector Mode

In this mode, you can perform the following functions:

– Inhibit pre/post equalization pulses

This removes all pre/post equalization pulses on the HSYNCO signal.

The inhibition starts on the falling edge of HSYNCO and lasts for (((HGENR+1)/4)-2) µs. The decrease of 2µs (one minimum pulse width) avoids the removal of the next pulse of HSYNCO.

Procedure:

1. HSYNCO and VSYNCO polarities must be positive.
2. Measure the low level of HSYNCO.
3. Set the 2FHINH bit in the CFGR register.

– Extend VSYNCO pulse width by several scan lines

This function can be also used to extend the video blanking signal.

Procedure:

1. HSYNCO and VSYNCO polarities must be positive.
2. Set the 2FHINH bit in the CFGR register only if some pre/post equalizations pulses are detected. (2FHLAT, 2FHDET flags).
3. The extension will be the number of HSYNCO periods set in the VGENR register.
4. Reset the VCORDIS bit in the POLR register.

– Extend VSYNCO width during all post equalization pulses.

This function extends the VSYNCO pulse width when post equalization pulses are detected (2FHDET bit in the POLR register and 2FHLAT bit in the LATR register).

Procedure:

1. HSYNCO and VSYNCO polarities must be positive.
2. Set the 2FHINH bit in the CFGR register to remove pre/post equalization pulses.
3. Measure the low level of HSYNCO.
4. Update HGENR =(FFh - (HGENR + 1)) + 4 to add tolerance
5. Write VGENR > 0.
6. Reset the VCORDIS bit in the POLR register
7. Set the VEXT bit in the CFGR register.

– Extend VSYNCO pulse width during pre and post equalization pulses (for test only).

This function allows extending the VSYNCO pulse width as long as equalization pulses are detected. (VSYNCO = VSYNCO + 2FHDET).

Procedure:

1. HSYNCO and VSYNCO polarities must be positive.
2. Set the 2FHINH bit in the CFGR register to remove pre/post equalization pulses.
3. Measure the low level of HSYNCO.
4. Update HGENR =(FFh - (HGENR + 1)) + 4.
5. Write VGENR > 0.
6. Reset the VCORDIS bit in the POLR register.
7. Set the VEXT bit in the CFGR register.
8. Set the 2FHEN bit in the ENR register.

Notes:

1. When corrector mode is active, the free-running frequencies generator and analyzer mode must be disabled.
(HVGEN=0 in ENR register, HACQ=0, VACQ=0 in the CFGR register).
2. If VGENR=0, all VSYNCO correction functions are disabled except the 2FHEN bit which must be cleared if VGENR = 0 or VCORDIS = 1.

SYNC PROCESSOR (SYNC) (Cont'd)

4.4.10 Register Description
CONFIGURATION REGISTER (CFGR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
HACQ	VACQ	-	2FHINH	VEXT	Q'2	Q'1	Q'0

Bit 7 = **HACQ** Horizontal Sync Analyzer Mode
Set by software, reset by hardware when the measurement is done. The sync generator must be disabled (HVGEN=0).

0 : Measurement is done, the result can be read in HGENR.
1: Start measuring HSYNCO/HFBACK low level.

Bit 6 = **VACQ** Vertical Sync Analyzer Mode
Set by software, reset by hardware when the measurement is done. The sync generator must be disabled (HVGEN=0).

0: Measurement is done, and the result can be read in VGENR.
1: Start measuring the number of scan lines during VSYNCO/VFBACK low level.

Bit 5 = Reserved. Must be cleared.

Bit 4 = **2FHINH** Inhibition of Pre/Post equalization pulses.
This function removes pre/post equalization pulses on HSYNCO signal. The sync generator and the Horizontal sync analyzer must both be disabled (HVGEN=HACQ=0).

0: Disable
1: Enable

Bit 3 = **VEXT** VSYNCO pulse width extension in case of post-equalization pulses.
The sync generator and the Horizontal and Vertical sync analyzer must be disabled (HVGEN = 0, HACQ = 0, VACQ = 0, VCORDIS=0). Vertical extension must be enabled (VGENR > 0).

0: Disable
1: Enable

Bits 2:0 = **Q'2..Q'0**
These are the read/write LSB of the VGENR 11-bit counter. Refer to Figure 44.

SYNC PROCESSOR (SYNC) (Cont'd)
MUX CONTROL REGISTER (MCR)

Read/Write

Reset Value: 0010 0000 (20h)

7							0
BP1	BP0	FBSEL	SCI0	HS1	HS0	VOP	-

Bit 7:6 = **BP1, BP0** *Back Porch Pulse control*

BP1	BP0	Back Porch pulse width
0	0	No Back Porch, Moire output selected
0	1	167ns Back Porch ± 10 ns
1	0	333ns Back Porch ± 10 ns
1	1	666ns Back Porch ± 10 ns

Bit 5 = **FBSEL** *VSYNCO/HSYNCO or VFBACK/ HFBACK analysis*

0: HFBACK & VFBACK
1: HSYNCO & VSYNCO

Bit 4 = **SCI0** *HSYNCI/CSYNCI selection*

0: HSYNCI
1: CSYNCI

Bit 3:2 = **HS1, HS0** *Horizontal Signal selection*
These bits allow inversion of the HSYNCI/CSYNCI

HS1	HS0	HSYNCI Selection Mode
0	0	CLAMPOUT after HSYNCO rising edge HSYNC0 <- (HSYNCI, CSYNCI)
0	1	CLAMPOUT after HSYNCO rising edge HSYNC0 <- ($\overline{\text{HSYNCI}}$, $\overline{\text{CSYNCI}}$)
1	0	CLAMPOUT after HSYNCO falling edge HSYNC0 <- ($\overline{\text{HSYNCI}}$, $\overline{\text{CSYNCI}}$)
1	1	CLAMPOUT after HSYNCO falling edge HSYNC0 <- (HSYNCI,CSYNCI)

Note: In case of composite sync, if HSYNCO blanking is enabled (HINH=0 in the ENR register), HS1 must = 1 (CLAMPOUT after HSYNCO rising edge not allowed).

Bit 1 = **VOP** *Vertical Polarity control*
The VOP bit inverts the VSYNCO Sync signal.

0: No polarity inversion (VSYNC0 <- VSYNCI)
1: Inversion enabled (VSYNC0 <- $\overline{\text{VSYNCI}}$)

Note: If at each vertical input capture the VPOL bit is copied by software on the VOP bit, the VSYNCO signal will have a constant positive polarity.

Note: The internally extracted VSYNCO has ALWAYS negative polarity.

Bit 0 = Reserved. Must always be cleared.

SYNC PROCESSOR (SYNC) (Cont'd)
COUNTER CONTROL REGISTER (CCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
PSCD	LCV1	LCV0	CV4	CV3	CV2	CV1	CV0

Bit 7 = **PSCD** *Prescaler Enable bit.*

0: Enable the Prescaler by 256
1: Disable the Prescaler and reset it to 7Fh. This also disables the ICAP2 event.

Bit 6:5 = **LCV1, LCV0** *VSYNCO Extraction Control*

LCV1	LCV0	VSYNC0 Control Bits
0	0	Normal mode Counter capture on input falling edge
0	1	Normal mode Counter capture on input rising edge
1	0	Extraction mode CSYNCI/HSYNCI Negative polarity CV4-0 = counter minimum threshold
1	1	Extraction mode CSYNCI/HSYNCI Positive polarity CV4-0 = counter maximum threshold

Bit 4:0 = **CV4-CV0** *Counter Captured Value.*
These bits contain the counter captured value in different modes.

In VSYNCO extraction mode, they contain the HSYNCI pulse-width measurement.

POLARITY REGISTER (POLR)

Bits 5-4 Read Only, other bits Read/Write

Reset Value: 0000 1000 (08h)

7							0
SOG	0	VPOL	2FHDET	HVSEL	VCORDIS	CLPINV	BLKINV

Bit 7 = **SOG** *Sync On Green Detector*
SOG is set by hardware if CSYNCI pulse is not included in the window between HSYNCI rising edge and HSYNCI falling edge + dt .
Cleared by software.

Table 19. Sync On Green Window

WINDOW DELAY	min.	max.
dt	165 ns	250 ns

Bit 6 = Reserved, forced by hardware to 0.

Bit 5 = **VPOL** *Vertical Sync polarity* (read only)

0: Positive polarity
1: Negative polarity

Note: If the Vertical Sync polarity is changing, the VPOL bit will be updated after a typical delay of 4 msec.

Bit 4 = **2FHDET** *Detection of Pre/Post Equalization pulses* (read only).
This bit is continuously updated by hardware. It is valid when the sync generator and horizontal analyzer are disabled (HVGEN = 0, HACQ = 0).

0: None detected
1: Pre/Post Equalization pulses detected

Bit 3 = **HVSEL** *Alternate Sync Input Select.*
This bit selects between the two sets of Horizontal and Vertical Sync inputs

0: HSYNCI2 / VSYNCI2
1: HSYNCI1 / VSYNCI1

Bit 2 = **VCORDIS** *Extension Disable Signal*
(Extension with VGENR Register)

0: enable
1: disable

Bit 1 = **CLPINV** *Programmable ClampOut pulse polarity.*

0: Positive
1: Negative

Bit 0 = **BLKINV** *Programmable blanking polarity*

0: Negative
1: Positive

SYNC PROCESSOR (SYNC) (Cont'd)
LATCH REGISTER (LATR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
CSYN	HSYN	VSYN	HFLY	VFLY	UPLAT	DWNLAT	2FHLAT

Bit 7 = **CSYN** *Detection of pulses on CSYNCI*
Set on falling edge of CSYNCI
Cleared by software (by writing zero).

Bit 6 = **HSYN** *Detection of pulses on HSYNCI*
Set on falling edge of HSYNCI1 or HSYNCI2
Cleared by software (by writing zero).

Bit 5 = **VSYN** *Detection of pulses on VSYNCI*
Set on falling edge of VSYNCI1 or VSYNCI2
Cleared by software (by writing zero).

Bit 4 = **HFLY** *Detection of pulses on HFBACK*
Set on falling edge of HFBACK input
Cleared by software (by writing zero).

Bit 3 = **VFLY** *Detection of pulses on VFBACK*
Set on falling edge of VFBACK input
Cleared by software (by writing zero).

Bit 2 = **UPLAT** *Detection of the maximum value of 5-bit up/down counter.*
Set when the 5 bit up/down counter reaches its maximum value (1Fh or Threshold)
Cleared by software (by writing zero).

Bit 1 = **DWNLAT** *Detection of minimum value of 5-bit up/down counter.*
Set when the 5 bit up/down counter reaches its minimum value (00 or Threshold)
Cleared by software (by writing zero).

Note: DWNLAT and UPLAT may be used for HSYNCI polarity detection and Composite Sync detection as follows:

UPLAT	DWNLAT	HSYNCI Characteristics
0	0	No Info
0	1	Positive Polarity
1	0	Negative Polarity
1	1	Composite Sync

Bit 0 = **2FHLAT** *equalization pulses latch.*
This bit may be used to detect pre/ postequalization pulses or a too high horizontal frequency.
Set by hardware when Pre/Post equalization pulses are detected.
Must be reset by software.
It is valid when the sync generator and Horizontal analyzer are disabled (HVGEN = 0, HACQ = 0)

SYNC PROCESSOR (SYNC) (Cont'd)

HORIZONTAL SYNC GENERATOR REGISTER (HGENR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
MSB							LSB

Case HVGEN = 1: Generation mode
In this mode, this register contains the Hsync free-running frequency.
The generated signal is:

- Pulse width: 2 µs.
- Period PH = ((HGENR+1)/4) µs.
- Polarity: Positive

Note: The value in HGENR must be in the range [8..255].

Case HVGEN = 0: Analyzer/corrector Mode

Sub-case HACQ = 1: Analyzer Mode
By setting HACQ bit by software the Analyzer mode starts. When HACQ is cleared by hardware, HGENR returns the duration of HSYNCO/ HFBACK low level. The analysis should be done before corrector mode.

Sub-case HACQ = 0: Corrector Mode
In this mode, the final HSYNCO signal on the pin can be corrected in order to detect and inhibit pre/ post equalization pulses.

VERTICAL SYNC GENERATOR REGISTER (VGENR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
MSB							LSB

Case HVGEN = 1: Generation mode
In this mode, this register contains the Vsync free-running frequency (11-bit value).
The generated signal is:

- Pulse width: 4 * PH µs (horizontal period).
- Period PV = PH * (V11bits) µs.
- Polarity: Positive

Note: The value in VGENR must be in the range [5..255] The Vsync generation mode works as an 11-bit horizontal line counter (2047 scan lines per frame max.). The 3 LSB are in the CFGR register. Refer to Figure 44.

Case HVGEN = 0: Analyzer/Corrector Mode

Sub-case VACQ = 1: Analyzer Mode
Set the VACQ bit to start analyzer mode. When VACQ is cleared by hardware, VGENR/CFGR returns the number of scan lines during the VSYNCO/VFBACK low level period.

Sub-case VACQ = 0: Corrector Mode
VSYNCO pulse width is extended by VGENR scan lines. If VGENR = 0, all VSYNCO corrections are disabled.

SYNC PROCESSOR (SYNC) (Cont'd)
ENABLE REGISTER (ENR)

Read/Write

Reset Value: 1100 0011 (C3h)

7							0
SYNOP	CLMPEN	BLKEN	HVGEN	2FHEN	HINH	HSIN1	VSIN1

Bit 7 = **SYNOP** *HSYNCO, VSYNCO outputs enable*
0: Enabled
1: Disabled

Bit 6 = **CLMPEN** *Clamping or Moire output enable*
0: Clamping or Moire output (function of BP0, BP1) enabled
1: Clamping or Moire output disabled

Bit 5 = **BLKEN** *Blanking Output*
0: Disabled
1: Enabled

Bit 4 = **HVGEN** *Sync Generation function*

0: Analyzer/Corrector Mode
1: Generation of HSYNCO and VSYNCO free-running frequencies

Bit 3 = **2FHEN** *VSYNCO Extension*
VSYNCO is forced high when detecting pre- and post-equalization pulses. It is valid when the sync generator and analyzer are disabled (HVGEN = 0, HACQ = 0, VACQ = 0). Refer to the procedure in Section 4.4.9 Corrector Mode.
0: Disabled
1: Enabled

Bit 2 = **HINH** *HSYNCO Blanking*
HSYNCO is blanked during the extracted VSYNCO pulse.
0: Enabled
1: Disabled

Bit 1 = **HSIN1** (read only)
Returns the HSYNCI1 pin level

Bit 0 = **VSIN1** (read only)
Returns the VSYNCI1 pin level

Table 20. Summary of the Main Sync Processor Modes

Sync Processor Mode	SYNOP	HVSEL	HVGEN	HACQ	VACQ
DSUB Selected as Inputs (HSYNCI1/VSYNCI1)	---	1	---	---	---
BNC Selected as Inputs (HSYNCI2/VSYNCI2)	---	0	---	---	---
Don't drive the monitor with any Sync signals	1	---	---	---	---
Generate Sync Signals to drive the Monitor hardware	0	---	1	0	0
Use the Sync Processor to drive the monitor hardware by incoming Sync signals	0	---	0	---	---
Analyse the number of Scan Lines during one vertical frame	---	--	0	---	1
Analyse the HSYNC delay between two pulses	---	---	0	1	---

SYNC PROCESSOR (SYNC) (Cont'd)

Table 21. SYNC Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
40	**CFGR**	HACQ	VACQ	-	2FHINH	VEXT	Q'2	Q'1	Q'0
	Reset Value	0	0	0	0	0	0	0	0
41	**MCR**	BP1	BP0	FBSEL	SCI0	HS1	HS0	VOP	-
	Reset Value	0	0	1	0	0	0	0	0
42	**CCR**	PSCD	LCV1	LCV0	CV4	CV3	CV2	CV1	CV0
	Reset Value	0	0	0	0	0	0	0	0
43	**POLR**	SOG	0	VPOL	2FHDET	HVSEL	VCORDIS	CLPINV	BLKINV
	Reset Value	0	0	0	0	1	0	0	0
44	**LATR**	CSYN	HSYN	VSYN	HFLY	VFLY	UPLAT	DWNLAT	2FHLAT
	Reset Value	0	0	0	0	0	0	0	0
45	**HGENR**	MSB							LSB
	Reset Value	0	0	0	0	0	0	0	0
46	**VGENR**	MSB							LSB
	Reset Value	0	0	0	0	0	0	0	0
47	**ENR**	SYNOP	CLMPEN	BLKEN	HVGEN	2FHEN	HINH	HSIN1	VSIN1
	Reset Value	1	1	0	0	0	0	1	1

4.5 TIMING MEASUREMENT UNIT (TMU)

4.5.1 Introduction

The timing measurement unit (TMU) allows the analysis of the current video timing characteristics in order to control display position and size.

It consists of measuring the timing between the horizontal or vertical sync output signals and the active video signal input (AV).

4.5.2 Main Features

- Horizontal or vertical timing measurement
- Oscillator clock f_{OSC} (24 or 12 MHz) used for horizontal measurement
- Horizontal sync signal (HSYNCO or HFBACK) and Vertical sync signal (VSYNCO or VFBACK) used for all measurements
- Measurements performed on positive signals only
- 11-bit counter
- Overflow detection

4.5.3 Functional Description

The Timing Measurement Unit is centered around an 11-bit counter. Depending on the H_V bit of the control register, the TMU measures the horizontal or vertical video characteristics.

For horizontal analysis (refer to Figure 48):

– Obtain the minimum number of oscillator clock cycles (H1) between the falling edge of the horizontal sync signal (HSYNCO or HFBACK) and the first rising edge of the active video input (AV), for all lines, between 2 consecutive vertical sync pulses.

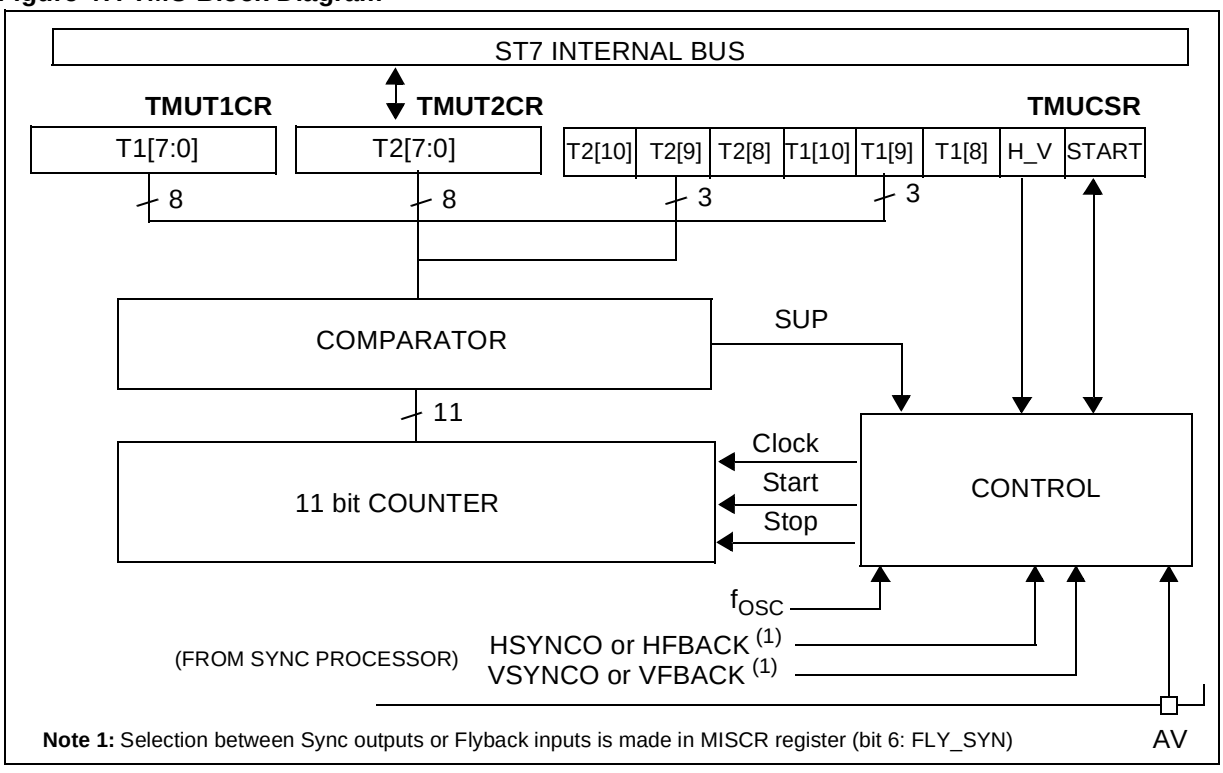
– Obtain the minimum number of oscillator clock cycles (H2) between the last falling edge of the active video input (AV) and the rising edge of the horizontal sync signal (HSYNCO or HFBACK) for all lines, between 2 consecutive vertical sync pulses.

Note: Horizontal measurement is inhibited during the high level of VSYNCO or VFLYBACK.

For vertical analysis (refer to Figure 49):

– Obtain the minimum number of horizontal sync pulses (V1) between the falling edge of the vertical sync signal (VSYNCO or VFBACK) and the first rising edge of the active video input, during 2 consecutive frames.

Figure 47. TMU Block Diagram



TIMING MEASUREMENT UNIT (Cont'd)

– Obtain the minimum number of horizontal sync output pulses (V2) between the last falling edge of the active video input (AV) and the rising edge of the vertical sync signal (VSYNCO or VFBACK) during two 2 consecutive frames.

The H_V bit selects horizontal or vertical measurement. This selection should be made prior to starting the measurement by setting the START bit. This bit is set by software but only cleared by hardware at the end of the measurement.

When the measurement is finished (rising edge of AV, horizontal or vertical sync signals), the results (T1,T2) are transferred into the corresponding registers (H1,H2) or (V1,V2).

Note: The values of the H1/H2 or V1/V2 registers are available only at the end of a measurement (after the START bit has been cleared).

4.5.3.1 Horizontal Measurement

When the H_V bit = 1, and when the START bit is set by software, the measurement starts after the next vertical sync pulse. The TMU searches the minimum values of H1 and H2 until the rising edge of the next following vertical sync pulse. The START bit is then cleared by hardware.

The values of the H1 and H2 registers are available only at the end of a measurement, in other words when the START bit is at 0.

4.5.3.2 Vertical Measurement

When the H_V bit = 0 and, when the START bit is set by software, the TMU measures the minimum V1 and V2 values during 2 consecutive vertical frames. The START bit is then cleared by hardware.

4.5.3.3 Special cases

– If an overflow of the counter occurs during any of the measurements, the measured T1 or T2 values will be 7FFh.

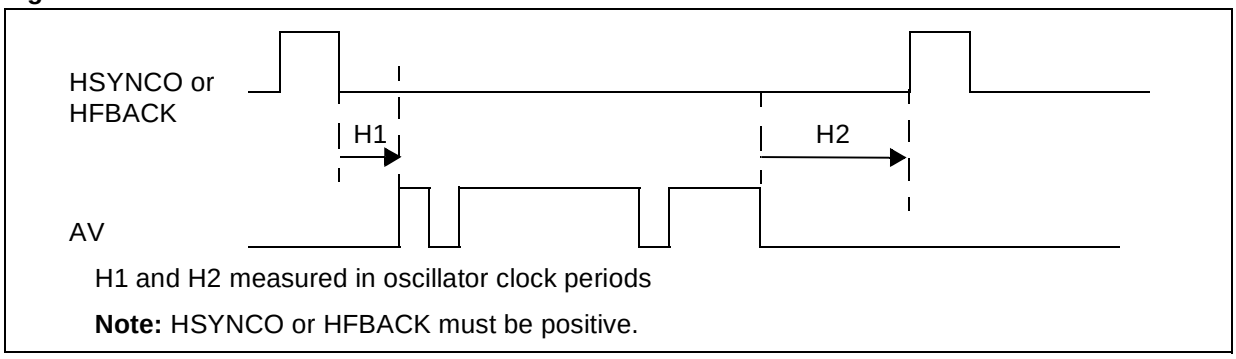
– If the AV signal is always low (no active video), the measured T1 or T2 values will also be 7FFh.

