High-Performance 8-Bit Microcontrollers

Z8 Encore!® F083A Series

Product Specification

PS026310-1212

PRELIMINARY

www.zilog.com

⚠ **Warning:** DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG’S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore! and Z8 Encore! XP are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Chapter/Section	Description	Page No.
Dec 2012	10	GPIO	Modified GPIO Port D0 language in Shared Reset Pin section and Port Alternate Function Mapping table.	35, 36
Jul 2011	09	LED Drive Enable Register, Flash Sector Protect Register, Packaging	Clarified statement surrounding the Alternate Function Register as it relates to the LED function; revised Flash Sector Protect Register description; revised Packaging chapter.	51, 122, 198
Dec 2007	08	n/a	Removed XP from F083A.	All
Nov 2007	07	DC Characteristics, Analog-to-Digital Converter Electrical Characteristics and Timing	Updated Tables 117 and 123.	185, 192
Sep 2007	06	DC Characteristics	Added ICC Active specification and ICC HALT specification in Table 117.	185
Aug 2007	05	n/a	First release	All

Table of Contents

List of Figures

List of Tables

Overview

Zilog's Z8 Encore! MCU family of products are the first in a line of Zilog microcontroller products based on the 8-bit eZ8 CPU. The Z8 Encore! F083A Series products expand on Zilog's extensive line of 8-bit microcontrollers. The Flash in-circuit programming capability allows for faster development time and program changes in the field. The new eZ8 CPU is upward-compatible with existing Z8 CPU instructions. The rich peripheral set of Z