– If $T1 \leq 0$ (AV already high when the falling edge of the sync signal occurs), the measured T1 value will be fixed to 1.

– If $T2 \leq 0$ (AV still high when the rising edge of the sync signal occurs), a specific T2 value will be returned.

Note: Refer to Application Note AN1183 for further details.

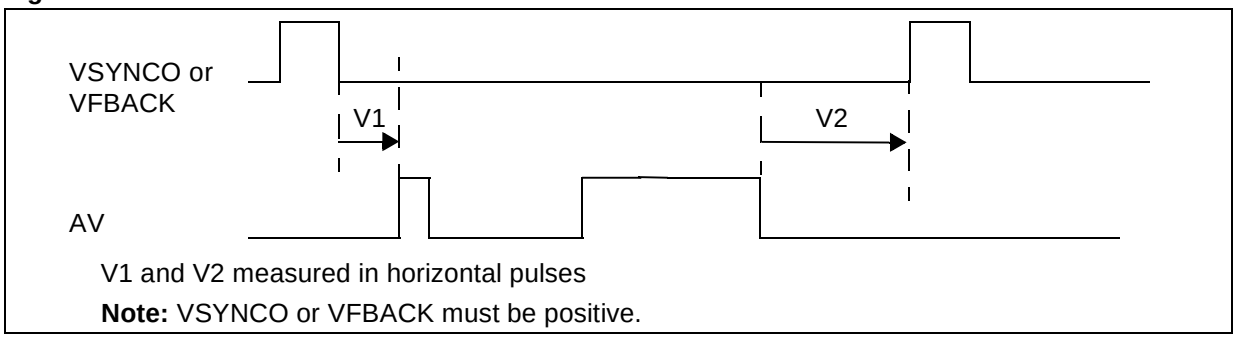
Figure 48. Horizontal Measurement



H1 and H2 measured in oscillator clock periods

Note: HSYNCO or HFBACK must be positive.

Figure 49. Vertical Measurement



V1 and V2 measured in horizontal pulses

Note: VSYNCO or VFBACK must be positive.

TIMING MEASUREMENT UNIT (Cont'd)

4.5.4 Register Description

CONTROL STATUS REGISTER (TMUCSR)

Bit 7:2 - Read only
Bit 1:0 - Read/Write
Reset Value: 1111 1100 (FCh)

7							0
T2[10]	T2[9]	T2[8]	T1[10]	T1[9]	T1[8]	H_V	START

Bit 7:5 = **T2[10:8]** *MSB of T2 Counter.*
Most Significant Bits of the T2 counter value (see T2 Counter register description).

Bit 4:2= **T1[10:8]** *MSB T1 Counter.*
Most Significant Bits of the T1 counter value (see T1 Counter register description).

Bit 1 = **H_V** *Horizontal or Vertical Measurement.*
This bit is set and cleared by software to select the type of measurement. It cannot be modified while the START bit = 1 (measurement in progress).

0: Vertical measurement.
1: Horizontal measurement.

Bit 0 = **START** *Start measurement.*
This bit is set by software and cleared by hardware when the measurements are completed. It can not be cleared by software.

0: Measurement done.
1: Start measurement.

T1 COUNTER REGISTER (TMUT1CR)

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the low part of the counter value.

7							0
T1[7]							T1[0]

When a T1 measurement is finished (rising edge on AV input), the 11-bit counter value is transferred to this register and to the T1[10:8] bits in the CSR register.

T1 is H1 value if the H_V bit = 1.

T1 is V1 value if the H_V bit = 0.

T2 COUNTER REGISTER (TMUT2CR)

Read Only
Reset Value: 1111 1111(FFh)

This is an 8-bit register that contains the low part of the counter value.

7							0
T2[7]							T2[0]

When a T2 measurement is finished (rising edge on the selected sync signal), the 11-bit counter value is transferred to this register and to the T2[10:8] bits in the CSR register.

T2 is H2 value if the H_V bit = 1.

T2 is V2 value if the H_V bit = 0.

TIMING MEASUREMENT UNIT (Cont'd)

Table 22. TMU Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0E	**CSR**	T2[10]	T2[9]	T2[8]	T1[10]	T1[9]	T1[8]	H_V	START
	Reset Value	1	1	1	1	1	1	0	0
0F	**T1CR**	T1[7]							T1[0]
	Reset Value	1	1	1	1	1	1	1	1
10	**T2CR**	T2[7]							T2[0]
	Reset Value	1	1	1	1	1	1	1	1

4.6 USB INTERFACE (USB)

4.6.1 Introduction

The USB Interface implements a low-speed function interface between the USB and the ST7 microcontroller. It is a highly integrated circuit which includes the transceiver, 3.3 voltage regulator, SIE and DMA. No external components are needed apart from the external pull-up on USBDM for low speed recognition by the USB host.

4.6.2 Main Features

- USB Specification Version 1.0 Compliant
- Supports Low-Speed USB Protocol
- Two or Three Endpoints (including default one) depending on the device (see device feature list and register map)
- CRC generation/checking, NRZI encoding/ decoding and bit-stuffing
- USB Suspend/Resume operations
- DMA Data transfers
- On-Chip 3.3V Regulator
- On-Chip USB Transceiver

4.6.3 Functional Description

The block diagram in Figure 50, gives an overview of the USB interface hardware.

For general information on the USB, refer to the “Universal Serial Bus Specifications” document available at http//:www.usb.org.

Serial Interface Engine

The SIE (Serial Interface Engine) interfaces with the USB, via the transceiver.

The SIE processes tokens, handles data transmission/reception, and handshaking as required by the USB standard. It also performs frame formatting, including CRC generation and checking.

Endpoints

The Endpoint registers indicate if the microcontroller is ready to transmit/receive, and how many bytes need to be transmitted.

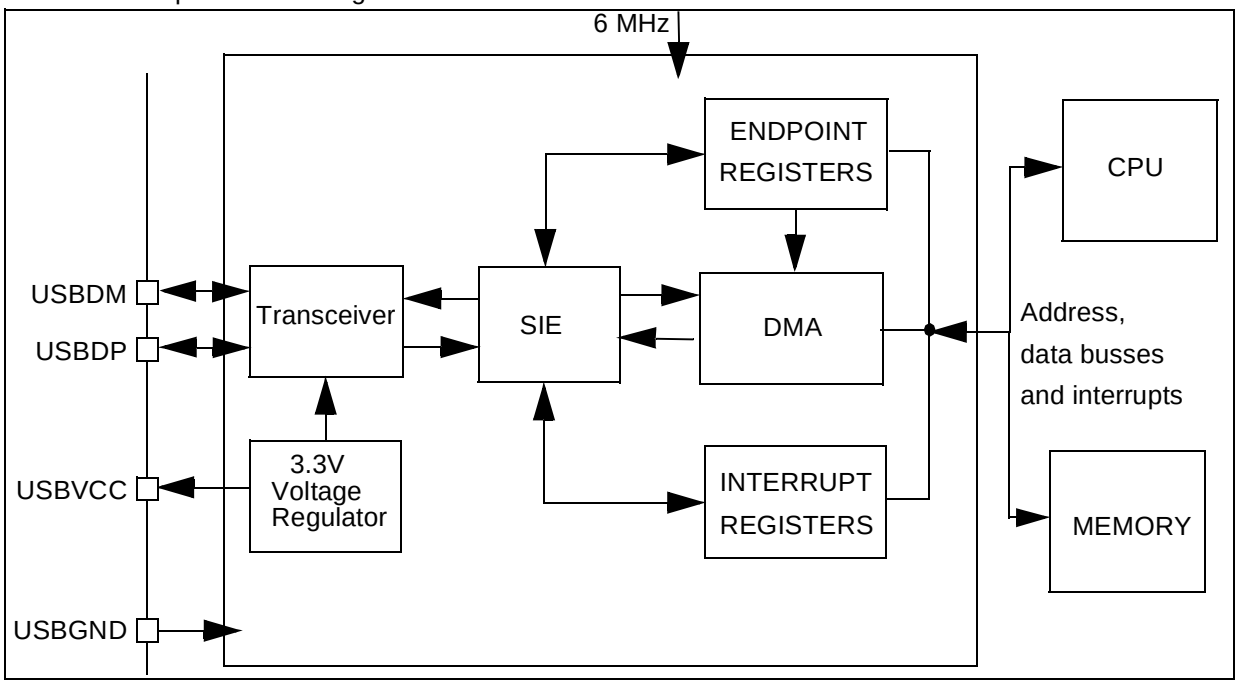
DMA

When a token for a valid Endpoint is recognized by the USB interface, the related data transfer takes place, using DMA. At the end of the transaction, an interrupt is generated.

Interrupts

By reading the Interrupt Status register, application software can know which USB event has occurred.

Figure 50. USB block diagram



USB INTERFACE (Cont'd)

4.6.4 Register Description

DMA ADDRESS REGISTER (DMAR)

Read / Write

Reset Value: Undefined

7							0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8

Bits 7:0=**DA[15:8]** *DMA address bits 15-8.*
See the description of bits DA7-6 in the next register (IDR).

INTERRUPT/DMA REGISTER (IDR)

Read / Write

Reset Value: xxxx 0000 (x0h)

7							0
DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0

Bits 7:6 = **DA[7:6]** *DMA address bits 7-6.*
The software must write the start address of the DMA memory area whose most significant bits are given by DA15-DA6. The remaining 6 address bits are set by hardware. See Figure 51.

Bits 5:4 = **EP[1:0]** *Endpoint number* (read-only). These bits identify the endpoint which required attention.

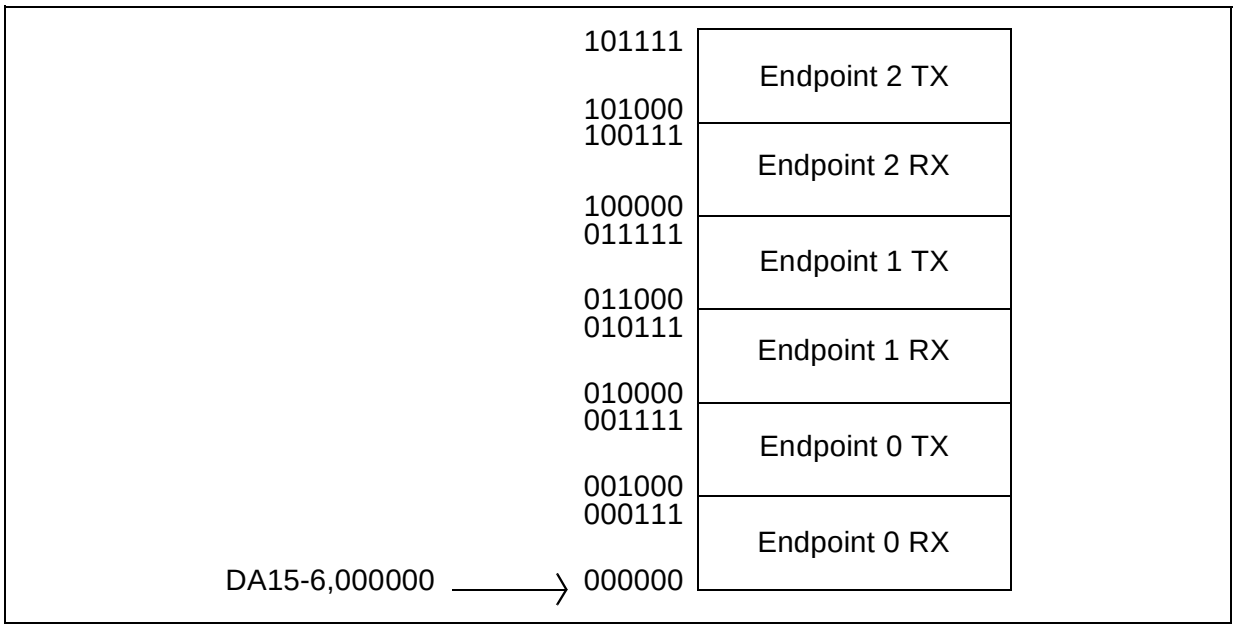
00: Endpoint 0
01: Endpoint 1
10: Endpoint 2

When a CTR interrupt occurs (see register ISTR) the software should read the EP bits to identify the endpoint which has sent or received a packet.

Bits 3:0 = **CNT[3:0]** *Byte count* (read only).
This field shows how many data bytes have been received during the last data reception.

Note: Not valid for data transmission.

Figure 51. DMA buffers



USB INTERFACE (Cont'd)
PID REGISTER (PIDR)

Read only

Reset Value: xx00 0000 (x0h)

7							0
TP3	TP2	0	0	0	0	0	0

Bits 7:6 =**TP3-TP2** *Token PID bits 3 & 2*.
USB token PIDs are encoded in four bits. **TP3-TP2** correspond to the variable token PID bits 3 & 2.

Note: PID bits 1 & 0 have a fixed value of 01.
When a CTR interrupt occurs (see register ISTR) the software should read the TP3 and TP2 bits to retrieve the PID name of the token received.

The USB standard defines TP bits as:

TP3	TP2	PID Name
0	0	OUT
1	0	IN
1	1	SETUP

Bit 5:0 Reserved. Forced by hardware to 0.

INTERRUPT STATUS REGISTER (ISTR)

Read / Write
Reset Value: 0000 0000 (00h)

7							0
0	DOVR	CTR	ERR	IOVR	ESUSP	RESET	SOF

When an interrupt occurs these bits are set by hardware. Software must read them to determine the interrupt type and clear them after servicing.

Note: These bits cannot be set by software.

Bit 7 = Reserved. Forced by hardware to 0.

Bit 6 = **DOVR** *DMA over/underrun*.
This bit is set by hardware if the ST7 processor can't answer a DMA request in time.

0: No over/underrun detected
1: Over/underrun detected

Bit 5 = **CTR** *Correct Transfer.* This bit is set by hardware when a correct transfer operation is performed. The type of transfer can be determined by looking at bits TP3-TP2 in register PIDR. The Endpoint on which the transfer was made is identified by bits EP1-EP0 in register IDR.

0: No Correct Transfer detected
1: Correct Transfer detected

Note: A transfer where the device sent a NAK or STALL handshake is considered not correct (the host only sends ACK handshakes). A transfer is considered correct if there are no errors in the PID and CRC fields, if the DATA0/DATA1 PID is sent as expected, if there were no data overruns, bit stuffing or framing errors.

Bit 4 = **ERR** *Error.*
This bit is set by hardware whenever one of the errors listed below has occurred:

0: No error detected
1: Timeout, CRC, bit stuffing or nonstandard framing error detected

Bit 3 = **IOVR** *Interrupt overrun.*
This bit is set when hardware tries to set ERR, ESUSP or SOF before they have been cleared by software.

0: No overrun detected
1: Overrun detected

Bit 2 = **ESUSP** *End suspend mode*.
This bit is set by hardware when, during suspend mode, activity is detected that wakes the USB interface up from suspend mode.

This interrupt is serviced by a specific vector.

0: No End Suspend detected
1: End Suspend detected

Bit 1 = **RESET** *USB reset.*
This bit is set by hardware when the USB reset sequence is detected on the bus.

0: No USB reset signal detected
1: USB reset signal detected

Note: The DADDR, EP0RA, EP0RB, EP1RA, EP1RB, EP2RA and EP2RB registers are reset by a USB reset.

USB INTERFACE (Cont'd)

Bit 0 = **SOF** *Start of frame.*
This bit is set by hardware when a low-speed SOF indication (keep-alive strobe) is seen on the USB bus.

0: No SOF signal detected
1: SOF signal detected

Note: To avoid spurious clearing of some bits, it is recommended to clear them using a load instruction where all bits which must not be altered are set, and all bits to be cleared are reset. Avoid read-modify-write instructions like AND , XOR..

INTERRUPT MASK REGISTER (IMR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM

Bit 7 = Reserved. Forced by hardware to 0.

Bits 6:0 = These bits are mask bits for all interrupt condition bits included in the ISTR. Whenever one of the IMR bits is set, if the corresponding ISTR bit is set, and the I bit in the CC register is cleared, an interrupt request is generated. For an explanation of each bit, please refer to the corresponding bit description in ISTR.

CONTROL REGISTER (CTLR)

Read / Write

Reset Value: 0000 0110 (06h)

7							0
0	0	0	0	RESUME	PDWN	SUSP	FRES

Bits 7:4 = Reserved. Forced by hardware to 0.

Bit 3 = **RESUME** *Resume.*
This bit is set by software to wake-up the Host when the ST7 is in suspend mode.

0: Resume signal not forced
1: Resume signal forced on the USB bus.

Software should clear this bit after the appropriate delay.

Bit 2 = **PDWN** *Power down.*
This bit is set by software to turn off the 3.3V on-chip voltage regulator that supplies the external pull-up resistor and the transceiver.

0: Voltage regulator on
1: Voltage regulator off

Note: After turning on the voltage regulator, software should allow at least 3 µs for stabilisation of the power supply before using the USB interface.

Bit 1 = **SUSP** *Suspend mode.*
This bit is set by software to enter Suspend mode.

0: Suspend mode inactive
1: Suspend mode active

When the hardware detects USB activity, it resets this bit (it can also be reset by software).

Bit 0 = **FRES** *Force reset.*
This bit is set by software to force a reset of the USB interface, just as if a RESET sequence came from the USB.

0: Reset not forced
1: USB interface reset forced.

The USB is held in RESET state until software clears this bit, at which point a "USB-RESET" interrupt will be generated if enabled.

DEVICE ADDRESS REGISTER (DADDR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

Bit 7 = Reserved. Forced by hardware to 0.

Bits 6:0 = **ADD[6:0]** *Device address, 7 bits.*

Software must write into this register the address sent by the host during enumeration.

Note: This register is also reset when a USB reset is received from the USB bus or forced through bit FRES in the CTLR register.

USB INTERFACE (Cont'd)
ENDPOINT n REGISTER A (EPnRA)

Read / Write

Reset Value: 0000 xxxx (0xh)

7							0
ST_ OUT	DTOG _TX	STAT _TX1	STAT _TX0	TBC 3	TBC 2	TBC 1	TBC 0

These registers (**EP0RA**, **EP1RA** and **EP2RA**) are used for controlling data transmission. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RA register are not available on some devices (see device feature list and register map).

Bit 7 = **ST_OUT** *Status out.*
This bit is set by software to indicate that a status out packet is expected: in this case, all nonzero OUT data transfers on the endpoint are STALLed instead of being ACKed. When ST_OUT is reset, OUT transactions can have any number of bytes, as needed.

Bit 6 = **DTOG_TX** *Data Toggle, for transmission transfers.*
It contains the required value of the toggle bit (0=DATA0, 1=DATA1) for the next transmitted data packet. This bit is set by hardware at the reception of a SETUP PID. DTOG_TX toggles only when the transmitter has received the ACK signal from the USB host. DTOG_TX and also DTOG_RX (see EPnRB) are normally updated by hardware, at the receipt of a relevant PID. They can be also written by software.

Bits 5:4 = **STAT_TX[1:0]** *Status bits, for transmission transfers.*
These bits contain the information about the endpoint status, which are listed below:

STAT_TX1	STAT_TX0	Meaning
0	0	**DISABLED:** transmission transfers cannot be executed.
0	1	**STALL**: the endpoint is stalled and all transmission requests result in a STALL handshake.
1	0	**NAK**: the endpoint is naked and all transmission requests result in a NAK handshake.
1	1	**VALID**: this endpoint is enabled for transmission.

These bits are written by software. Hardware sets the STAT_TX bits to NAK when a correct transfer has occurred (CTR=1) related to a IN or SETUP transaction addressed to this endpoint; this allows the software to prepare the next set of data to be transmitted.

Bits 3:0 = **TBC[3:0]** *Transmit byte count for Endpoint n.*
Before transmission, after filling the transmit buffer, software must write in the TBC field the transmit packet size expressed in bytes (in the range 0-8).

USB INTERFACE (Cont'd)
ENDPOINT n REGISTER B (EPnRB)

Read / Write

Reset Value: 0000 xxxx (0xh)

7							0
CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0

These registers (**EP1RB** and **EP2RB**) are used for controlling data reception on Endpoints 1 and 2. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RB register are not available on some devices (see device feature list and register map).

Bit 7 = **CTRL** *Control.*
This bit should be 0.

Note: If this bit is 1, the Endpoint is a control endpoint. (Endpoint 0 is always a control Endpoint, but it is possible to have more than one control Endpoint).

Bit 6 = **DTOG_RX** *Data toggle, for reception transfers*.
It contains the expected value of the toggle bit (0=DATA0, 1=DATA1) for the next data packet. This bit is cleared by hardware in the first stage (Setup Stage) of a control transfer (SETUP transactions start always with DATA0 PID). The receiver toggles DTOG_RX only if it receives a correct data packet and the packet's data PID matches the receiver sequence bit.

Bit 5:4 = **STAT_RX [1:0]** *Status bits, for reception transfers.*
These bits contain the information about the endpoint status, which are listed in the following table:

STAT_RX1	STAT_RX0	Meaning
0	0	**DISABLED**: reception transfers cannot be executed.
0	1	**STALL:** the endpoint is stalled and all reception requests result in a STALL handshake.
1	0	**NAK**: the endpoint is naked and all reception requests result in a NAK handshake.
1	1	**VALID**: this endpoint is enabled for reception.

These bits are written by software. Hardware sets the STAT_RX bits to NAK when a correct transfer has occurred (CTR=1) related to an OUT or SETUP transaction addressed to this endpoint, so the software has the time to elaborate the received data before acknowledging a new transaction.

Bits 3:0 = **EA[3:0]** *Endpoint address*.
Software must write in this field the 4-bit address used to identify the transactions directed to this endpoint. Usually EP1RB contains "0001" and EP2RB contains "0010".

ENDPOINT 0 REGISTER B (EP0RB)

Read / Write

Reset Value: 1000 0000 (80h)

7							0
1	DTOG_RX	STAT_RX1	STAT_RX0	0	0	0	0

This register is used for controlling data reception on Endpoint 0. It is also reset by the USB bus reset.

Bit 7 = Forced by hardware to 1.

Bit 6:4 = Refer to the EPnRB register for a description of these bits.

Bit 3:0 = Forced by hardware to 0.

USB INTERFACE (Cont'd)

4.6.5 Programming Considerations

In the following, the interaction between the USB interface and the application program is described. Apart from system reset, action is always initiated by the USB interface, driven by one of the USB events associated with the Interrupt Status Register (ISTR) bits.

4.6.5.1 Initializing the Registers

At system reset, the software must initialize all registers to enable the USB interface to properly generate interrupts and DMA requests.

1. Initialize the DMAR, IDR, and IMR registers (choice of enabled interrupts, address of DMA buffers). Refer the paragraph titled initializing the DMA Buffers.
2. Initialize the EP0RA and EP0RB registers to enable accesses to address 0 and endpoint 0 to support USB enumeration. Refer to the paragraph titled Endpoint Initialization.
3. When addresses are received through this channel, update the content of the DADDR.
4. If needed, write the endpoint numbers in the EA fields in the EP1RB and EP2RB register.

4.6.5.2 Initializing DMA buffers

The DMA buffers are a contiguous zone of memory whose maximum size is 48 bytes. They can be placed anywhere in the memory space, typically in RAM, to enable the reception of messages. The 10 most significant bits of the start of this memory area are specified by bits DA15-DA6 in registers DMAR and IDR, the remaining bits are 0. The memory map is shown in Figure 51.

Each buffer is filled starting from the bottom (last 3 address bits=000) up.

4.6.5.3 Endpoint Initialization

To be ready to receive:

Set STAT_RX to VALID (11b) in EP0RB to enable reception.

To be ready to transmit:

1. Write the data in the DMA transmit buffer.
2. In register EPnRA, specify the number of bytes to be transmitted in the TBC field
3. Enable the endpoint by setting the STAT_TX bits to VALID (11b) in EPnRA.

Note: Once transmission and/or reception are enabled, registers EPnRA and/or EPnRB (respectively) must not be modified by software, as the hardware can change their value on the fly.

When the operation is completed, they can be accessed again to enable a new operation.

4.6.5.4 Interrupt Handling

Start of Frame (SOF)

The interrupt service routine must monitor the SOF events and measure the interval between each SOF event. If 3ms pass without a SOF event, the software should set the USB interface to suspend mode.

USB Reset (RESET)

When this event occurs, the DADDR register is reset, and communication is disabled in all endpoint registers (the USB interface will not respond to any packet). Software is responsible for reenabling endpoint 0 within 10 ms of the end of reset. To do this you set the STAT_RX bits in the EP0RB register to VALID.

End Suspend (ESUSP)

The CPU is alerted by activity on the USB, which causes an ESUSP interrupt.

Correct Transfer (CTR)

1. When this event occurs, the hardware automatically sets the STAT_TX or STAT_RX to NAK.

Note: Every valid endpoint is NAKed until software clears the CTR bit in the ISTR register, independently of the endpoint number addressed by the transfer which generated the CTR interrupt.

Note: If the event triggering the CTR interrupt is a SETUP transaction, both STAT_TX and STAT_RX are set to NAK.

2. Read the PIDR to obtain the token and the IDR to get the endpoint number related to the last transfer.

Note: When a CTR interrupt occurs, the TP3-TP2 bits in the PIDR register and EP1-EP0 bits in the IDR register stay unchanged until the CTR bit in the ISTR register is cleared.

3. Clear the CTR bit in the ISTR register.

USB INTERFACE (Cont'd)

Table 23. USB Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
25	PIDR	TP3	TP2	0	0	RX_SEZ	RXD	0	0
	Reset Value	x	x	0	0	0	0	0	0
26	DMAR	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
	Reset Value	x	x	x	x	x	x	x	x
27	IDR	DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0
	Reset Value	x	x	x	x	0	0	0	0
28	ISTR	SUSP	DOVR	CTR	ERR	IOVR	ESUSP	RESET	SOF
	Reset Value	0	0	0	0	0	0	0	0
29	IMR	SUSPM	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM
	Reset Value	0	0	0	0	0	0	0	0
2A	CTLR	0	0	0	0	RESUME	PDWN	FSUSP	FRES
	Reset Value	0	0	0	0	0	1	1	0
2B	DADDR	0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	Reset Value	0	0	0	0	0	0	0	0
2C	EP0RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset Value	0	0	0	0	x	x	x	x
2D	EP0RB	1	DTOG_RX	STAT_RX1	STAT_RX0	0	0	0	0
	Reset Value	1	0	0	0	0	0	0	0
2E	EP1RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset Value	0	0	0	0	x	x	x	x
2F	EP1RB	CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0
	Reset Value	0	0	0	0	x	x	x	x
30	EP2RA	ST_OUT	DTOG_TX	STAT_TX1	STAT_TX0	TBC3	TBC2	TBC1	TBC0
	Reset Value	0	0	0	0	x	x	x	x
31	EP2RB	CTRL	DTOG_RX	STAT_RX1	STAT_RX0	EA3	EA2	EA1	EA0
	Reset Value	0	0	0	0	x	x	x	x

4.7 I²C SINGLE MASTER BUS INTERFACE (I2C)

4.7.1 Introduction

The I^2C Bus Interface serves as an interface between the microcontroller and the serial I^2C bus. It provides single master functions, and controls all I^2C bus-specific sequencing, protocol and timing. It supports fast I²C mode (400kHz).

4.7.2 Main Features

- Parallel bus/I^2C protocol converter
- Interrupt generation
- Standard I^2C mode/Fast I^2C mode
- 7-bit Addressing

■ **I^2C single Master Mode**

- End of byte transmission flag
- Transmitter/Receiver flag
- Clock generation

4.7.3 General Description

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa, using either an interrupt or polled handshake. The interrupts are enabled or disabled by software. The interface is connected to the I^2C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I^2C bus and a Fast I^2C bus. This selection is made by software.

Mode Selection

The interface can operate in the two following modes:

– Master transmitter/receiver
By default, it is idle.

The interface automatically switches from idle to master after it generates a START condition and from master to idle after it generates a STOP condition.

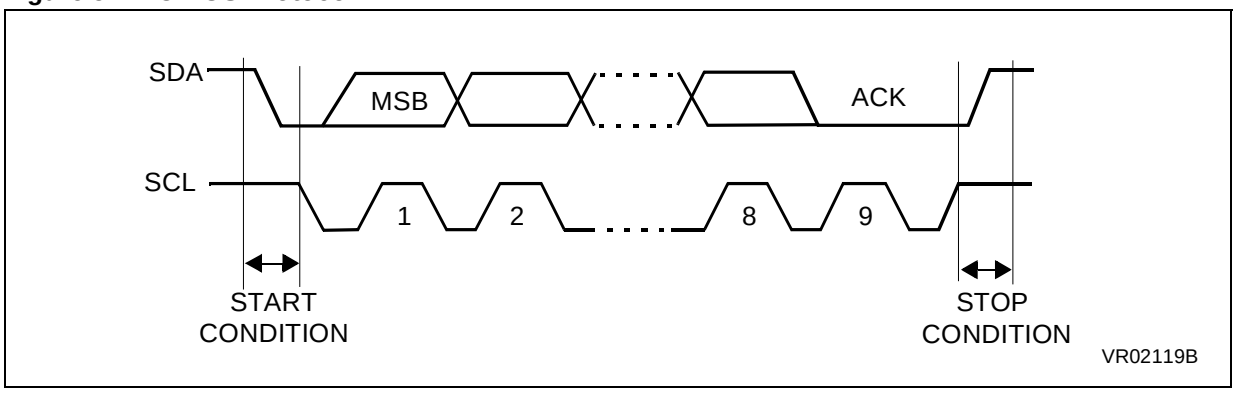
Communication Flow

The interface initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte following the start condition is the address byte.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to Figure 52.

Figure 52. I^2C BUS Protocol



I²C SINGLE MASTER BUS INTERFACE (Cont'd)
Acknowledge may be enabled and disabled by software.

The speed of the I^2C interface may be selected between Standard (0-100KHz) and Fast I^2C (100-400KHz).

SDA/SCL Line Control

Transmitter mode: the interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the Data Register.

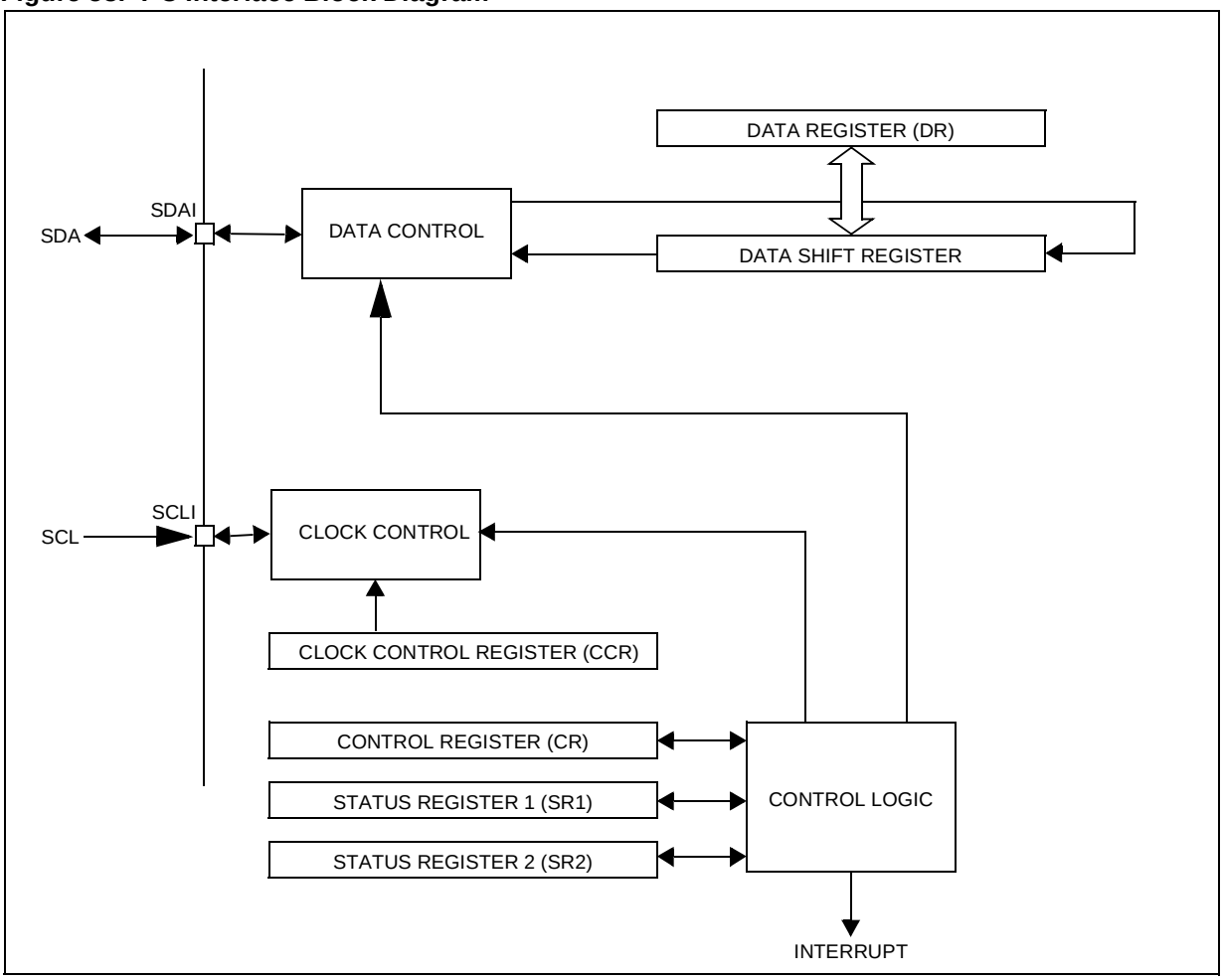
Receiver mode: the interface holds the clock line low after reception to wait for the microcontroller to read the byte in the Data Register.

The SCL frequency (F_{scl}) is controlled by a programmable clock divider which depends on the I^2C bus mode.

When the I2C cell is enabled, the SDA and SCL ports must be configured as floating open-drain output or floating input. In this case, the value of the external pull-up resistance used depends on the application.

When the I2C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 53. I^2C Interface Block Diagram



I²C SINGLE MASTER BUS INTERFACE (Cont'd)

4.7.4 Functional Description (Master Mode)

Refer to the CR, SR1 and SR2 registers in Section 4.7.5. for the bit definitions.

By default the I^2C interface operates in idle mode (M/IDL bit is cleared) except when it initiates a transmit or receive sequence.

To switch from default idle mode to Master mode a Start condition generation is needed.

Start condition and Transmit Slave address

Setting the START bit causes the interface to switch to Master mode (M/IDL bit set) and generates a Start condition.

Once the Start condition is sent:

– The EVF and SB bits are set by hardware with an interrupt if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address byte, **holding the SCL line low** (see Figure 54 Transfer sequencing EV1).

Then the slave address byte is sent to the SDA line via the internal shift register.

After completion of this transfer (and acknowledge from the slave if the ACK bit is set):

– The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see Figure 54 Transfer sequencing EV2).

Next the master must enter Receiver or Transmitter mode.

Master Receiver

Following the address transmission and after SR1 and CR registers have been accessed, the master receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

– Acknowledge pulse if if the ACK bit is set

– EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see Figure 54 Transfer sequencing EV3).

To close the communication: before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to idle mode (M/IDL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

Master Transmitter

Following the address transmission and after SR1 register has been read, the master sends bytes from the DR register to the SDA line via the internal shift register.

The master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 54 Transfer sequencing EV4).

When the acknowledge bit is received, the interface sets:

– EVF and BTF bits with an interrupt if the ITE bit is set.

To close the communication: after writing the last byte to the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to idle mode (M/IDL bit cleared).

Error Case

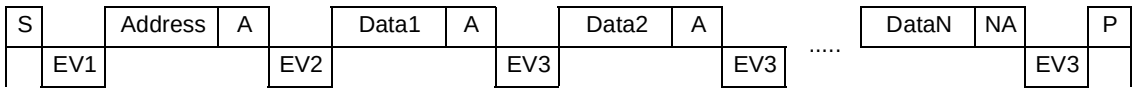
– **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the START or STOP bit.

Note: The SCL line is not held low.

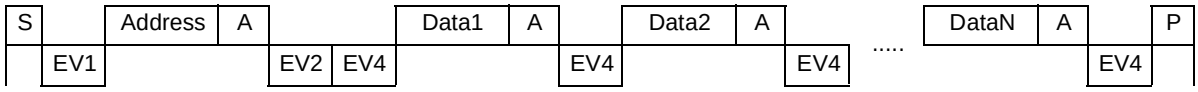
I²C SINGLE MASTER BUS INTERFACE (Cont'd)

Figure 54. Transfer Sequencing

Master receiver:



Master transmitter:



Legend:

S=Start, P=Stop, A=Acknowledge, NA=Non-acknowledge

EVx=Event (with interrupt if ITE=1)

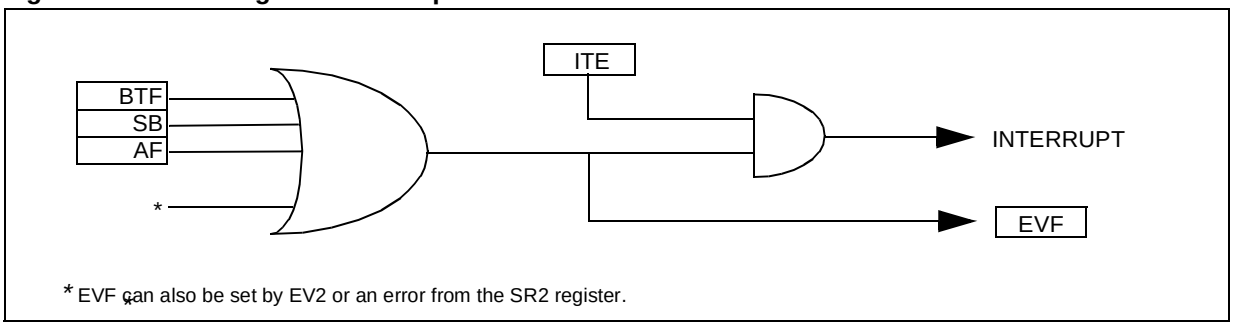
EV1: EVF=1, SB=1, cleared by reading SR1 register followed by writing DR register.

EV2: EVF=1, cleared by reading SR1 register followed by writing CR register (for example PE=1).

EV3: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV4: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

Figure 55. Event Flags and Interrupt Generation



I²C SINGLE MASTER BUS INTERFACE (Cont'd)

4.7.5 Register Description

I²C CONTROL REGISTER (CR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	PE	0	START	ACK	STOP	ITE

Bit 7:6 = Reserved. Forced to 0 by hardware.

Bit 5 = **PE** *Peripheral enable.*
This bit is set and cleared by software.

0: Peripheral disabled
1: Master capability

Note: When PE=0, all the bits of the CR register and the SR register except the Stop bit are reset. All outputs are released while PE=0

Note: When PE=1, the corresponding I/O pins are selected by hardware as alternate functions.

Note: To enable the I²C interface, write the CR register **TWICE** with PE=1 as the first write only activates the interface (only PE is set).

Bit 4 = Reserved. Forced to 0 by hardware.

Bit 3 = **START** *Generation of a Start condition*. This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0) or when the Start condition is sent (with interrupt generation if ITE=1).

In master mode:

0: No start generation
1: Repeated start generation

In idle mode:

0: No start generation
1: Start generation when the bus is free

Bit 2 = **ACK** *Acknowledge enable.*
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0).

0: No acknowledge returned
1: Acknowledge returned after a data byte is received

Bit 1 = **STOP** *Generation of a Stop condition*.
This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0) or when the Stop condition is sent.

In Master mode only:

0: No stop generation
1: Stop generation after the current byte transfer or after the current Start condition is sent.

Bit 0 = **ITE** *Interrupt enable.*
This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE=0).

0: Interrupts disabled
1: Interrupts enabled

Refer to Figure 55 for the relationship between the events and the interrupt.
SCL is held low when the SB or BTF flags or an EV2 event (See Figure 54) is detected.

I²C SINGLE MASTER BUS INTERFACE (Cont'd)
I²C STATUS REGISTER 1 (SR1)

Read Only

Reset Value: 0000 0000 (00h)

7							0
EVF	0	TRA	0	BTF	0	M/IDL	SB

Bit 7 = **EVF** *Event flag.*
This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in Figure 54. It is also cleared by hardware when the interface is disabled (PE=0).

0: No event
1: One of the following events has occurred:
– BTF=1 (Byte received or transmitted)
– SB=1 (Start condition generated)
– AF=1 (No acknowledge received after byte transmission if ACK=1)
– Address byte successfully transmitted.

Bit 6 = Reserved. Forced to 0 by hardware.

Bit 5 = **TRA** *Transmitter/Receiver.*
When BTF is set, TRA=1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware when the interface is disabled (PE=0).

0: Data byte received (if BTF=1)
1: Data byte transmitted

Bit 4 = Reserved. Forced to 0 by hardware.

Bit 3 = **BTF** *Byte transfer finished.*
This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE=1. It is cleared by software reading SR1 register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE=0).

– Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV2 event (See Figure 54). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.
– Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

0: Byte transfer not done
1: Byte transfer succeeded

Bit 2 = Reserved. Forced to 0 by hardware.

Bit 1 = **M/IDL** *Master/Idle.*
This bit is set by hardware as soon as the interface is in Master mode (writing START=1). It is cleared by hardware after generating a Stop condition on the bus. It is also cleared when the interface is disabled (PE=0).

0: Idle mode
1: Master mode

Bit 0 = **SB** *Start bit generated.*
This bit is set by hardware as soon as the Start condition is generated (following a write START=1). An interrupt is generated if ITE=1. It is cleared by software reading SR1 register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE=0).

0: No Start condition
1: Start condition generated

I²C SINGLE MASTER BUS INTERFACE (Cont'd)

I²C STATUS REGISTER 2 (SR2)

Read Only

Reset Value: 0000 0000 (00h)

7							0
0	0	0	AF	0	0	0	0

Bit 7:5 = Reserved. Forced to 0 by hardware.

Bit 4 = **AF** *Acknowledge failure*.
This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while AF=1.

0: No acknowledge failure

1: Acknowledge failure

Bit 3:0 = Reserved. Forced to 0 by hardware.

I²C CLOCK CONTROL REGISTER (CCR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0

Bit 7 = **FM/SM** *Fast/Standard I²C mode.*
This bit is set and cleared by software. It is not cleared when the interface is disabled (PE=0).

0: Standard I²C mode
1: Fast I²C mode

Bit 6:0 = **CC6-CC0** *7-bit clock divider.*
These bits select the speed of the bus (F_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE=0).

– Standard mode (FM/SM=0): F_{SCL} <= 100kHz
$F_{SCL} = F_{CPU}/(2x([CC6..CC0]+2))$

– Fast mode (FM/SM=1): F_{SCL} > 100kHz
$F_{SCL} = F_{CPU}/(3x([CC6..CC0]+2))$

Note: The programmed F_{SCL} assumes no load on SCL and SDA lines.

I²C DATA REGISTER (DR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D7-D0** *8-bit Data Register.*
These bits contains the byte to be received or transmitted on the bus.

– Transmitter mode: Byte transmission start automatically when the software writes in the DR register.

– Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address.
Then, the next data bytes are received one by one after reading the DR register.

I2C SINGLE MASTER BUS INTERFACE (Cont'd)

Table 24. I²C Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
5F	CR			PE		START	ACK	STOP	ITE
	Reset Value	0	0	0	0	0	0	0	0
5E	SR1	EVF		TRA		BTF		M/IDL	SB
	Reset Value	0	0	0	0	0	0	0	0
5D	SR2				AF				
	Reset Value	0	0	0	0	0	0	0	0
5C	CCR	FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0
	Reset Value	0	0	0	0	0	0	0	0
59	DR	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
	Reset Value	0	0	0	0	0	0	0	0

4.8 DDC INTERFACE (DDC)

4.8.1 Introduction

The DDC (Display Data Channel) Bus Interface is mainly used by the monitor to identify itself to the video controller, by the monitor manufacturer to perform factory alignment, and by the user to adjust the monitor's parameters.

The DDC interface consists of two parts:

- A fully hardware-implemented interface, supporting DDC1 and DDC2B (VESA specification 3.0 compliant). It accesses the ST7 on-chip memory directly through a built-in DMA engine.
- A second interface, supporting the slave I^2C functions for handling DDC/CI mode (DDC2Bi), factory alignment or Enhanced DDC (EDDC) by software.

4.8.2 DDC Interface Features

4.8.2.1 Hardware DDC1/2B Interface Features

- Full hardware support for DDC1/2B communications (VESA specification versions 2 and 3)
- Hardware detection of DDC2B addresses A0h/A1h and optionally A2h/A3h (P&D) or A6h/A7h (FPDI-2)
- Separate mapping of EDID version 1 (128 bytes) and EDID version 2 (256 bytes) when both must coexist
- Support for error recovery mechanism
- Detection of misplaced Start and Stop conditions
- I^2C byte, random and sequential read modes
- DMA transfer from any memory location and to RAM
- Automatic memory address incrementation
- End of data downloading flag and interrupt capability

4.8.2.2 DDC/CI - Factory Interface Features

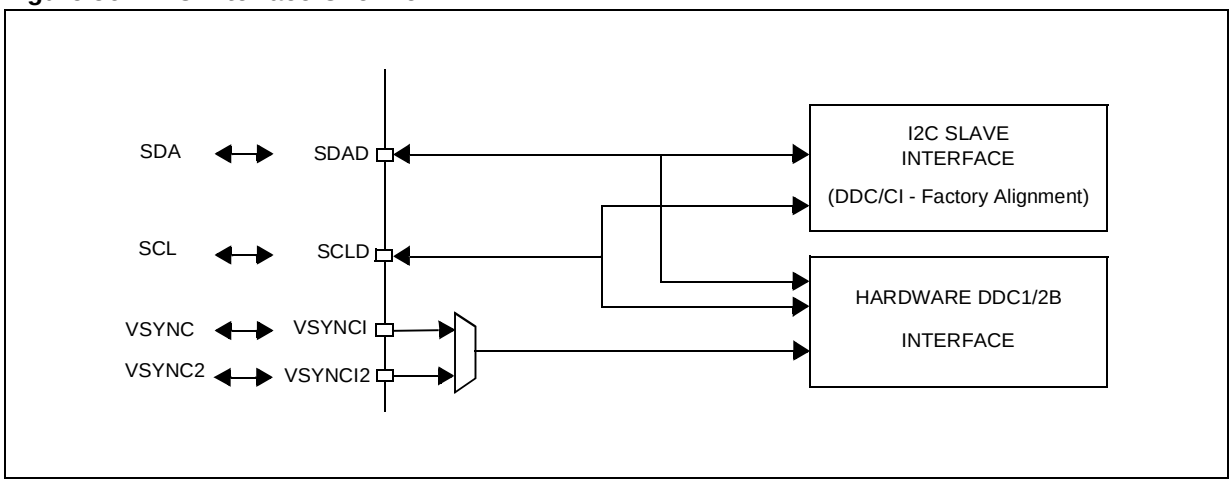
General I^2C Features:

- Parallel bus /I^2C protocol converter
- Interrupt generation
- Standard I^2C mode/Fast I^2C mode
- 7-bit Addressing

I^2C Slave Features:

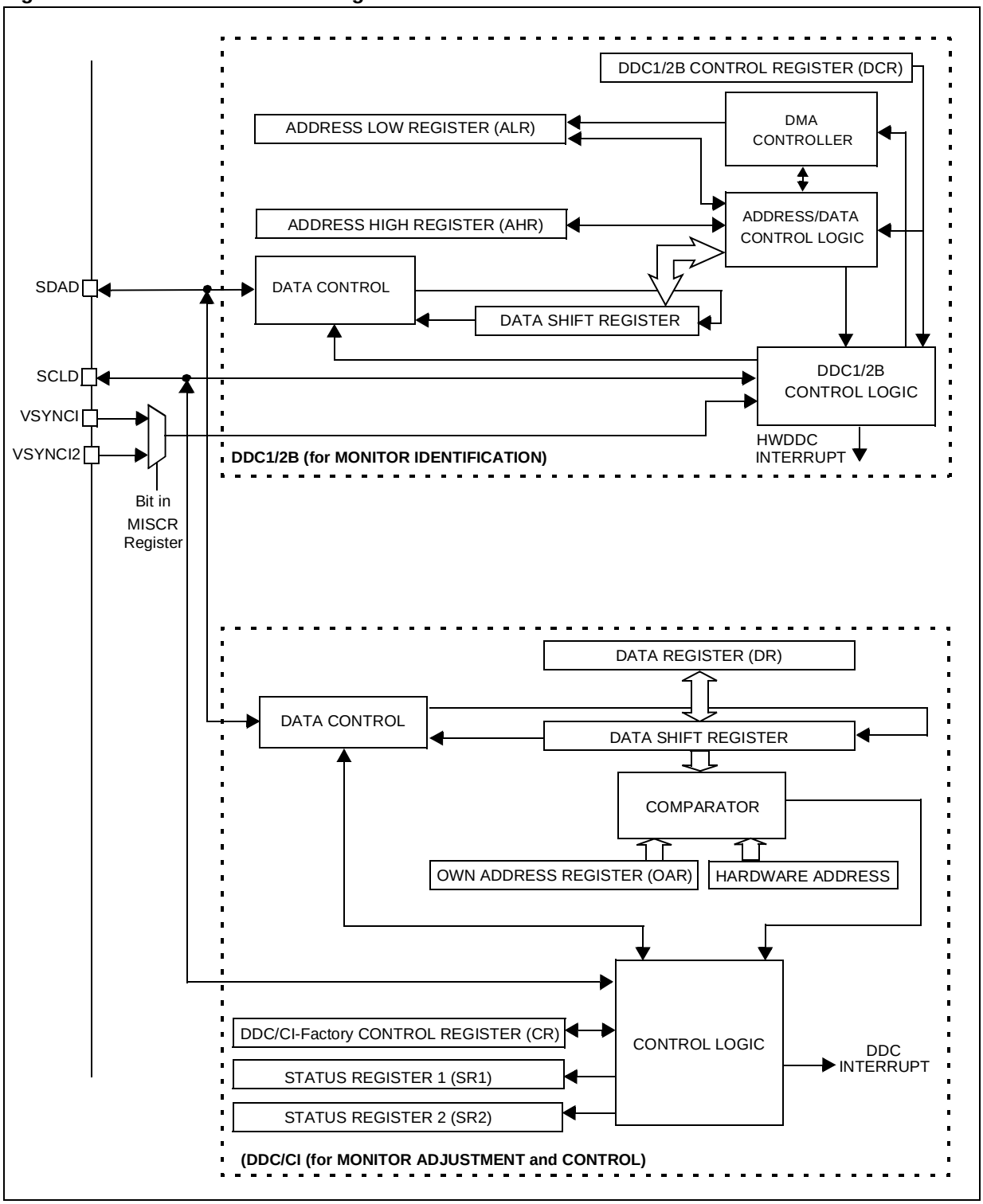
- I^2C bus busy flag
- Start bit detection flag
- Detection of misplaced Start or Stop condition
- Transfer problem detection
- Address Matched detection
- Programmable Address detection and/or Hardware detection of Enhanced DDC (EDDC) addresses (60h/61h)
- End of byte transmission flag
- Transmitter/Receiver flag
- Stop condition Detection

Figure 56. DDC Interface Overview



DDC INTERFACE (Cont'd)

Figure 57. DDC Interface Block Diagram



DDC INTERFACE (Cont'd)

4.8.3 Signal Description

Serial Data (SDA)

The SDA bidirectional pin is used to transfer data in and out of the device. It is an open-drain output that may be or-wired with other open-drain or open-collector pins. An external pull-up resistor must be connected to the SDA line. Its value depends on the load of the line and the transfer rate.

Serial Clock (SCL)

The SCL input pin is used to synchronize all data in and out of the device when in I^2C bidirectional mode. An external pull-up resistor must be connected to the SCL line. Its value depends on the load of the line and the transfer rate.

Note: When the DDC1/2B and DDC/CI-Factory Interfaces are disabled (HWPE bit=0 in the DCR register and PE bit=0 in the CR register), SDA and SCL pins revert to standard I/O pins.

Transmit-only Clock (Vsync/Vsync2)

The Vsync input pins are used to synchronize all data in and out of the device when in Transmit-only mode.

These pins are ONLY used by the DDC1/2B interface (when in DDC1 mode).

DDC INTERFACE (Cont'd)

4.8.4 I²C BUS Protocol

A standard I²C communication is normally based on four parts: START condition, device slave address transmission, data transfer and STOP condition. They are described brielfly in the following section and illustrated in Figure 58 (for more details, refer to the I²C bus specification).

4.8.4.1 START condition

When the bus is free (both SCL and SDA lines are at a high level), a master can initiate a communication by sending a START signal. This signal is defined as a high-to-low transition of SDA while SCL is stable high. The bus is considered to be busy after a START condition.

This START condition must precede any command for data transfer.

4.8.4.2 Slave Address Transmission

The first byte following a START condition is the slave address transmitted by the master. This address is 7-bit long followed by an 8th bit (Least significant bit: LSB) which is the data direction bit (R/$\overline{W}$ bit).

– A "0" indicates a transmission (WRITE) from the master to the slave.

– A "1" indicates a request for data (READ) from the slave to the master.

If a slave device is present on the bus at the given address, an Acknowledge will be generated on the 9th clock pulse.

4.8.4.3 Data Transfer

Once the slave address is acknowledged, the data transfer can proceed in the direction given by the R/$\overline{W}$ bit sent in the address.

Data is transferred with the most significant bit (MSB) first. Data bits can be changed only when SCL is low and must be held stable when SCL is high.

One complete data byte transfer requires 9 clock pulses: 8 bits + 1 acknowledge bit.

4.8.4.4 Acknowledge Bit (ACK / NACK)

Every byte put on the SDA line is 8-bit long followed by an acknowledge bit.

This bit is used to indicate a successful data transfer. The bus transmitter, either master or slave, releases the SDA line during the 9th clock period (after sending all 8 bits of data), then:

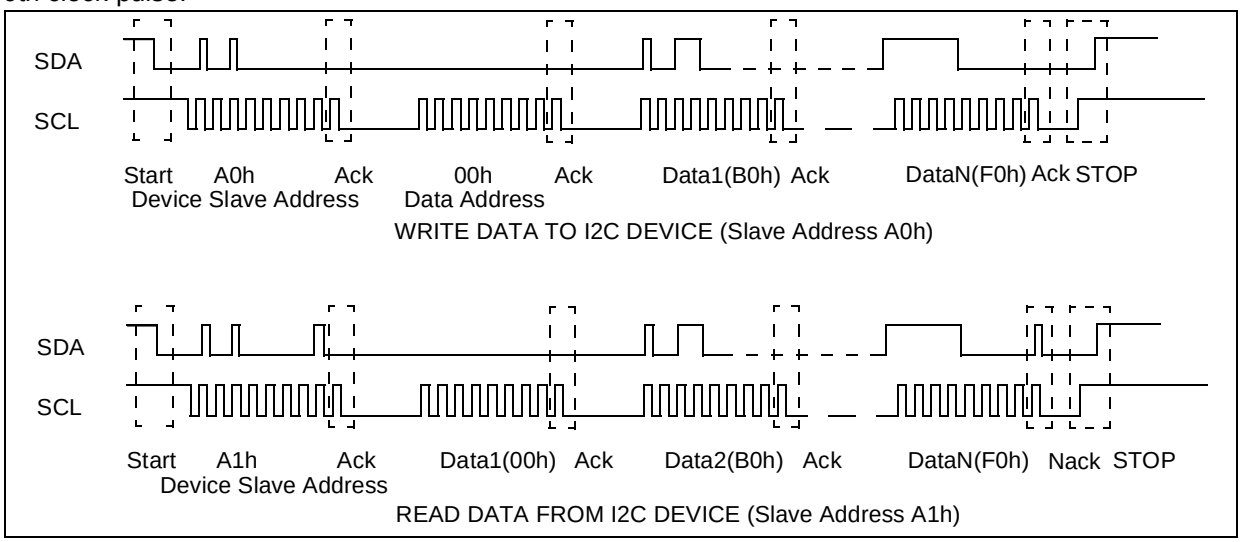
– To generate an Acknowledge (ACK) of the current byte, the receiver pulls the SDA line low.

– To generate a No-Acknowledge (NACK) of the current byte, the receiver releases the SDA line (hence at a high level).

4.8.4.5 STOP Condition

A STOP condition is defined by a low-to-high transition of SDA while SCL is stable high. It ends the communication between the Interface and the bus master.

Figure 58. I²C Signal Diagram



DDC INTERFACE (Cont'd)

4.8.5 DDC Standard

The DDC standard is divided in several data transfer protocols: DDC1, DDC2B, DDC/CI.

For DDC1/2B, refer to the "VESA DDC Standard v3.0" specification. For DDC/CI refer to the "VESA DDC Commands Interface v1.0"

– DDC1 is a uni-directional transmission of EDID v1 (128 bytes) from display to host clocked by VSYNCI.

– DDC2B is a uni-directional channel from display to host. The host computer uses base-level I^2C commands to read the EDID data from the display which is always in slave mode.
Specific types of display contain EDID at fixed I^2C device addresses within the device (refer to Table 25).

– DDC/CI is a bi-directional channel between the host computer and the display. The DDC/CI offers a display control interface based on I^2C bus. It includes the DDC2Bi and DDC2AB standards.

Note: The DDC2AB standard is no longer handled by the interface.

4.8.5.1 DDC1/2B Interface

4.8.5.1.1 Functionnal description

Refer to the DCR, AHR registers in Section 4.8.6. for the bit definitions.

The DDC1/2B Interface acts as an I/O interface between a DDC bus and the microcontroller memory. In addition to receiving and transmitting serial data, this interface directly transfers parallel data to and from memory using a DMA engine, only halting CPU activity for two clock cycles during each byte transfer.

The interface supports by hardware:

– Two DDC communication protocols called DDC1 and DDC2B.

– Write operations into RAM.

– Read operations from RAM.

In DDC1, the interface reads sequential EDID v1 data bytes from the microcontroller memory, and transmits them on SDA synchronized with Vsync.

In DDC2B mode, it operates in I^2C slave mode.

The DDC1/2B Interface supports several DDC versions configured using the CF[2:0] bits in the DCR register which can only be changed while the interface is disabled (HWPE bit=0 in the DCR register). They define which EDID structure version is used and which Device Addresses are recognized.

Depending on the DDC version, one or two device address pairs will be recognized and the corresponding EDID structure will be validated (refer to Table 25):

– **DDC v2 (CF2=0,CF1=0,CF0=0)**: DDC1 is enabled and device addresses A0h/A1h are recognized. EDID v1 is used.

– **DDC v2 (CF2=1,CF1=0,CF0=0)**: DDC1 is disabled and device addresses A0h/A1h are recognized. EDID v1 is used.

– **Plug and Display (CF2=0,CF1=0,CF0=1)**: DDC1 is disabled and device addresses A2h/A3h are recognized. EDID v2 is used.

– **Plug and Display + DDC v2 (CF2=0,CF1=1, CF0=0)**: DDC1 is enabled and device addresses A0h/A1h and A2h/A3h are recognized. Both EDID structures v1 and v2 are used.

– **Plug and Display + DDC v2 (CF2=1,CF1=1, CF0=0)**: DDC1 is disabled and device addresses A0h/A1h and A2h/A3h are recognized. Both EDID structures v1 and v2 are used.

– **FPDI (CF2=0,CF1=1, CF0=1)**: DDC1 is disabled and device addresses A6h/A7h are recognized. EDID v2 is used.

Table 25. Valid Device Addresses and EDID structure

Device Address	CF2 bit	CF1 bit	CF0 bit	Transfer Type
EDID v1: **A0h / A1h** = 1010 000x	x	x	0	128-byte EDID structure write/read
EDID v2: **A2h / A3h** = 1010 001x	0	0	1	256-byte EDID structure write/read
	0	1	0	
	1	1	0	
EDID v2: **A6h / A7h** = 1010 011x	1	1	1	256-byte EDID structure write/read
reserved	1	x	1	reserved

DDC INTERFACE (Cont'd)

The Write and Read operations allow the EDID data to be downloaded during factory alignment (for example).

Writes to the memory by the DMA engine can be inhibited by means of the WP bit in the DCR register.

A write of the last data structure byte sets a flag and may be programmed to generate an interrupt request.

The Data address (sub-address) is either the second byte of write transfers or is pointed to by the internal address counter automatically incremented after each byte transfer.

Physical address mapping of the data structure within the memory space is performed with a dedicated register accessible by software.

4.8.5.1.2 Mode description

DDC1 Mode: This mode is only enabled when the DDC v2 or P&D-DDC v2 standards are validated. It transmits only the EDID v1 data (128 bytes).

To switch the DDC1/2B Interface to DDC1 mode, software must first clear the CF0 bit in the DCR register while the HWPE bit=0 and then set the HWPE bit to enable the DDC1/2B Interface.

A proper initialization sequence (see Figure 59) must supply nine clock pulses on the VSYNCI pin in order to internally synchronize the device. During this initialization sequence, the SDA pin is in high impedance. On the rising edge of the 10th pulse applied on VSYNCI, the device outputs on SDA the most significant (MSB) bit of the byte located at data address 00h.

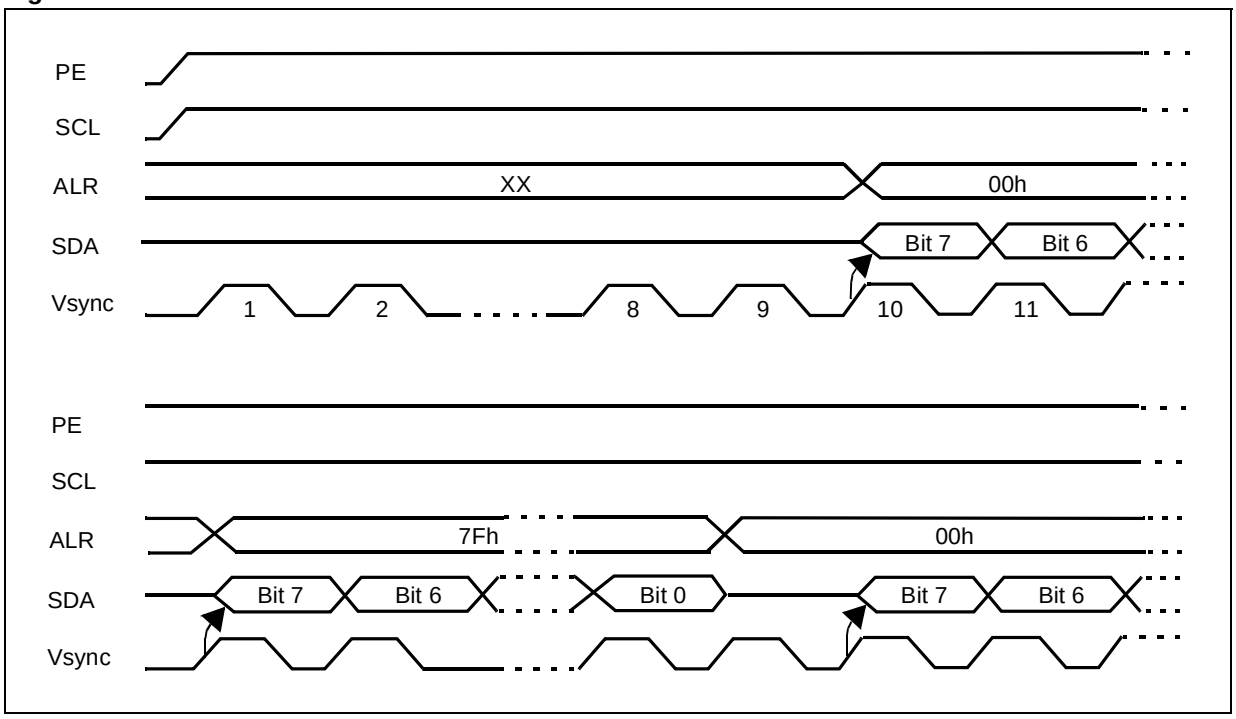
A byte is clocked out by means of 9 clock pulses on Vsync, 8 clock pulses for the data byte itself and an extra pulse for a Don't Care bit.

As long as SCL is not held low, each byte of the memory array is transmitted serially on SDA.

The internal address counter is incremented automatically until the last byte is transmitted. Then, it rolls over to relative location 00h.

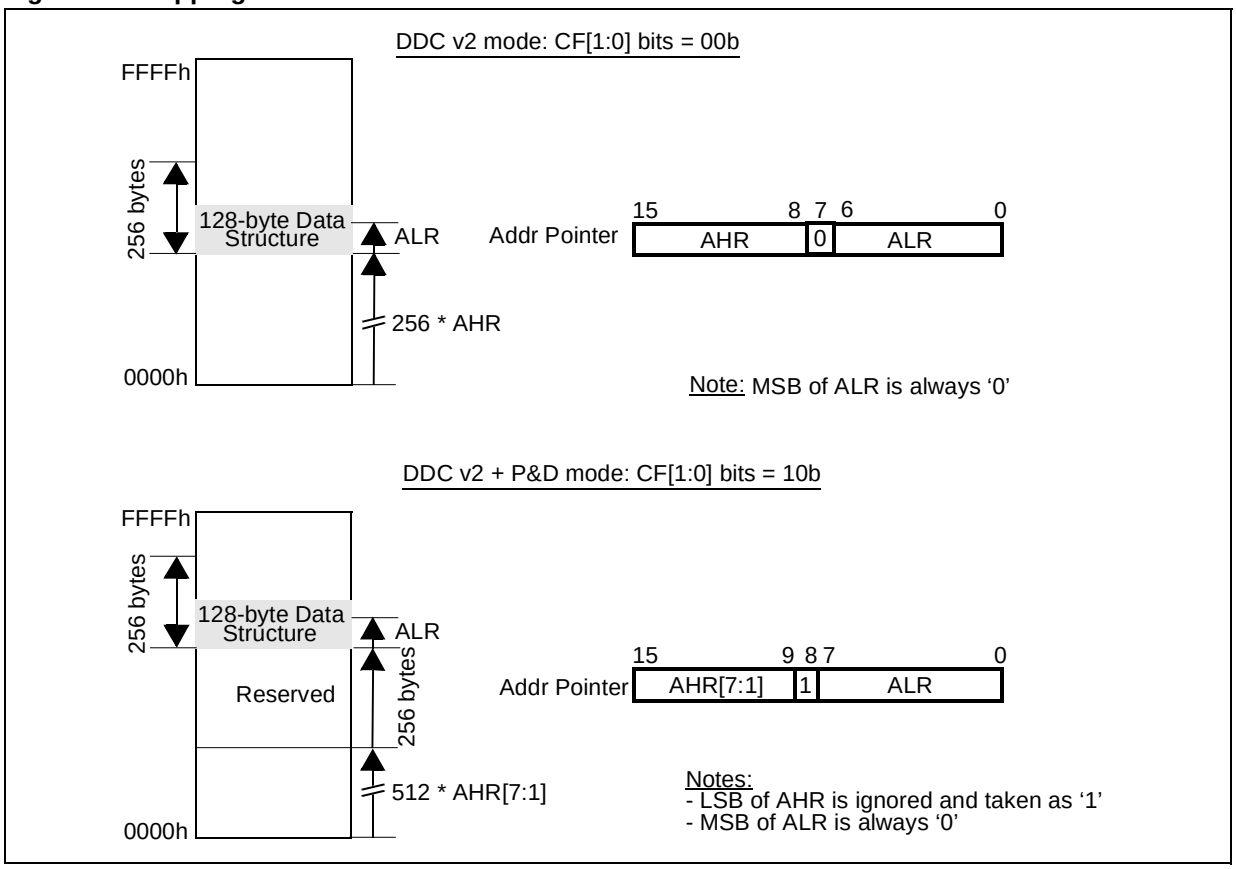
The physical mapping of the data structure depends on the configuration and on the content of the AHR register which can be set by software (see Figure 60).

Figure 59. DDC1 Waveforms



DDC INTERFACE (Cont'd)

Figure 60. Mapping of DDC1 data structure



DDC2B Transition Mode: This mode avoids the display switching to DDC2B mode if spurious noise is detected on SCL while the host is in DDC1 mode.

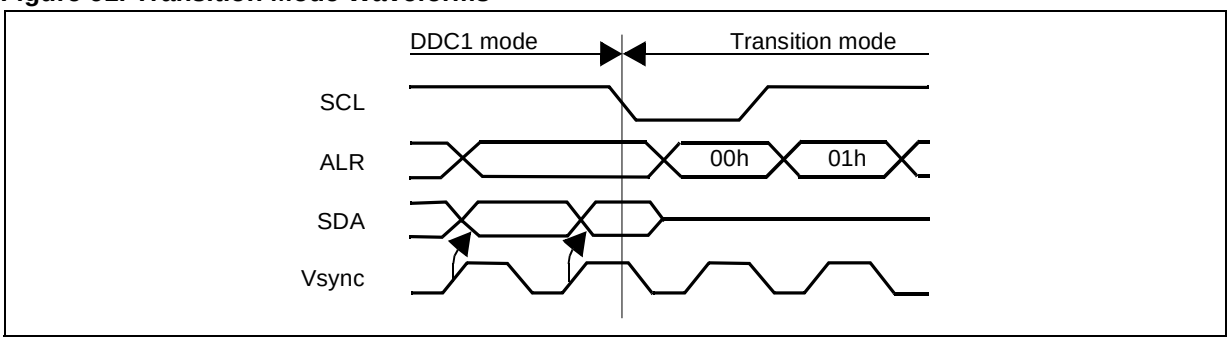
When the DDC1/2B interface is in DDC1 mode and detects a falling edge on SCL, it enters the transition state (see Figure 61).

If a valid I^2C sequence (START followed by a valid Device Address for CF0 =0 (see Table 25)) is not received within either 128 Vsync pulses or a period of approximately 2 seconds, then the interface will revert to DDC1 mode at the EDID start address.

If the interface decodes a valid DDC2B Device Address, it will lock into DDC2B mode and subsequently disregard VSYNCI.

When in transition mode, the Vsync pulse counter or the 2-sec. timeout counter is reset by any activity on the SCL line.

Figure 61. Transition Mode Waveforms



DDC INTERFACE (Cont'd)

DDC2B Mode: The DDC1/2B Interface enters DDC2B mode either from the transition state or from the initial state if software sets the HWPE bit while P&D only or FPDI-2 mode is selected.

Once in DDC2B mode, the Interface always acts as a slave following the protocol described in Figure 62.

The DDC1/2B Interface continuously monitors the SDA and SCL lines for a START condition and will not respond (no acknowledge) until one is found.

A STOP condition at the end of a Read command (after a NACK) forces the stand-by state. A STOP condition at the end of a Write command triggers the internal DMA write cycle.

The Interface samples the SDA line on the rising edge of SCL and outputs data on the falling edge of SCL. In any case SDA can only change when SCL is low.

Figure 62. DDC2B protocol (example)

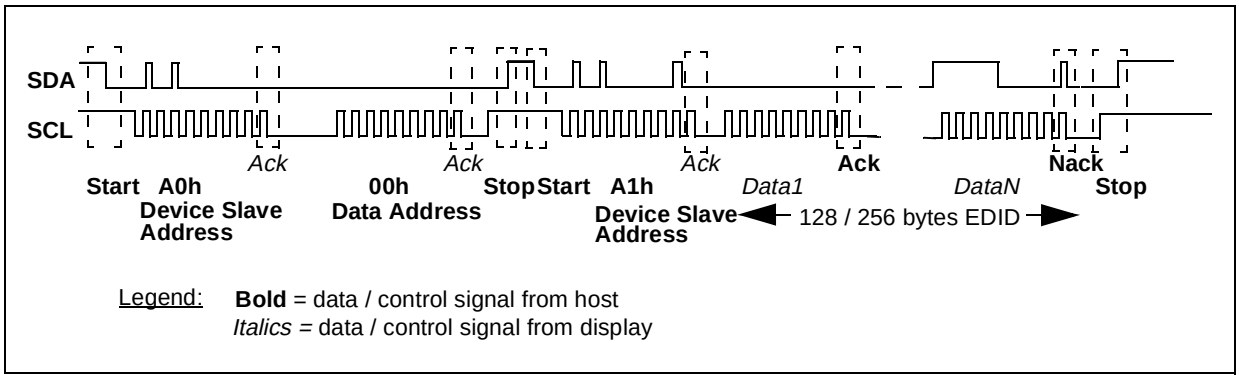
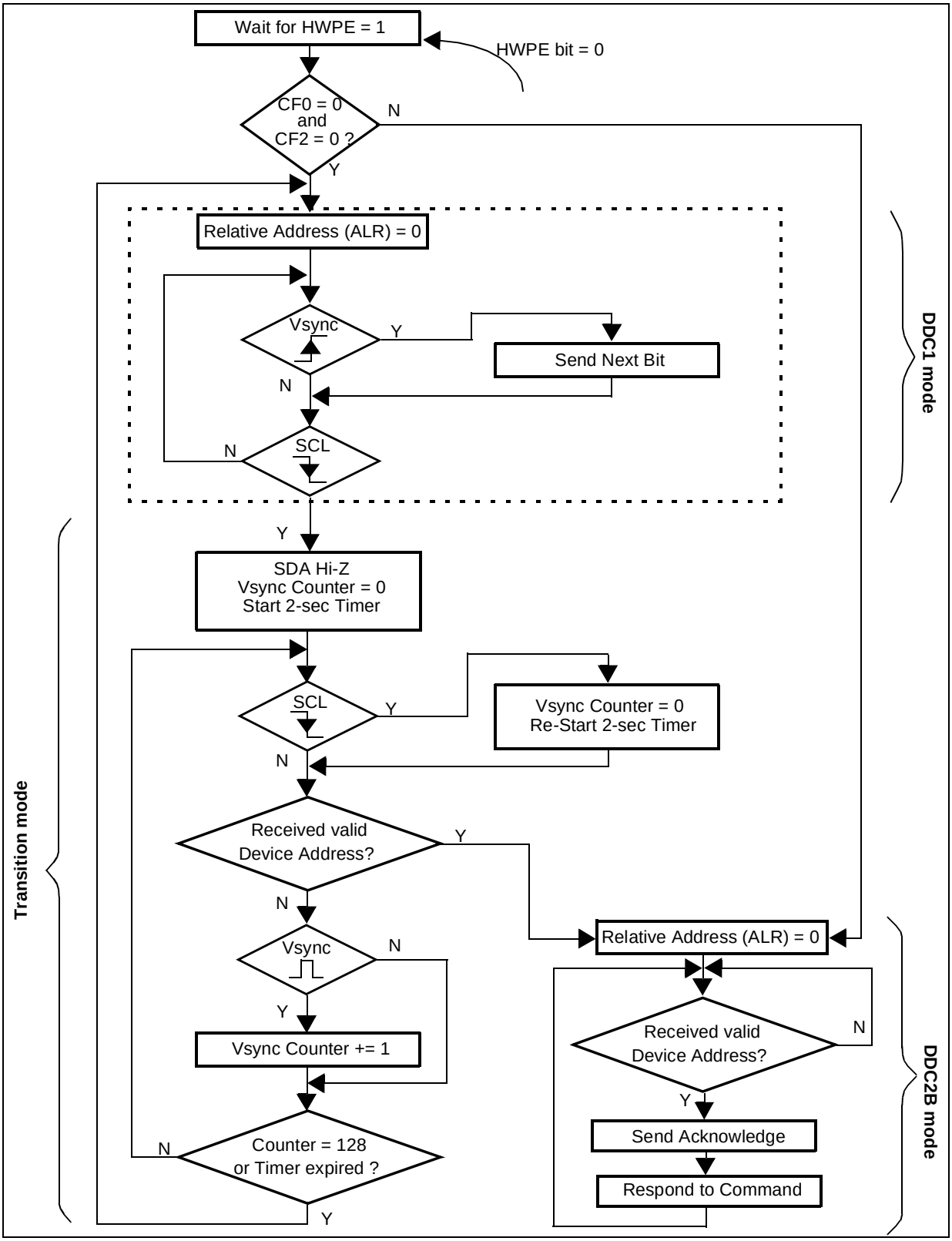


Figure 63. DDC1/2B Operation Flowchart



DDC INTERFACE (Cont'd)

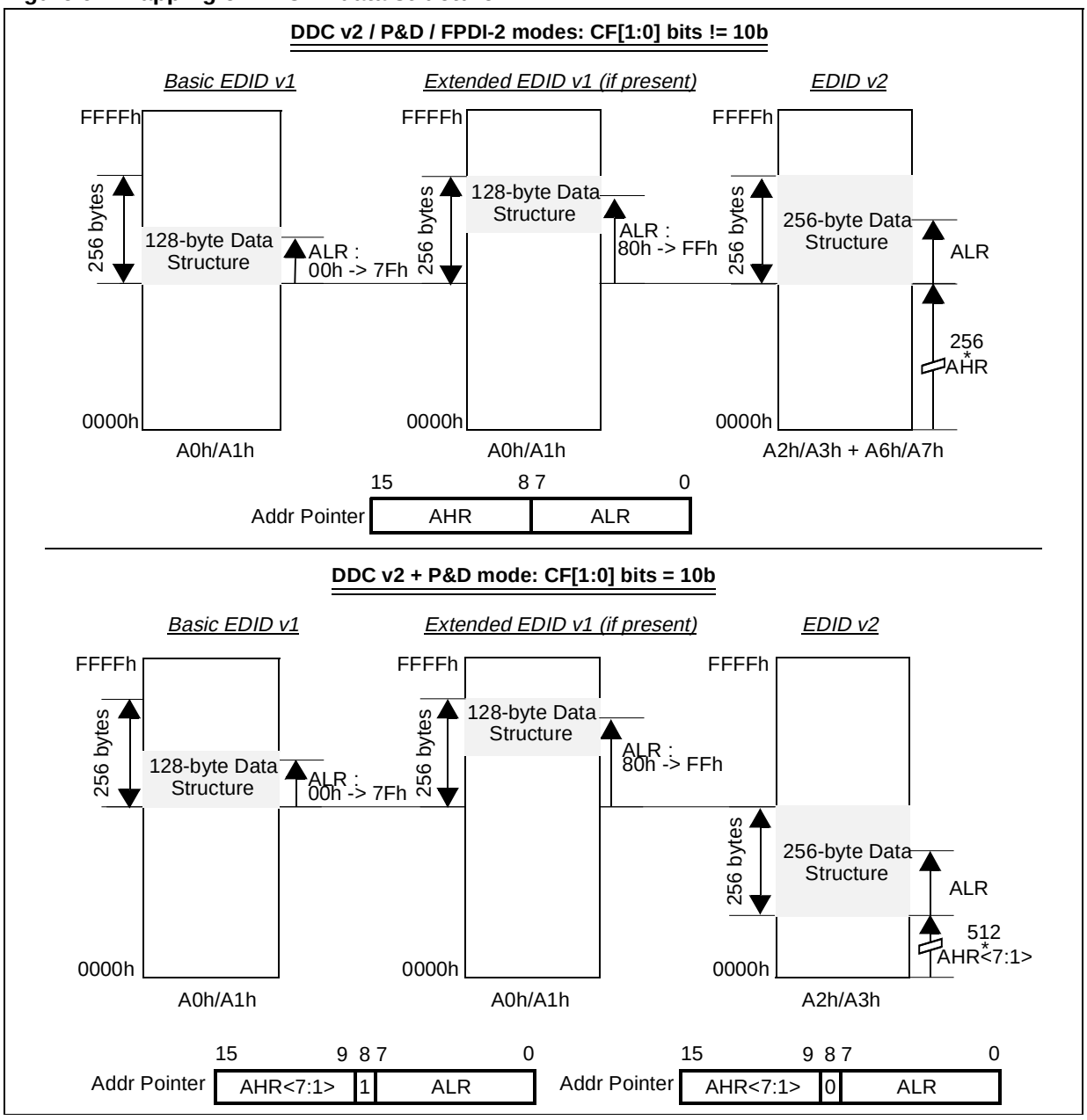
EDID Data structure mapping: An internal address pointer defines the memory location being addressed. It is made of two 8-bit registers AHR and ALR.

AHR is initialized by software. It defines the 256-byte block within the 64K address space containing the data structure.

ALR is loaded with the data address sent by the master after a write Device Address. It defines the byte within the data structure currently addressed. ALR is reset upon entry into the DDC2B mode.

One exception to this arrangement is when the CF[1:0] bits = 10b. In this case the two EDID versions must coexist at non-overlapping addresses. The LSB of AHR is therefore ignored and automatically set to 1 to address the 128-byte EDID and set to 0 to address its 256-byte counterpart (see Figure 64).

Figure 64. Mapping of DDC2B data structure



DDC INTERFACE (Cont'd)

■ **Write Operation**

Once the DDC1/2B Interface has acknowledged a write transfer request, i.e. a Device Address with RW=0, it waits for a data address. When the latter is received, it is acknowledged and loaded into the ALR.

Then, the master may send any number of data bytes that are all acknowledged by the DDC1/2B Interface. The data bytes are written in RAM if the WP bit=0 in the DCR register, otherwise the RAM location is not modified.

In any case, all write operations are performed in RAM and therefore do not delay DDC transfers, although concurrent software execution is halted for 2 cycles.

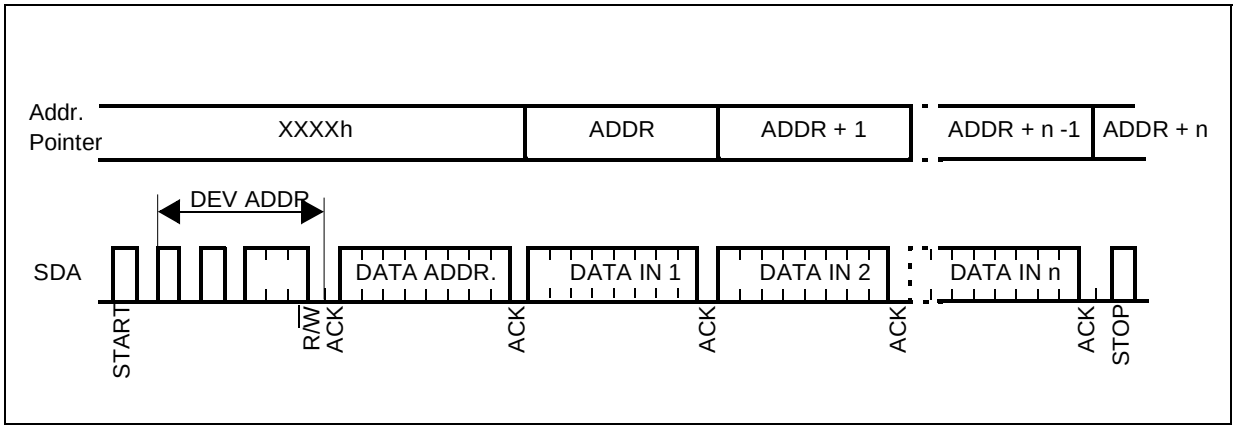
After each byte is transferred, the internal address counter is automatically incremented.

If the counter is pointing to the top of the structure, it rolls over to the bottom since the incrementation is performed only on the 7 or 8 LSB's of the pointer depending on the selected data structure size. In other words, ALR rolls over from FFh to 00h for Device Addresses A2h/A3h and A6h/A7h. Otherwise, it rolls over from 7Fh to 00h or from FFh to 80h depending on the MSB of the last data address received.

Then after that last byte has been effectively written in RAM, the EDF flag is set and an interrupt is generated if EDE is set.

The transfer is terminated by the master generating a STOP condition.

Figure 65. Write sequence



■ **Read Operations**

All read operations consist of retrieving the data pointed to by an internal address counter which is initialized by a dummy write and incremented by any read. The DDC1/2B Interface always waits for an acknowledge during the 9th bit-time. If the master does not pull the SDA line low during this bit-time, the DDC1/2B Interface ends the transfer and switches to a stand-by state.

– **Current address read:** After generating a START condition the master sends a read device address (RW = 1). The DDC1/2B Interface acknowledges this and outputs the data byte pointed to by the internal address pointer which is subsequently incremented. The master must NOT acknowledge this byte and must terminate the transfer with a STOP condition.

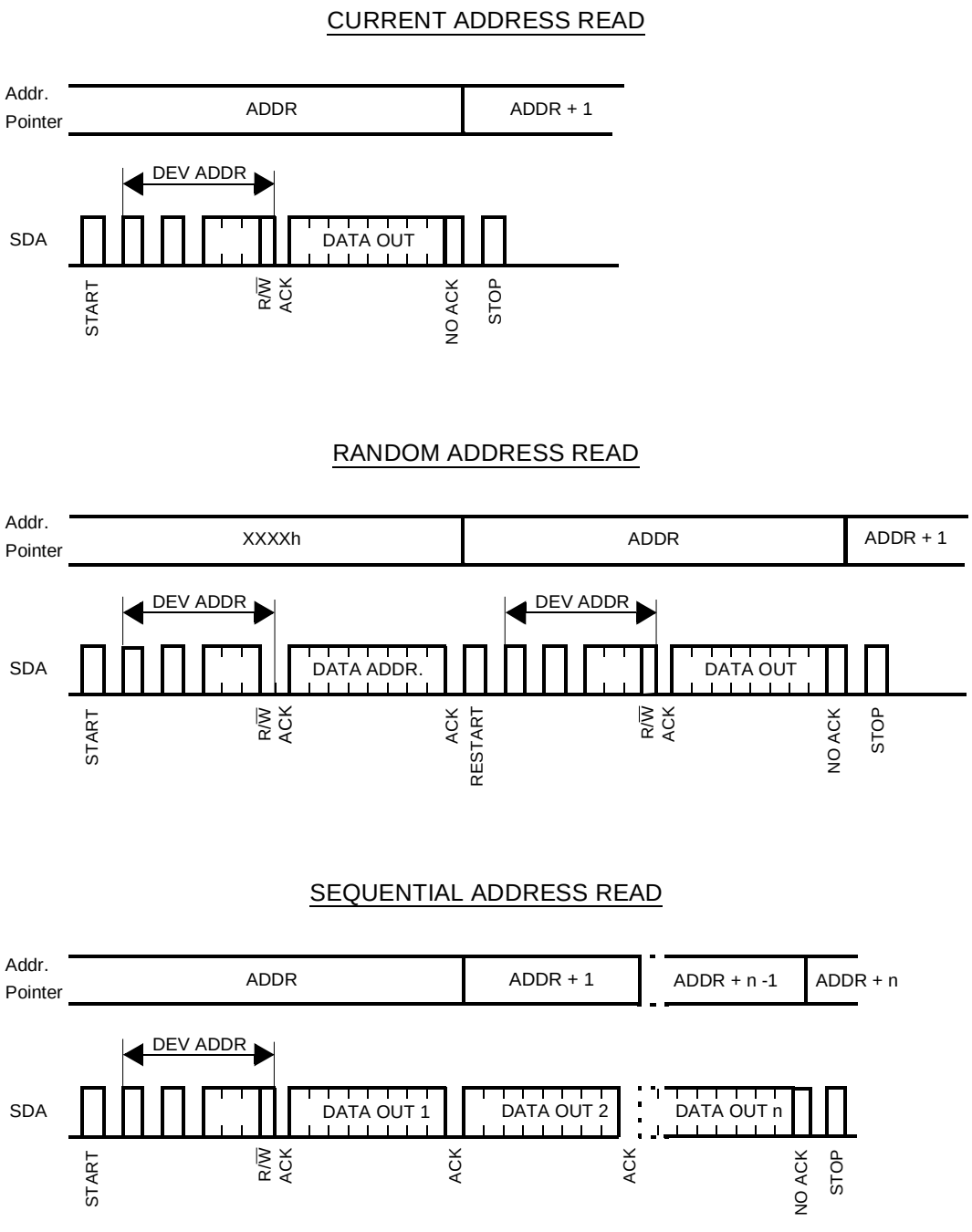
– **Random address read:** The master performs a dummy write to load the data address into the ALR. Then the master sends a RESTART condition followed by a read Device Address (RW=1).

– **Sequential address read:** This mode is similar to the current and random address reads, except that the master DOES acknowledge the data byte for the DDC1/2B Interface to output the next byte in sequence. To terminate the read operation the master must NOT acknowledge the last data byte and must generate a STOP condition.

The data output are from consecutive memory addresses. The internal address counter is incremented automatically after each byte. If the counter is pointing to the top of the structure, it rolls over to the bottom since the incrementation is performed only on the 7 or 8 LSB's of the counter depending on the selected data structure size.

DDC INTERFACE (Cont'd)

Figure 66. Read sequences



DDC INTERFACE (Cont'd)

4.8.5.2 DDC/CI - Factory Alignment Interface

4.8.5.2.1 Functional Description

Refer to the CR, SR1 and SR2 registers in Section 4.8.6. for the bit definitions.

The DDC/CI interface works as an I/O interface between the microcontroller and the DDC2Bi, EDDC or Factory alignment protocols. It receives and transmits data in Slave I^2C mode using an interrupt or polled handshaking.

The interface is connected to the I^2C bus by a data pin (SDAD) and a clock pin (SCLD) configured as open drain.

The DDC/CI interface has five internal register locations.

Two of them are used for initialization of the interface:

– Own Address Register OAR

– Control register CR

The following four registers are used during data transmission/reception:

– Data Register DR

– Control Register CR

– Status Register 1 SR1

– Status Register 2 SR2

The interface decodes an I^2C or DDC2Bi address stored by software in the OAR register and/or the EDDC address (60h/61h) as its default hardware address.

After a reset, the interface is disabled.

4.8.5.2.2 I^2C Modes

■ **General description**

In I^2C mode, the interface can operate in the following modes:

– Slave transmitter/receiver

Both start and stop conditions are generated by the master. The I2C clock (SCL) is always received by the interface from a master, but the interface is able to stretch the clock line.

The interface is capable of recognizing both its own programmable address (7-bit) and its default hardware address (Enhanced DDC address: 60h/61h). The Enhanced DDC address detection may be enabled or disabled by software. It never recognizes the Start byte (01h) whatever its own address is.

■ **Slave Mode**

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register; then it is compared with the programmable address of the interface or the Enhanced DDC address (if selected by software).

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the following events occur in sequence:

– Acknowledge pulse is generated if the ACK bit is set.

– EVF and ADSL bits are set.

– An interrupt is generated if the ITE bit is set.

Then the interface waits for a read of the SR1 register, **holding the SCL line low** (see Figure 67 Transfer sequencing EV1).

Next, the DR register must be read to determine from the least significant bit if the slave must enter Receiver or Transmitter mode.

DDC INTERFACE (Cont'd)

Slave Receiver

Following the address reception and after SR1 register has been read, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte, the following events occur in sequence:

– Acknowledge pulse is generated if the ACK bit is set.

– EVF and BTF bits are set.

– An interrupt is generated if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see Figure 67 Transfer sequencing EV2).

Slave Transmitter

Following the address reception and after SR1 register has been read, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 67 Transfer sequencing EV3).

When the acknowledge pulse is received:

– EVF and BTF bits are set.

– An interrupt is generated if the ITE bit is set.

Closing slave communication

After the last data byte is transferred, a Stop Condition is generated by the master. The interface detects this condition and in this case:

– EVF and STOPF bits are set.

– An interrupt is generated if the ITE bit is set.

Then the interface waits for a read of the SR2 register (see Figure 67 Transfer sequencing EV4).

Error Cases

– **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and the BERR bits are set and an interrupt is generated if the ITE bit is set.
If it is a Stop then the interface discards the data, releases the lines and waits for another Start condition.
If it is a Start then the interface discards the data and waits for the next slave address on the bus.

– **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set and an interrupt is generated if the ITE bit is set.

Note: In both cases, SCL line is not held low; however, SDA line can remain low due to possible «0» bits transmitted last. It is then necessary to release both lines by software.

How to release the SDA / SCL lines

Set and subsequently clear the STOP bit while BTF is set. The SDA/SCL lines are released after the transfer of the current byte.

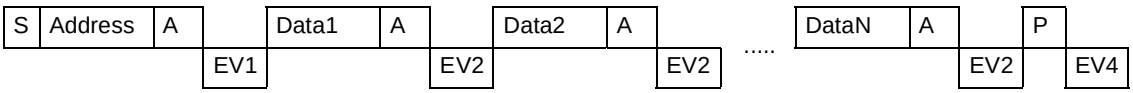
Other Events

- **ADSL**: Detection of a Start condition after an acknowledge time-slot.
The state machine is reset and starts a new process. The ADSL bit is set and an interrupt is generated if the ITE bit is set. The SCL line is stretched low.
- **STOPF**: Detection of a Stop condition after an acknowledge time-slot.
The state machine is reset. Then the STOPF flag is set and an interrupt is generated if the ITE bit is set.

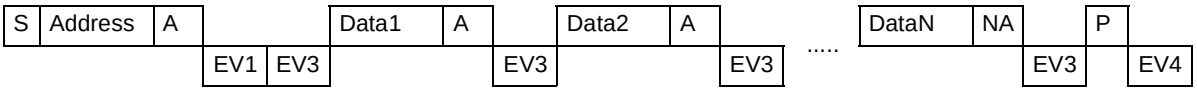
DDC INTERFACE (Cont'd)

Figure 67. Transfer Sequencing

Slave receiver:



Slave transmitter:



Legend:

S=Start, P=Stop, A=Acknowledge, NA=Non-acknowledge

EVx=Event (with interrupt if ITE=1)

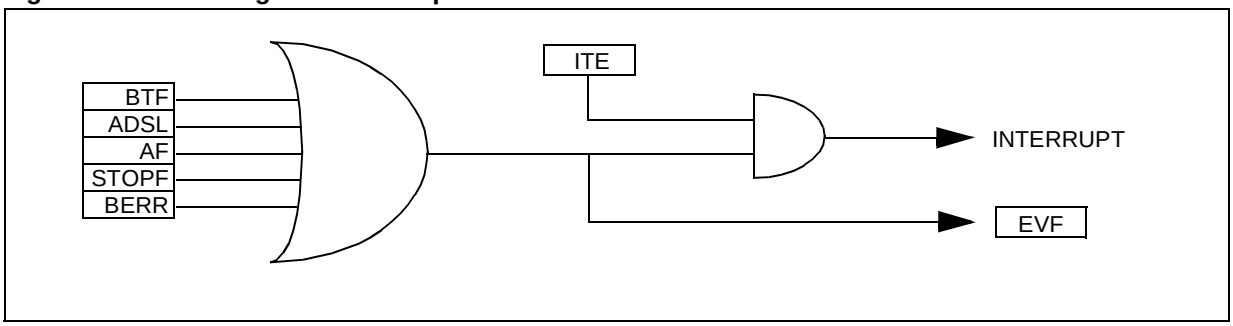
EV1: EVF=1, ADSL=1, cleared by reading SR1 register.

EV2: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV3: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

EV4: EVF=1, STOPF=1, cleared by reading SR2 register.

Figure 68. Event Flags and Interrupt Generation



DDC INTERFACE (Cont'd)

4.8.6 Register Description

DDC CONTROL REGISTER (CR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	PE	EDDC EN	0	ACK	STOP	ITE

Bit 7:6 = Reserved. Forced to 0 by hardware.

Bit 5 = **PE** *Peripheral enable.*

This bit is set and cleared by software.

0: Peripheral disabled
1: Slave capability

Notes:

– When PE=0, all the bits of the CR register and the SR register are reset. All outputs are released while PE=0
– When PE=1, the corresponding I/O pins are selected by hardware as alternate functions.
– To enable the I^2C interface, write the CR register **TWICE** with PE=1 as the first write only activates the interface (only PE is set).

Bit 4 = **EDDCEN** *Enhanced DDC address detection enabled.*

This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0). The 60h/61h Enhanced DDC address is acknowledged.

0: Enhanced DDC address detection disabled
1: Enhanced DDC address detection enabled

Bit 3 = Reserved. Forced to 0 by hardware.

Bit 2 = **ACK** *Acknowledge enable.*

This bit is set and cleared by software. It is also cleared by hardware when the interface is disabled (PE=0).

0: No acknowledge returned
1: Acknowledge returned after an address byte or a data byte is received

Bit 1 = **STOP** *Release I2C bus.*

This bit is set and cleared by software or when the interface is disabled (PE=0).

– Slave Mode:
  0: Nothing
  1: Release the SCL and SDA lines after the current byte transfer (BTF=1). The STOP bit has to be cleared by software.

Bit 0 = **ITE** *Interrupt enable.*

This bit is set and cleared by software and cleared by hardware when the interface is disabled (PE=0).

0: Interrupts disabled
1: Interrupts enabled

Refer to Figure 68 for the relationship between the events and the interrupt.
SCL is held low when the BTF or ADSL is detected.

DDC INTERFACE (Cont'd)
DDC STATUS REGISTER 1 (SR1)

Read Only

Reset Value: 0000 0000 (00h)

7							0
EVF	0	TRA	BUSY	BTF	ADSL	0	0

Bit 7 = **EVF** *Event flag.*
This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in Figure 67. It is also cleared by hardware when the interface is disabled (PE=0).

0: No event
1: One of the following events has occurred:
– BTF=1 (Byte received or transmitted)
– ADSL=1 (Address matched in Slave mode while ACK=1)
– AF=1 (No acknowledge received after byte transmission if ACK=1)
– STOPF=1 (Stop condition detected in Slave mode)
– BERR=1 (Bus error, misplaced Start or Stop condition detected)

Bit 6 = Reserved. Forced to 0 by hardware.

Bit 5 = **TRA** *Transmitter/Receiver.*
When BTF is set, TRA=1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware after detection of Stop condition (STOPF=1) or when the interface is disabled (PE=0).

0: Data byte received (if BTF=1)
1: Data byte transmitted

Bit 4 = **BUSY** *Bus busy.*
This bit is set by hardware on detection of a Start condition and cleared by hardware on detection of a Stop condition. It indicates a communication in progress on the bus. This information is still updated when the interface is disabled (PE=0).

0: No communication on the bus
1: Communication ongoing on the bus

Bit 3 = **BTF** *Byte transfer finished.*
This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE=1. It is cleared by software reading SR1 register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE=0).

– Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. . BTF is cleared by reading SR1 register followed by writing the next byte in DR register.
– Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

0: Byte transfer not done
1: Byte transfer succeeded

Bit 2 = **ADSL** *Address matched (Slave mode).* This bit is set by hardware as soon as the received slave address matched with the OAR register content or the Enhanced DDC address is recognized. An interrupt is generated if ITE=1. It is cleared by software reading SR1 register or by hardware when the interface is disabled (PE=0).

The SCL line is held low while ADSL=1.

0: Address mismatched or not received
1: Received address matched

Bit 1:0 = Reserved. Forced to 0 by hardware.

DDC INTERFACE (Cont'd)

DDC STATUS REGISTER 2 (SR2)

Read Only

Reset Value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	0	BERR	EDDCF

Bit 7:5 = Reserved. Forced to 0 by hardware.

Bit 4 = **AF** *Acknowledge failure*.
This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while AF=1.

0: No acknowledge failure
1: Acknowledge failure

Bit 3 = **STOPF** *Stop detection.*
This bit is set by hardware when a Stop condition is detected on the bus after an acknowledge (if ACK=1). An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while STOPF=1.

0: No Stop condition detected
1: Stop condition detected

Bit 2 = Reserved. Forced to 0 by hardware.

Bit 1 = **BERR** *Bus error.*
This bit is set by hardware when the interface detects a misplaced Start or Stop condition. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while BERR=1.

0: No misplaced Start or Stop condition
1: Misplaced Start or Stop condition

Bit 0 = **EDDCF** *Enhanced DDC address detected.*
This bit is set by hardware when the Enhanced DDC address (60h/61h) is detected on the bus while EDDCEN=1. It is cleared by hardware when a Start or a Stop condition (STOPF=1) is detected, or when the interface is disabled (PE=0).

0: No Enhanced DDC address detected on bus
1: Enhanced DDC address detected on bus

DDC INTERFACE (Cont'd)

DDC DATA REGISTER (DR)

Read / Write
Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D7-D0** *8-bit Data Register.*
These bits contain the byte to be received or transmitted on the bus.

– Transmitter mode: Byte transmission start automatically when the software writes in the DR register.

– Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address.
Then, the next data bytes are received one by one after reading the DR register.

DDC OWN ADDRESS REGISTER (OAR)

Read / Write
Reset Value: 0000 0000 (00h)

7							0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

Bit 7:1 = **ADD7-ADD1** *Interface address.*
These bits define the I^2C bus programmable address of the interface. They are not cleared when the interface is disabled (PE=0).

Bit 0 = **ADD0**.
This bit is Don't Care, the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE=0).

Note: Address 01h is always ignored.

DDC INTERFACE (Cont'd)

DDC1/2B CONTROL REGISTER (DCR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	CF2	EDF	EDE	CF1	CF0	WP	HWPE

Bit 7 = Reserved.
Forced by hardware to 0.

Bit 5 = **EDF** *End of Download interrupt Flag*.
This bit is set by hardware and cleared by software.

0: Download not started or not completed yet.
1: Download completed. Last byte of data structure (relative address 7Fh or FFh) has been stored in RAM.

Bit 4 = **EDE** *End of Download interrupt Enable*.
This bit is set and cleared by software.

0: Interrupt disabled.
1: A DDC1/2B interrupt is generated if EDF bit is set.

Bits 6, 3:2 = **CF[2:0]** *Configuration bits.*
These bits are set and cleared by software only when the peripheral is disabled (HWPE = 0). They define which EDID structure version is used and which Device Addresses are recognized as shown in the following table:

Bit 1 = **WP** *Write Protect*.
This bit is set and cleared by software.

0: Enable writes to the RAM.
1: Disable DMA write transfers and protect the RAM content. CPU writes to the RAM are not affected.

Bit 0= **HWPE** *Peripheral Enable*.
This bit is set and cleared by software.

0: Release the SDA port pin and ignore Vsync and SCL port pins. The other bits of the DCR and the content of the AHR are left unchanged.
1: Enable the DDC Interface and respond to the DDC1/DDC2B protocol.

ADDRESS POINTER HIGH REGISTER (AHR)

Read / Write

Reset Value: see Register Map

7							0
MSB							LSB

AHR contains the 8 MSB's of the 16-bit address pointer. It therefore defines the location of the 256-byte block containing the data structure within the CPU address space.

Note: AHR0 is ignored when CF[1:0] = 10 (P&D+ v2 mode) to allow non-overlapping 128-byte and 256-byte data structures.

CF[2:0] Bit Values	EDID version used	DDC1 Mode support / Transition Mode support	DDC2B Addresses Recognized
000	DDC v2	Yes (128b EDID) / Yes	128b-EDID @A0h/A1h
001	P&D	No	256b-EDID @ A2h/A3h
010	v2 + P&D	Yes (128b EDID) / Yes	128b-EDID @A0h/A1h
			256b-EDID @ A2h/A3h
011	FPDI-2	No	256b-EDID @ A6h/A7h
100	DDC v2	No	128b-EDID @A0h/A1h
101	Reserved		
110	v2 + P&D	No	128b-EDID @A0h/A1h
			256b-EDID @ A2h/A3h
111	Reserved		

DDC INTERFACE (Cont'd)

Table 26. DDC Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
50	**CR**			PE	EDDCEN		ACK	STOP	ITE
	Reset Value	0	0	0	0	0	0	0	0
51	**SR1**	EVF		TRA	BUSY	BTF	ADSL		
	Reset Value	0	0	0	0	0	0	0	0
52	**SR2**				AF	STOPF		BERR	EDDCF
	Reset Value	0	0	0	0	0	0	0	0
54	**OAR**	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	Reset Value	0	0	0	0	0	0	0	0
56	**DR**	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	0	0	0	0	0	0	0	0
0C	**DCR**		CF2	EDF	EDE	CF1	CF0	WP	HWPE
	Reset Value	0	0	0	0	0	0	0	0
0D	**AHR**	AHR7	AHR6	AHR5	AHR4	AHR3	AHR2	AHR1	AHR0
	Reset Value	0	0	0	0	0	0	0	0

4.9 PWM/BRM GENERATOR (DAC)

4.9.1 Introduction

This PWM/BRM peripheral includes two types of PWM/BRM outputs, with differing step resolutions based on the Pulse Width Modulator (PWM) and Binary Rate Multiplier (BRM) Generator technique are available. It allows the digital to analog conversion (DAC) when used with external filtering.

4.9.2 Main Features

- Fixed frequency: $f_{CPU}/64$
- Resolution: T_{CPU}
- 10-Bit PWM/BRM generator with a step of $V_{DD}/2^{10}$ (5mV if V_{DD}=5V)

4.9.3 Functional Description

4.9.3.1 PWM/BRM

The 10 bits of the 10-bit PWM/BRM are distributed as 6 PWM bits and 4 BRM bits. The generator consists of a 12-bit counter (common for all channels), a comparator and the PWM/BRM generation logic.

PWM Generation

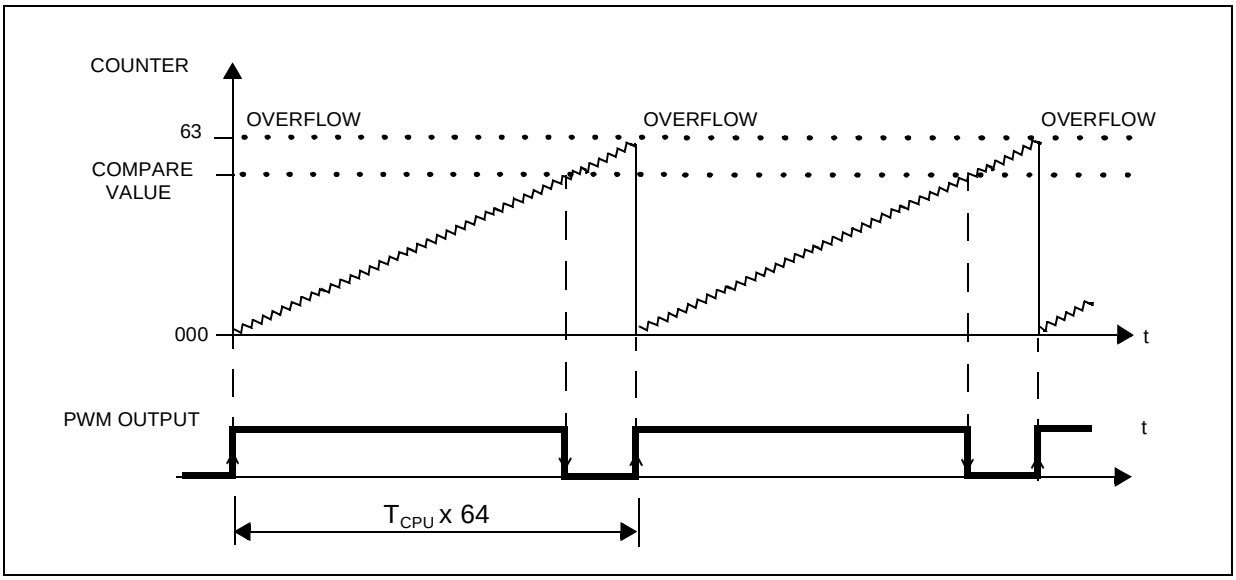
The counter increments continuously, clocked at internal CPU clock. Whenever the 6 least significant bits of the counter (defined as the PWM counter) overflow, the output level for all active channels is set.

The state of the PWM counter is continuously compared to the PWM binary weight for each channel, as defined in the relevant PWM register, and when a match occurs the output level for that channel is reset.

This Pulse Width modulated signal must be filtered, using an external RC network placed as close as possible to the associated pin. This provides an analog voltage proportional to the average charge passed to the external capacitor. Thus for a higher mark/space ratio (High time much greater than Low time) the average output voltage is higher. The external components of the RC network should be selected for the filtering level required for control of the system variable.

Each output may individually have its polarity inverted by software, and can also be used as a logical output.

Figure 69. PWM Generation



PWM/BRM Outputs (Cont'd)

PWM/BRM Outputs

The PWM/BRM outputs are assigned to dedicated pins.

The RC filter time must be higher than TCPUx64.

Figure 70. **Typical PWM Output Filter**

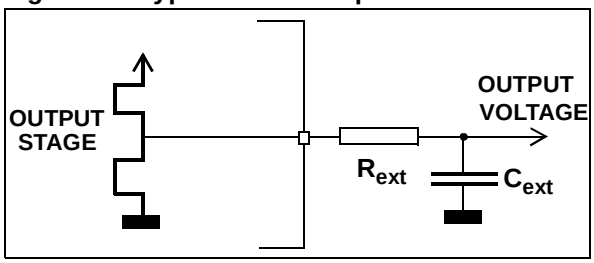


Table 27. **6-Bit PWM Ripple After Filtering**

C_{ext} (µF)	V RIPPLE (mV)
0.128	78
1.28	7.8
12.8	0.78

With RC filter (R=1KΩ),

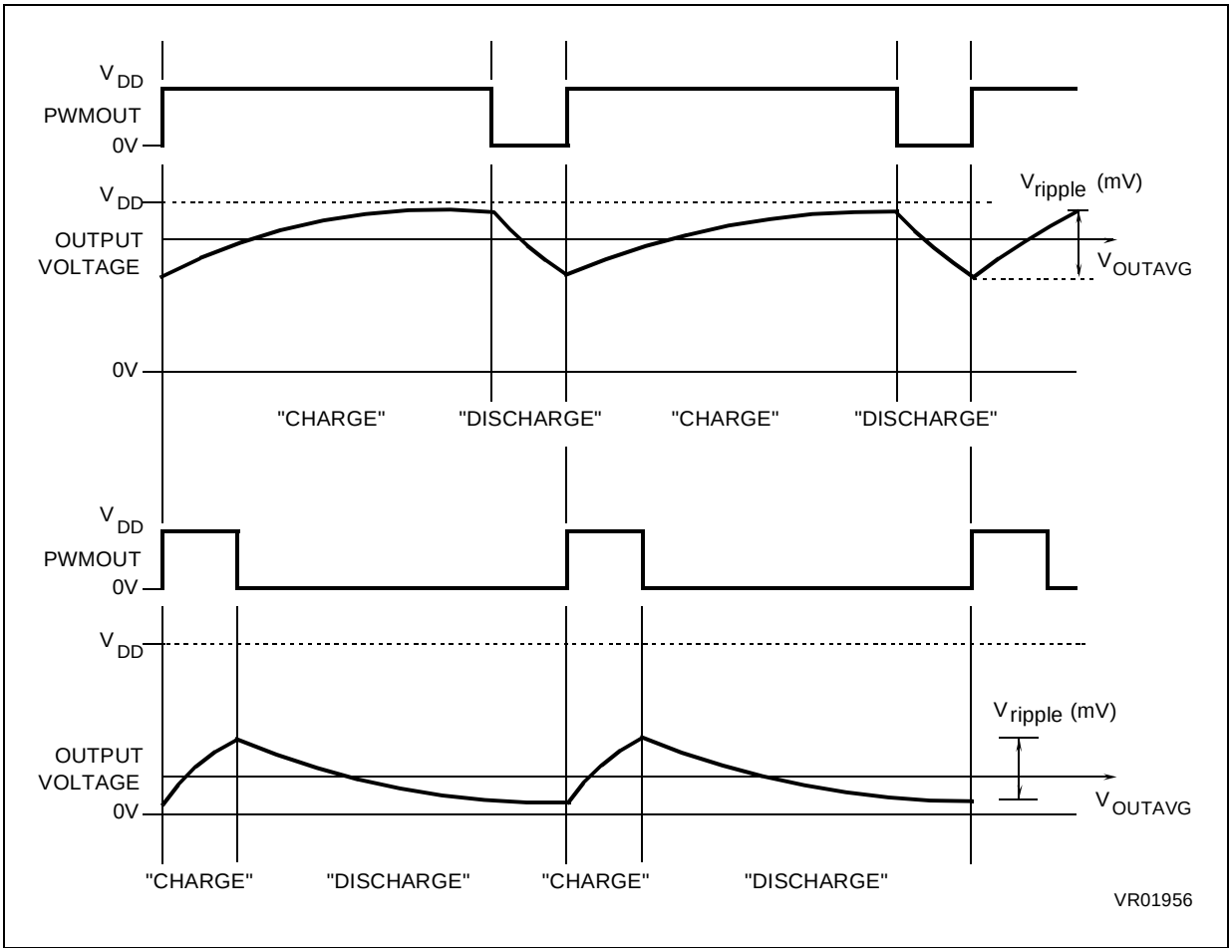
f_{CPU} = 8 MHz

V_{DD} = 5V

PWM Duty Cycle 50%

R=Rext.

Figure 71. PWM Simplified Voltage Output After Filtering



PWM/BRM GENERATOR (Cont'd)
BRM Generation

The BRM bits allow the addition of a pulse to widen a standard PWM pulse for specific PWM cycles. This has the effect of "fine-tuning" the PWM Duty cycle (without modifying the base duty cycle), thus, with the external filtering, providing additional fine voltage steps.

The incremental pulses (with duration of T_{CPU}) are added to the beginning of the original PWM pulse. The PWM intervals which are added to are specified in the 4-bit BRM register and are encoded as shown in the following table. The BRM values shown may be combined together to provide a summation of the incremental pulse intervals specified.

The pulse increment corresponds to the PWM resolution.

For example,if

– Data 18h is written to the PWM register
– Data 06h (00000110b) is written to the BRM register
– with a 8MHz internal clock (125ns resolution)

Then 3.0 µs-long pulse will be output at 8 µs intervals, except for cycles numbered 2,4,6,10,12,14, where the pulse is broadened to 3.125 µs.

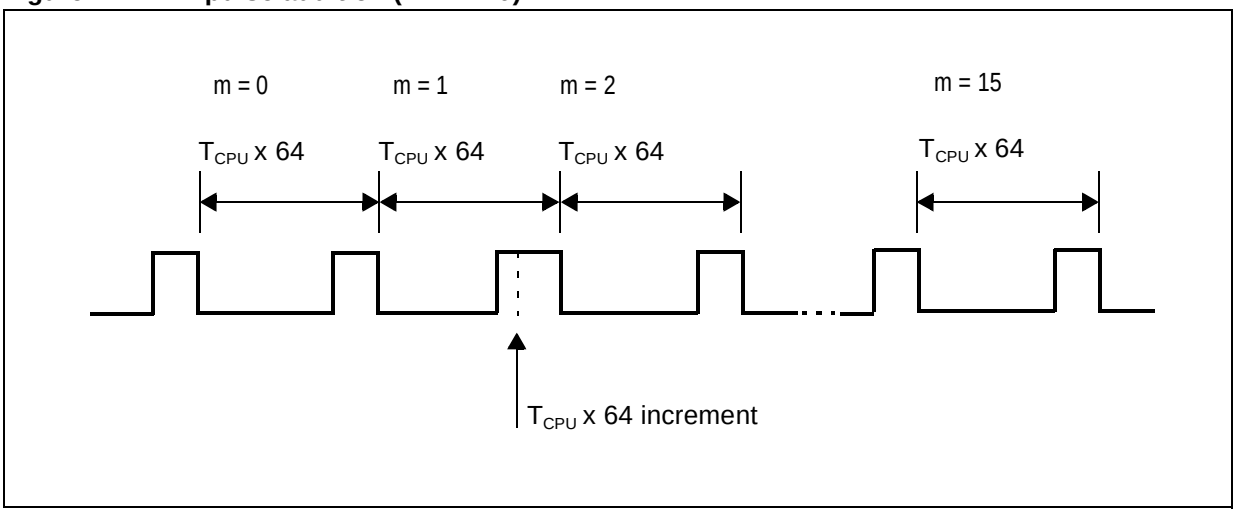
Note: If 00h is written to both PWM and BRM registers, the generator output will remain at "0". Conversely, if both registers hold data 3Fh and 0Fh, respectively, the output will remain at "1" for all intervals #1 to #15, but it will return to zero at interval #0 for an amount of time corresponding to the PWM resolution (T_{CPU}).

An output can be set to a continuous "1" level by clearing the PWM and BRM values and setting POL = "1" (inverted polarity) in the PWM register. This allows a PWM/BRM channel to be used as an additional I/O pin if the DAC function is not required.

Table 28. Bit BRM Added Pulse Intervals (Interval #0 not selected).

BRM 4 - Bit Data	Incremental Pulse Intervals
0000	none
0001	i = 8
0010	i = 4,12
0100	i = 2,6,10,14
1000	i = 1,3,5,7,9,11,13,15

Figure 72. BRM pulse addition (PWM > 0)



PWM/BRM GENERATOR (Cont'd)

Figure 73. Simplified Filtered Voltage Output Schematic with BRM added

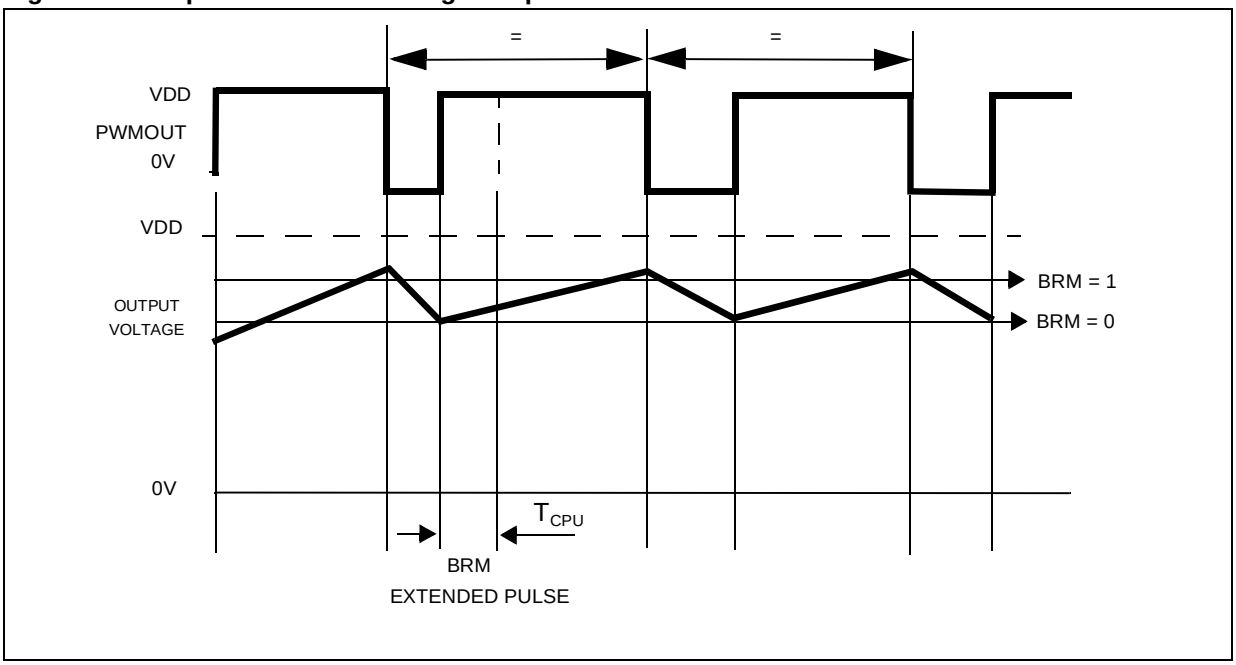
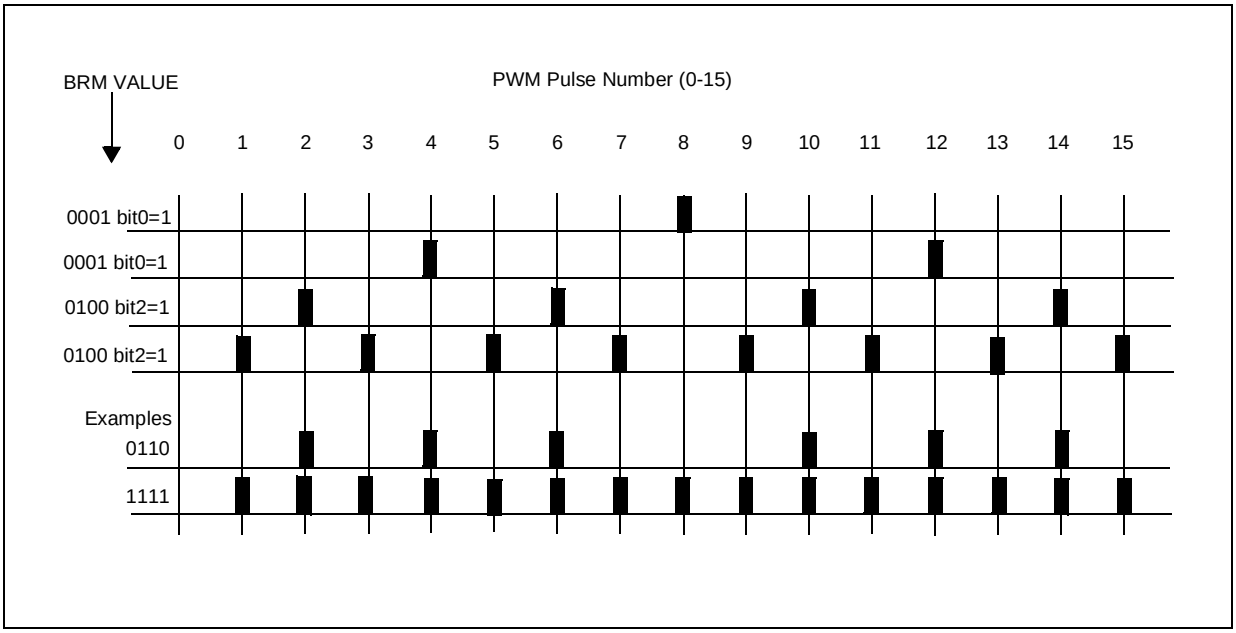


Figure 74. Graphical Representation of 4-Bit BRM Added Pulse Positions



PWM/BRM GENERATOR (Cont'd)

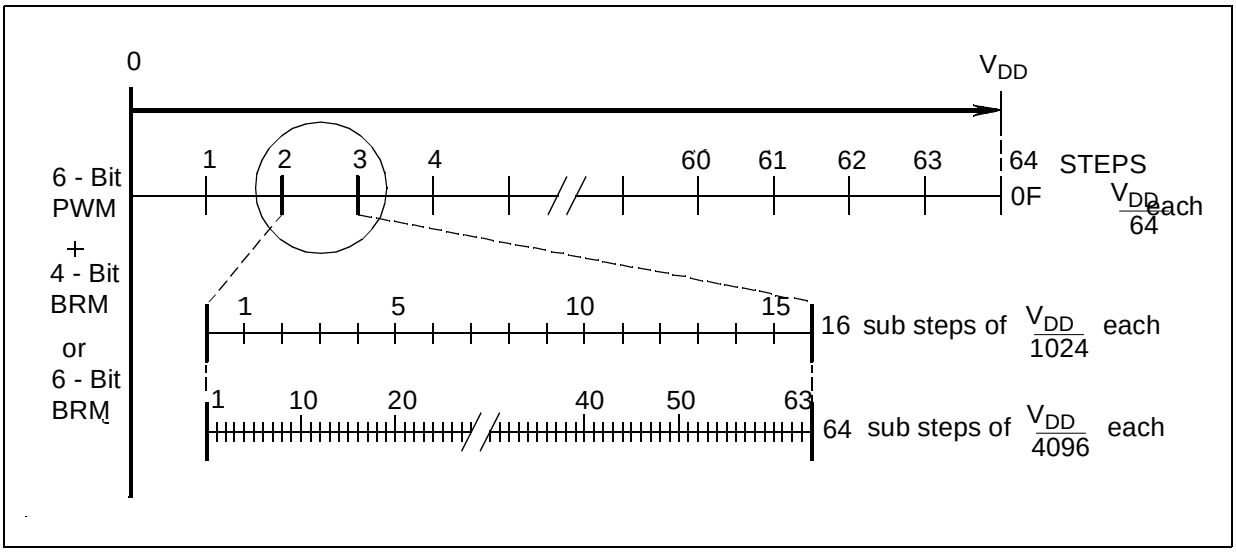
4.9.3.2 PWM/BRM OUTPUTS

The PWM/BRM outputs are assigned to dedicated pins.

If necessary, these pins can be used in push-pull or open-drain modes under software control.

In these pins, the PWM/BRM outputs are connected to a serial resistor which must be taken into account to calculate the RC filter.

Figure 75. Precision for PWM/BRM Tuning for VOUTEFF (After filtering)



PWM/BRM GENERATOR (Cont'd)

4.9.4 Register Description

4.9.4.1 PWM/BRM REGISTERS

On a channel basis, the 10 bits are separated into two data registers:

– A 6-bit PWM register corresponding to the binary weight of the PWM pulse.

– A 4-bit BRM register defining the intervals where an incremental pulse is added to the beginning of the original PWM pulse. Two BRM channel values share the same register.

Note: The number of PWM and BRM channels available depends on the device. Refer to the device pin description and register map.

PWM[1:8] REGISTERS

Read/Write
Reset Value 1000 0000 (80h)

7							0
1	POL	P5	P4	P3	P2	P1	P0

Bit 7 = Reserved (read as "1")

Bit 6 = **POL** *Polarity Bit.*
When POL is set, output signal polarity is inverse; otherwise, no change occurs.

Bits 5:0 = **P[5:0]** PWM Pulse Binary Weight for channel i .

For example (10-bit)

0	POL	P	P	P	P	P	P	+	B	B	B	B

Effective (with external RC filtering) DAC value

0	POL	P	P	P	P	P	P	B	B	B	B

BRM REGISTERS

BRM21 (Channels 2 + 1)

BRM43 (Channels 4 + 3)

BRM65 (Channels 6 + 5)

BRM87 (Channels 8 + 7)

Read/Write
Reset Value: 0000 0000 (00h)

7							0
B7	B6	B5	B4	B3	B2	B1	B0

Bits 7:4 = **B[7:4]** *BRM Bits (channel i+1)*

Bits 3:0 = **B[3:0]** *BRM Bits (channel i)*

4.9.4.2 OUTPUT ENABLE REGISTER

Read/Write
Reset Value 1000 0000 (80h)

7							0
OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0

Bit 7:0 = **OE[7:0]** *Output Enable Bit.*
When OEi is set, PWM output function is enabled.

0: PWM output is disabled
1: PWM output is enabled

Note: From the programmer's point of view, the PWM and BRM registers can be regarded as being combined to give one data value.

Table 29. PWM Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
32	**PWM1**		POL	P5 ..P0					
33	**BRM21**	BRM Channel 2				BRM Channel 1			
34	**PWM2**		POL	P5 ..P0					
35	**PWM3**		POL	P5 ..P0					
36	**BRM43**	BRM Channel 4				BRM Channel 3			
37	**PWM4**		POL	P5 ..P0					
38	**PWM5**		POL	P5 ..P0					
39	**BRM65**	BRM Channel 6				BRM Channel 5			
3A	**PWM6**		POL	P5 ..P0					
3B	**PWM7**		POL	P5 ..P0					
3C	**BRM87**	BRM Channel 8				BRM Channel 7			
3D	**PWM8**		POL	P5 ..P0					
3E	**PWMCR**	OE7 ..OE0							

4.10 8-BIT A/D CONVERTER (ADC)

4.10.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 8-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

The result of the conversion is stored in a 8-bit Data Register. The A/D converter is controlled through a Control/Status Register.

4.10.2 Main Features

- 8-bit conversion
- Up to 16 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in Figure 76.

4.10.3 Functional Description

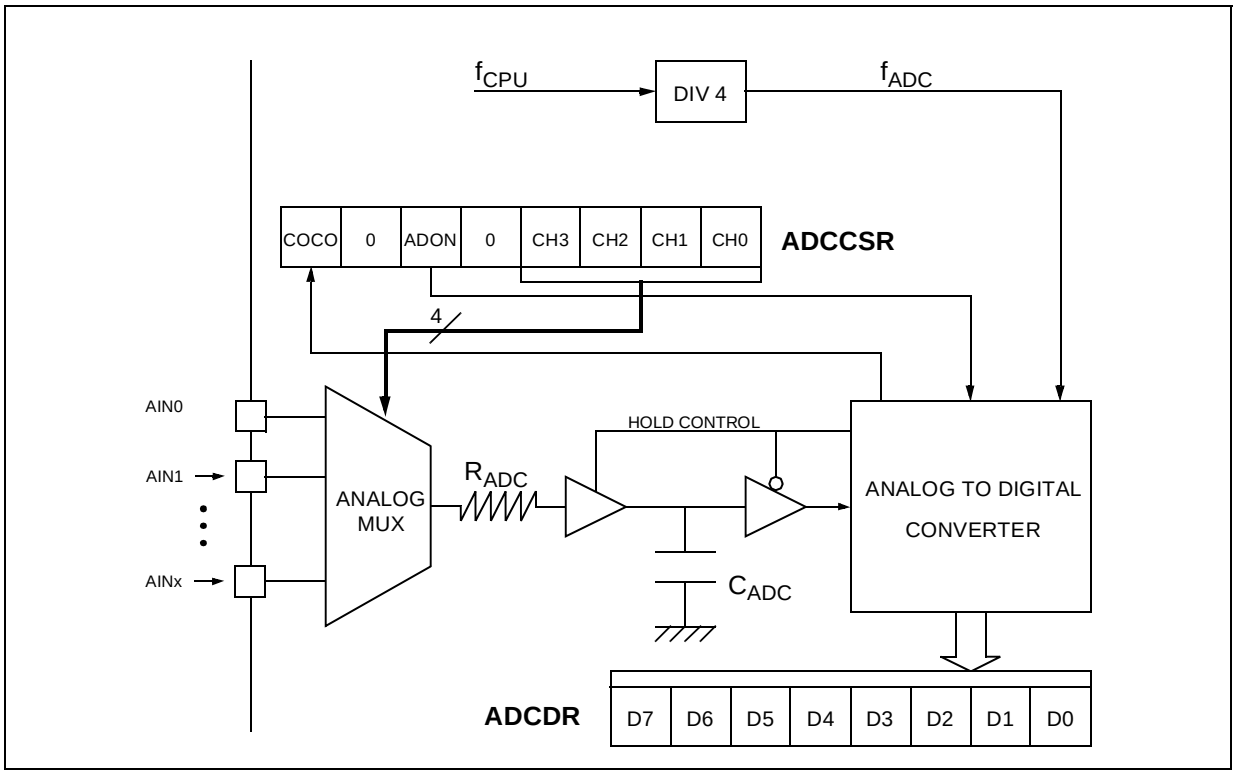
4.10.3.1 Analog Power Supply

V_{DDA} and V_{SSA} are the high and low level reference voltage pins. In some devices (refer to device pin out description) they are internally connected to the V_{DD} and V_{SS} pins.

Conversion accuracy may therefore be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

See electrical characteristics section for more details.

Figure 76. ADC Block Diagram



4.10.3.2 Digital A/D Conversion Result

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than or equal to V_{DDA} (high-level voltage reference) then the conversion result in the DR register is FFh (full scale) without overflow indication.

If input voltage (V_{AIN}) is lower than or equal to V_{SSA} (low-level voltage reference) then the conversion result in the DR register is 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDR register. The accuracy of the conversion is described in the parametric section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

4.10.3.3 A/D Conversion Phases

The A/D conversion is based on two conversion phases as shown in Figure 77:

- Sample capacitor loading [duration: t_{LOAD}]
  During this phase, the V_{AIN} input voltage to be measured is loaded into the C_{ADC} sample capacitor.
- A/D conversion [duration: t_{CONV}]
  During this phase, the A/D conversion is computed (8 successive approximations cycles) and the C_{ADC} sample capacitor is disconnected from the analog input pin to get the optimum analog to digital conversion accuracy.

While the ADC is on, these two phases are continuously repeated.

At the end of each conversion, the sample capacitor is kept loaded with the previous measurement load. The advantage of this behaviour is that it minimizes the current consumption on the analog pin in case of single input channel measurement.

4.10.3.4 Software Procedure

Refer to the control/status register (CSR) and data register (DR) in Section 4.10.6 for the bit definitions and to Figure 77 for the timings.

ADC Configuration

The total duration of the A/D conversion is 12 ADC clock periods ($1/f_{ADC}=4/f_{CPU}$).

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the «I/O ports» chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the CSR register:

– Select the CH[3:0] bits to assign the analog channel to be converted.

ADC Conversion

In the CSR register:

– Set the ADON bit to enable the A/D converter and to start the first conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete

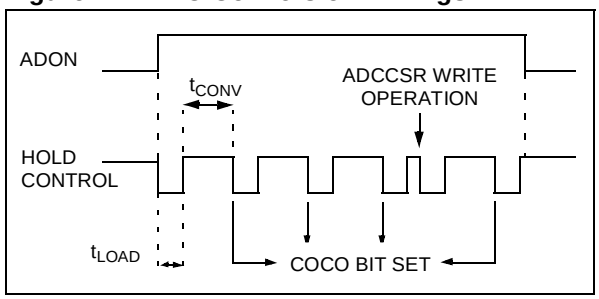
– The COCO bit is set by hardware.

– No interrupt is generated.

– The result is in the DR register and remains valid until the next conversion has ended.

A write to the CSR register (with ADON set) aborts the current conversion, resets the COCO bit and starts a new conversion.

Figure 77. ADC Conversion Timings



4.10.4 Low Power Mode

Mode	Description
WAIT	No effect on A/D Converter

Note: The A/D converter is disabled by resetting the ADON bit. With this feature, power consumption is reduced when no conversion is needed and between single shot conversions.

4.10.5 Interrupts

None

8-BIT A/D CONVERTER (ADC) (Cont'd)

4.10.6 Register Description

CONTROL/STATUS REGISTER (CSR)
Read/Write

Reset Value: 0000 0000 (00h)

7							0
COCO	0	ADON	0	CH3	CH2	CH1	CH0

Bit 7 = **COCO** *Conversion Complete*
This bit is set by hardware. It is cleared by software reading the result in the DR register or writing to the CSR register.

0: Conversion is not complete
1: Conversion can be read from the DR register

Bit 6 = **Reserved.** *must always be cleared.*

Bit 5 = **ADON** *A/D Converter On*
This bit is set and cleared by software.

0: A/D converter is switched off
1: A/D converter is switched on

Bit 4 = **Reserved.** *must always be cleared.*

Bit 3:0 = **CH[3:0]** *Channel Selection*
These bits are set and cleared by software. They select the analog input to convert.

Note: The number of pins AND the channel selection varies according to the device. Refer to the device pinout.

Channel Pin*	CH3	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
AIN6	0	1	1	0
AIN7	0	1	1	1
AIN8	1	0	0	0
AIN9	1	0	0	1
AIN10	1	0	1	0
AIN11	1	0	1	1
AIN12	1	1	0	0
AIN13	1	1	0	1
AIN14	1	1	1	0
AIN15	1	1	1	1

DATA REGISTER (DR)
Read Only

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[7:0]** *Analog Converted Value*
This register contains the converted analog value in the range 00h to FFh.

Note: Reading this register reset the COCO flag.

Table 30. ADC Register Map

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0A	**ADCDR**	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	0	0	0	0	0	0	0	0
0B	**ADCCSR**	COCO	-	ADON	-	CH3	CH2	CH1	CH0
	Reset Value	0	0	0	0	0	0	0	0

5 INSTRUCTION SET

5.1 ST7 ADDRESSING MODES

The ST7 Core features 17 different addressing modes which can be classified in 7 main groups:

Addressing Mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 Instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two sub-modes called long and short:

– Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.

– Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 31. ST7 Addressing Mode Overview

Mode			Syntax	Destination/ Source	Pointer Address (Hex.)	Pointer Size (Hex.)	Length (Bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00..FF			+ 0 (with X register) + 1 (with Y register)
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC-128/PC+127[1)]			+ 1
Relative	Indirect		jrne [$10]	PC-128/PC+127[1)]	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

Note 1. At the time the instruction is executed, the Program Counter (PC) points to the instruction following JRxx.

ST7 ADDRESSING MODES (Cont'd)

5.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Inherent Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	**Disabled, forces a RESET**
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask
RIM	Reset Interrupt Mask
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

5.1.2 Immediate

Immediate instructions have two bytes, the first byte contains the opcode, the second byte contains the operand value.

Immediate Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

5.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two sub-modes:

Direct (short)
The address is a byte, thus requires only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)
The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

5.1.4 Indexed (No Offset, Short, Long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three sub-modes:

Indexed (No Offset)
There is no offset, (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed (Short)
The offset is a byte, thus requires only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long)
The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

5.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two sub-modes:

Indirect (short)
The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

5.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two sub-modes:

Indirect Indexed (Short)
The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)
The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 32. Instructions Supporting Direct, Indexed, Indirect and Indirect Indexed Addressing Modes

Long and Short Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Addition/subtraction operations
BCP	Bit Compare

Short Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

5.1.7 Relative mode (Direct, Indirect)

This addressing mode is used to modify the PC register value, by adding an 8-bit signed offset to it.

Available Relative Direct/ Indirect Instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two sub-modes:

Relative (Direct)
The offset follows the opcode.

Relative (Indirect)

The offset is defined in memory, of which the address follows the opcode.

5.2 INSTRUCTION GROUPS

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx							
Interruption management	TRAP	WFI	IRET					
Code Condition Flag modification	SIM	RIM	SCF	RCF				

Using a pre-byte

The instructions are described with one to four bytes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC opcode

PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A . M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg	reg	M					
		pop CC	CC	M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= Dst <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => Dst => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= Dst <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= Dst <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => Dst => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	Dst7 => Dst => C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	Dst[7..4] <=> Dst[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz lbl1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

6 ELECTRICAL CHARACTERISTICS

The ST727x4 device contains circuitry to protect the inputs against damage due to high static voltage or electric field. Nevertheless it is advised to take normal precautions and to avoid applying to this high impedance voltage circuit any voltage higher than the maximum rated voltages. It is recommended for proper operation that V_{IN} and V_{OUT} be constrained to the range:

$V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$

To enhance reliability of operation, it is recommended to connect unused inputs to an appropriate logic voltage level such as V_{SS} or V_{DD}. All the voltages in the following table, are referenced to V_{SS}.

Table 33. Absolute Maximum Ratings

Symbol	Ratings	Value	Unit
V_{DD}	Recommended Supply Voltage	-0.3 to +6.0	V
V_{IN}	Input Voltage	V_{SS} -0.3 to V_{DD} + 0.3	V
V_{AIN}	Analog Input Voltage (A/D Converter)	V_{SS} -0.3 to V_{DD} + 0.3	V
V_{OUT}	Output Voltage	V_{SS} -0.3 to V_{DD} + 0.3	V
I_{IN}	Input Current	-10.....+10	mA
I_{OUT}	Output Current	-10.....+10	mA
I_{INJ}	Accumulated injected current of all I/O pins (V_{DD}, V_{SS})	40	mA
T_A	Operating Temperature Range	0 to +70	°C
T_{STG}	Storage Temperature Range	-65 to +150	°C
T_J	Junction Temperature	150	°C
PD	Power Dissipation	TBD	mW
ESD	ESD susceptibility	2000	V

6.1 POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in degrees Celsius, may be calculated using the following equation:

$$T_J = T_A + (P_D \times \theta J_A) \quad (1)$$

Where:

– T_A is the Ambient Temperature in °C,

– θJ_A is the Package Junction-to-Ambient Thermal Resistance, in °C/W,

– P_D is the sum of P_{INT} and $P_{I/O}$,

– P_{INT} is the product of $I_{DD\ and}$ V_{DD}, expressed in Watts. This is the Chip Internal Power

– $P_{I/O}$ represents the Power Dissipation on Input and Output Pins; User Determined.

For most applications $P_{I/O} < P_{INT}$ and may be neglected. $P_{I/O}$ may be significant if the device is configured to drive Darlington bases or sink LED Loads.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is given by:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Therefore:

$$K = P_D \times (T_A + 273°C) + \theta J_A \times P_D^2 \quad (3)$$

Where:

– K is a constant for the particular part, which may be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J may be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Table 34. Thermal Characteristics

Symbol	Package	Value	Unit
θJ_A	PSDIP42	95	°C/W
θJ_A	TQFP44	95	°C/W

6.2 AC/DC ELECTRICAL CHARACTERISTICS

(T_A = 0 to +70°C unless otherwise specified)

GENERAL						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ.**	**Max**	**Unit**
V_{DD}	Operating Supply Voltage	RUN & WAIT mode	4.0	5	5.5	V
I_{DD}	CPU RUN mode	I/O in input mode		14	18	mA
	CPU WAIT mode	V_{DD} = 5V		12	18	mA
	CPU HALT mode (see Note 1)	f_{CPU} = 8 MHz,	N/A			
	USB Suspend mode (see Note 2)	T_A = 20° C	N/A			

Note 1: HALT mode no longer exists.

Note 2: The USB cell must be put in suspend mode as well as the MCU in HALT mode. Since the latter no longer exists for enhanced arcing protection, the measurement of the USB suspend consumption parameter is no longer relevant.

CONTROL TIMING						
Symbol	**Parameter**	**Conditions**	**Value**			**Unit**
			Min	**Typ.**	**Max**	
f_{OSC} f_{CPU}	Frequency of Operation: external frequency internal frequency internal frequency	 f_{OSC}= 24MHz f_{OSC}=12MHz			 24 8 4	MHz
t_{BU}	Startup Time Built-Up Time	Crystal Resonator		8	20	ms
t_{RL}	External RESET Input pulse Width		1000			ns
t_{PORL}	Internal Power Reset Duration		4096			t_{CPU}
t_{POWL}	Watchdog RESET Output Pulse Width		500			ns
t_{DOG}	Watchdog Time-out	 f_{CPU} = 8 MHz	49152 6		3145728 384	t_{CPU} ms
t_{ILIL}	Interrupt Pulse Period			(1)		t_{CPU}
t_{OXOV}	Crystal Oscillator Start-up Time				50	ms
t_{DDR}	Power up rise time	V_{DD} min	1		100	ms

Note: : The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 cycles.

AC/DC ELECTRICAL CHARACTERISTICS (Cont'd)

STANDARD I/O PORT PINS						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
V_{OL}	Output Low Level Voltage Port A[7,2:0], Port B[7:4], Port C[7:0], Port D[6:0] Push Pull	I_{OL} = -1.6mA V_{DD}=5V	-	-	0.4	V
V_{OL}	Output Low Level Voltage Port A[6:3] Open Drain	I_{OL} = -1.6mA V_{DD}=5V	-	-	0.4	V
V_{OL}	Output Low Level Voltage Port A and Port C	I_{OL} = -10mA V_{DD}=5V	-	-	1.5	V
V_{OL}	Output Low Level Voltage Port B[3:0] Open Drain	I_{OL} = -3mA V_{DD}=5V	-	-	0.4	V
V_{OH}	Output High Level Voltage Port A[7, 2:0], Port B[7:4], Port C [7:0], Port D [6:0] Push Pull	I_{OH} = 1.6mA	V_{DD}-0.8	-	-	V
V_{IH}	Input High Level Voltage Port A [7:0], Port B [7:0], Port C [7:0], Port D[6:0], $\overline{RESET}$	Leading Edge	0.7xV_{DD}		V_{DD}	V
V_{IH}	HSYNC,VSYNCI, CSYNCI, HFBACK, VFBACK	V_{DD}= 5V	2.0			V
V_{IL}	HSYNC,VSYNCI, CSYNCI, HFBACK, VFBACK	V_{DD}= 5V			0.8	V
V_{IL}	Input Low Voltage Port A [7:0], Port B[7:0], Port C[7:0], Port D [6:0], $\overline{RESET}$	Trailing Edge	V_{SS}		0.3xV_{DD}	V
I_{IL}	I/O Ports Hi-Z Leakage Current Port A [7:0], Port B[7:0], Port C[7:0], Port D [6:0], $\overline{RESET}$				10	µA
C_{OUT} C_{IN}	Capacitance: Ports (as Input or Output), $\overline{RESET}$				12 8	pF pF
IRPU	Pull-up resistor current	V_{DD}=5V V_{IN}=V_{SS} T=25°C		280		µA

Note: Note: All voltages are referred to V_{SS} unless otherwise specified.

POWER ON/OFF Electrical Specifications						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
V_{TRH}	Power ON/OFF Reset Trigger V_{DD} rising edge	V_{DD} Variation 50mV/mS	3.35	3.65	3.9	V
V_{TRL}	Power ON/OFF Reset Trigger V_{DD} falling edge	V_{DD} Variation 50mV/mS	3.1	3.4	3.7	V
V_{TRM}	V_{DD} minimum for Power ON/OFF Reset active	V_{DD} Variation 50mV/mS		2.0	2.2	V
V_{TRHyst}	Power ON/OFF LVD Hysteresis	V_{DD} Variation 50mV/mS		250		mV

AC/DC ELECTRICAL CHARACTERISTICS (Cont'd)

8-bit A/D Converter						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
f_{ADC}	Analog control frequency	V_{DD}=5V			2	MHz
|TUE|	Total unadjusted error	f_{CPU}=8MHz, f_{ADC}=2MHz V_{DD}=5V			2	LSB
OE	Offset error		-1		1	
GE	Gain error		-1		1	
|DLE|	Differential linearity error				0.5	
|ILE|	Integral linearity error				0.5	
V_{AIN}	Conversion range voltage		V_{SS}		V_{DD}	V
I_{ADC}	A/D conversion supply current	f_{CPU}=8MHz, f_{ADC}=2MHz V_{DD}=5V		1		mA
t_{STAB}	Stabilization time after enable ADC				1	µs
t_{LOAD}	Sample capacitor loading time			1 4		µs $1/f_{ADC}$
t_{CONV}	Conversion time			2 8		µs $1/f_{ADC}$
R_{AIN}	External input resistor				15	kΩ
R_{ADC}	Internal input resistor			1.5		kΩ
C_{SAMPLE}	Sample capacitor			6		pF

PWM/BRM Electrical and Timings						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
F	Repetition rate	T_{CPU} =125ns		125		kHz
Res	Resolution	T_{CPU}=125ns		125		ns
s	Output step	V_{DD}=5V, 10 bits		5		mV

AC/DC ELECTRICAL CHARACTERISTICS (Cont'd)

I2C/DDC-Bus Electrical specifications						
Parameter	**Symbol**	**Unit**	**Standard mode I2C**		**Fast mode I2C**	
			Min	**Max**	**Min**	**Max**
Hysteresis of Schmitt trigger inputs Fixed input levels V_{DD}-related input levels	V_{HYS}	V	 na na	 na na	 0.2 0,05 V_{DD}	
Pulse width of spikes which must be suppressed by the input filter	T_{SP}	ns	na	na	0 ns	50 ns
Output fall time from VIH min to VIL max with a bus capacitance from 10 pF to 400 pF with up to 3 mA sink current at VOL1 with up to 6 mA sink current at VOL2	T_{OF}	ns	 na	 250 na	 20+0.1Cb 20+0.1Cb	 250 250
Input current each I/O pin with an input voltage between 0.4V and 0.9 V_{DD} max	I	µA	- 10	10	-10	10
Capacitance for each I/O pin	C	pF		10		10

na = not applicable
Cb = capacitance of one bus in pF

I2C/DDC-Bus Timings						
Parameter	**Standard I2C**		**Fast I2C**		**Symbol**	**Unit**
	Min	**Max**	**Min**	**Max**		
Bus free time between a STOP and START condition	4.7		1.3		T_{BUF}	ms
Hold time START condition. After this period, the first clock pulse is generated	4.0		0.6		$T_{HD:STA}$	µs
LOW period of the SCL clock	4.7		1.3		T_{LOW}	µs
HIGH period of the SCL clock	4.0		0.6		T_{HIGH}	µs
Set-up time for a repeated START condition	4.7		0.6		$T_{SU:STA}$	µs
Data hold time	0 (1)		0 (1)	0.9(2)	$T_{HD:DAT}$	ns
Data set-up time	250		100		$T_{SU:DAT}$	ns
Rise time of both SDA and SCL signals		1000	20+0.1Cb	300	T_R	ns
Fall time of both SDA and SCL signals		300	20+0.1Cb	300	TF	ns
Set-up time for STOP condition	4.0		0.6		$T_{SU:STO}$	ns
Capacitive load for each bus line		400		400	Cb	pF

1)The device must internally provide a hold time of at least 300 ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL

2)The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal

Cb = total capacitance of one bus line in pF

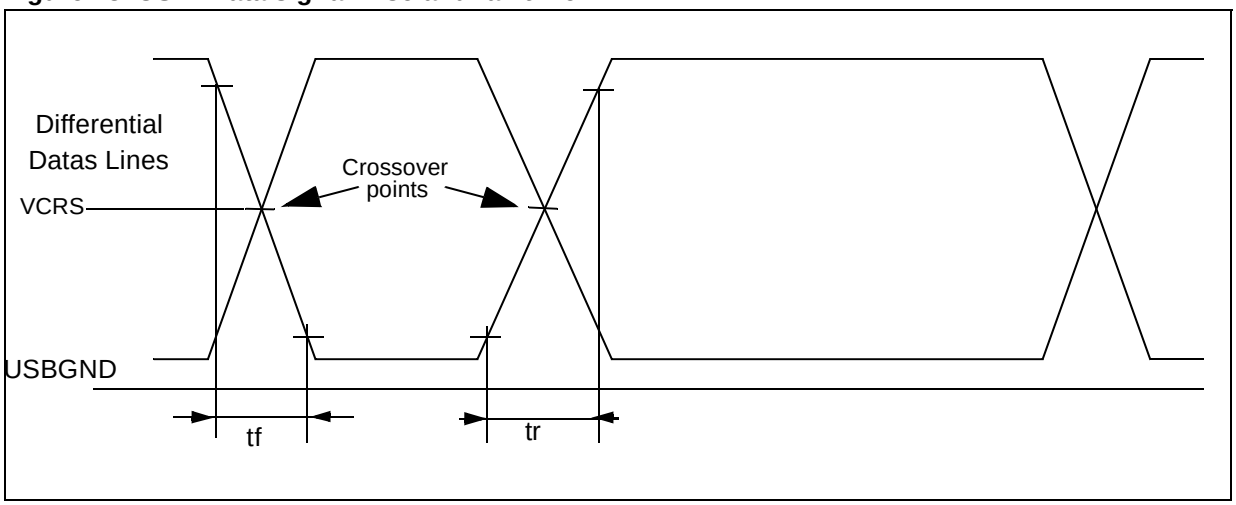
USB DC Electrical Characteristics					
Parameter	**Symbol**	**Conditions**	**Min.**	**Max.**	**Unit**
Inputs Levels:					
Differential Input Sensitivity	VDI	AbsI((D+) - (D-))	0.2		V
Differential Common Mode Range	VCM	Includes VDI range	0.8	2.5	V
Single Ended Receiver Threshold	VSE		0.8	2.0	V
Output Levels					
Static Output Low	VOL	RL of 1.5K ohms to 3.6V		0.3	V
Static Output High	VOH	RL of 15K ohms to USBGND	2.8	3.6	V
USBVCC: voltage level	USBV	V_{DD}=5V	3	3.6	V

Notes:

– RL is the load connected on the USB drivers.

– All the voltages are measured from the local ground potential (USBGND).

Figure 78. USB: Data signal Rise and fall time



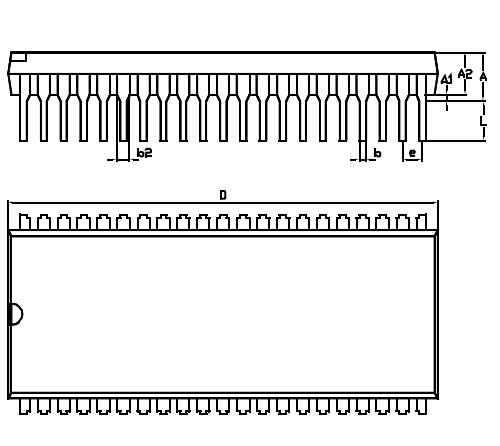
USB: Low speed electrical characteristics					
Parameter	**Symbol**	**Conditions**	**Min**	**Max**	**Unit**
Driver characteristics:					
Rise time	tr	Note 1, CL=50 pF	75		ns
		Note 1, CL=600 pF		300	ns
Fall Time	tf	Note 1, CL=50 pF	75		ns
		Note 1, CL=600 pF		300	ns
Rise/ Fall Time matching	trfm	tr/tf	80	120	%
Output signal Crossover Voltage	VCRS		1.3	2.0	V

Note1: Measured from 10% to 90% of the data signal

7 GENERAL INFORMATION

7.1 PACKAGE MECHANICAL DATA

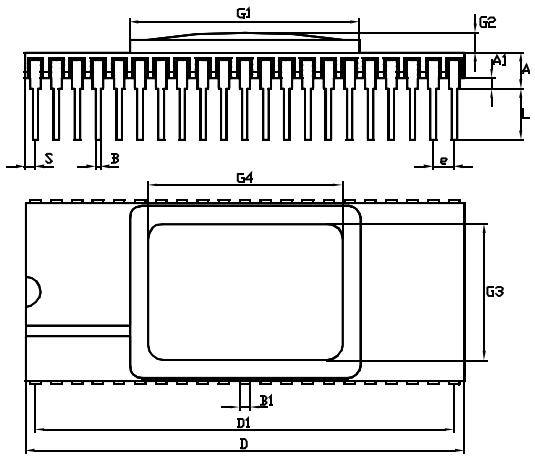
Figure 79. 42-Pin Shrink Plastic Dual In-Line Package, 600-mil Width



PDIP42S

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.200
A1	0.51			0.020		
A2	3.05	3.81	4.57	0.120	0.150	0.180
b		0.46	0.56		0.018	0.022
b2		1.02	1.14		0.040	0.045
C	0.23	0.25	0.38	0.009	0.010	0.015
D	36.58	36.83	37.08	1.440	1.450	1.460
E	15.24		16.00	0.600		0.630
E1	12.70	13.72	14.48	0.500	0.540	0.570
e		1.78			0.070	
eA		15.24			0.600	
eB			18.54			0.730
eC	0.00		1.52	0.000		0.060
L	2.54	3.30	3.56	0.100	0.130	0.140
	Number of Pins					
N	42					

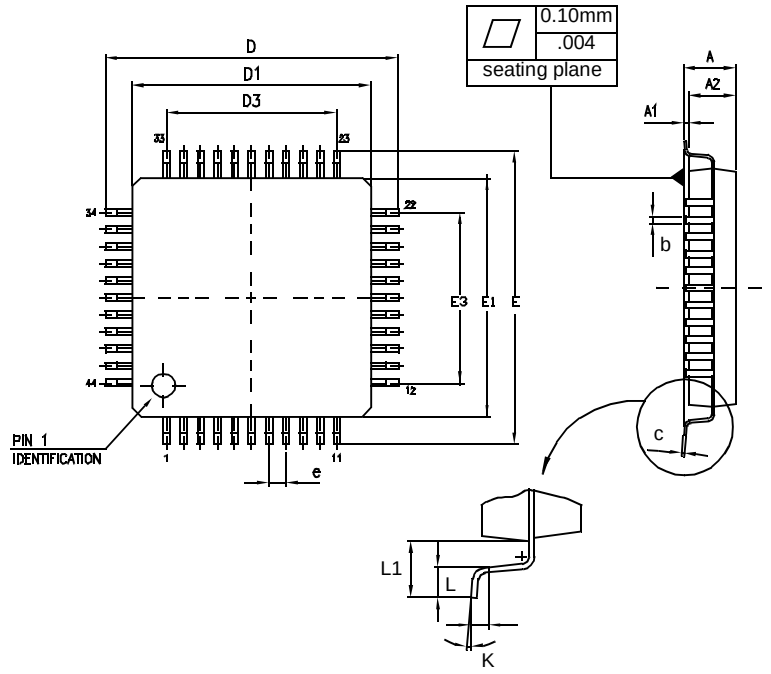
Figure 80. 42-Pin Shrink Ceramic Dual In-Line Package, 600-mil Width



CDIP42SW

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			4.01			0.158
A1	0.76			0.030		
B	0.38	0.46	0.56	0.015	0.018	0.022
B1	0.76	0.89	1.02	0.030	0.035	0.040
C	0.23	0.25	0.38	0.009	0.010	0.015
D	36.68	37.34	38.00	1.444	1.470	1.496
D1		35.56			1.400	
E1	14.48	14.99	15.49	0.570	0.590	0.610
e		1.78			0.070	
G	14.12	14.38	14.63	0.556	0.566	0.576
G1	18.69	18.95	19.20	0.736	0.746	0.756
G2		1.14			0.045	
G3	11.05	11.30	11.56	0.435	0.445	0.455
G4	15.11	15.37	15.62	0.595	0.605	0.615
L	2.92		5.08	0.115		0.200
S		0.89			0.035	
	Number of Pins					
N	42					

Figure 81. 44-Pin Thin Quad Flat Package



Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	0.37	0.45	0.012	0.015	0.018
c	0.09		0.20	0.004		0.008
D		12.00			0.472	
D1		10.00			0.394	
D3		8.00			0.315	
E		12.00			0.472	
E1		10.00			0.394	
E3		8.00			0.315	
e		0.80			0.031	
K	0°	3.5°	7°			
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
	Number of Pins					
N	44					

8 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to STMicroelectronics.

8.1 Transfer of Customer Code

Customer code is made up of the ROM contents and the list of the selected mask options (if any). The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file in .S19 format generated by the development tool. All unused bytes must be set to 9Dh (opcode for NOP).

The selected mask options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Figure 82. Sales Type Coding Rules

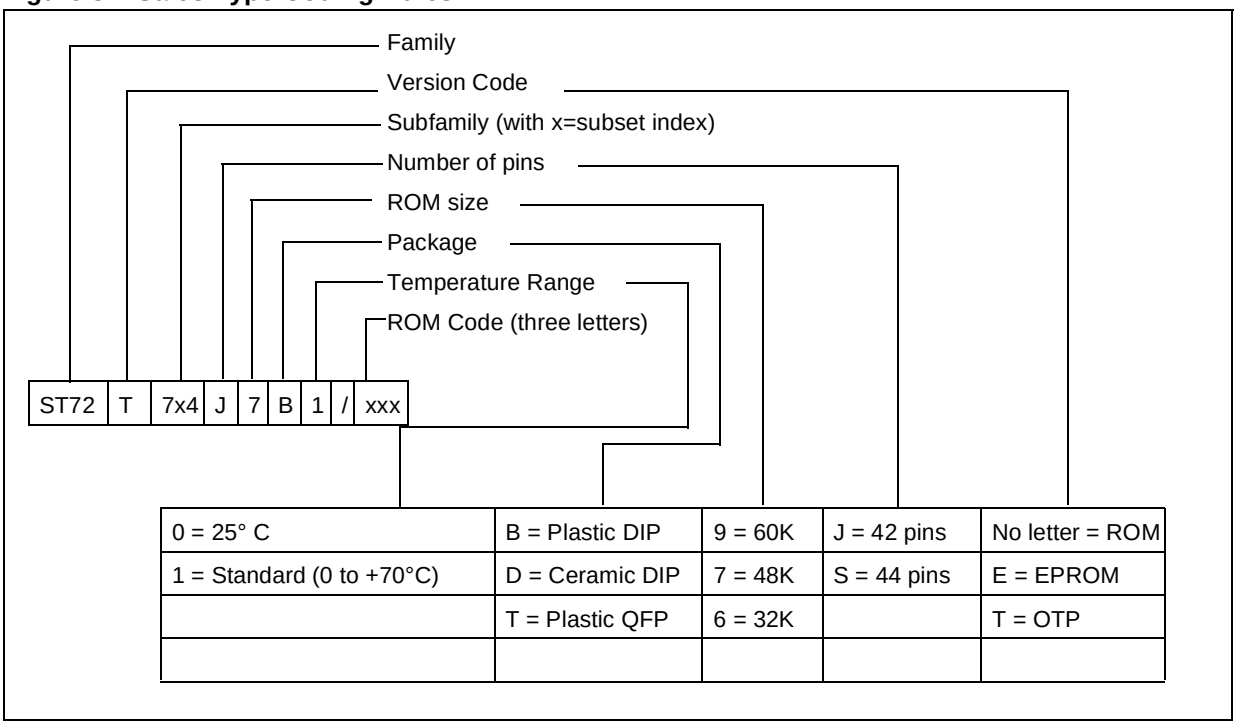


Table 35. Development Tools

Development Tool	Sales Type	Remarks
Real time Emulator	ST727x4-EMU2B	
EPROM Programmer Board	ST727x4-EPB/xx[1]	220V Power Supply EU 110V Power Supply US
Gang Programmer	ST72E774-GP/D42 ST72E774-GP/Q44	DIL42 PQFP44

[1] xx stands for the power supply code assigned by ST Microelectronics: EU=220V; US=110V

Table 36. Ordering Information

Sales Type	ROM/EPROM (bytes)	RAM (bytes)	TMU	USB	Package
ST72X774 (1)					
ST72E774J9D0	60K EPROM	1K	Yes	Yes	CSDIP42
ST72T774J9B1	60K OTP				PSDIP42
ST72774J9B1/xxx	60K ROM				
ST72774J7B1/xxx	48K ROM				
ST72774S7T1/xxx	48K ROM				TQFP44
ST72T774S9T1	60K OTP				
ST72774S9T1/xxx	60K ROM				
ST72X754 (1)					
ST72E754J9D0	60K EPROM	1K	Yes	No	CSDIP42
ST72T754J9B1	60K OTP				PSDIP42
ST72754J9B1/xxx	60K ROM				
ST72754J7B1/xxx	48K ROM				
ST72T754S9T1	60K OTP				TQFP44
ST72754S9T1	60K ROM				
ST72754S7T1/xxx	48K ROM				
ST72X734 (2)					
ST72E734J6D0	32K EPROM	512	No	No	CSDIP42
ST72T734J6B1/xxx	32K OTP				PSDIP42
ST72734J6B1/xxx	32K ROM				

(1) 8 bit ±2 LSB A/D converter

(2) 8 bit ±4 LSB A/D converter

STMicroelectronics OPTION LIST

ST727x4 MICROCONTROLLER FAMILY

Customer: . . . . . .
Address: . . . . . .
. . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference/ROM Code* : . . . . . .
*The ROM code name assigned by ST.

STMicroelectronics reference:

Device (SDIP42):	[ ]	ST72774J9B1 (60K ROM)
	[ ]	ST72774J7B1 (48K ROM)
	[ ]	ST72754J9B1 (60K ROM)
	[ ]	ST72754J7B1 (48K ROM)
	[ ]	ST72734J6B1 (32K ROM)
Device (TQFP44):	[ ]	ST72774S9T1 (60K ROM)
	[ ]	ST72774S7T1 (48K ROM)
	[ ]	ST72754S9T1 (60K ROM)
	[ ]	ST72754S7T1 (48K ROM)
Device (CSDIP42):	[ ]	ST72E774J9D0 (60K EPROM)
	[ ]	ST72E754J9D0 (60K EPROM)
	[ ]	ST72E734J6D0 (32K EPROM)
Software Development:	[ ]	STMicroelectronics
	[ ]	Customer
	[ ]	External laboratory

Special Marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _"
For marking, one line is possible with maximum 16 characters for SDIP42 and 10 characters for TQFP44.
Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Mask Options: None.

We have checked the ROM code verification file returned to us by STMicroelectronics. It conforms exactly with the ROM code file orginally supplied. We therefore authorize STMicroelectronics to proceed with device manufacture.

Signature . . . . . .
Date . . . . . .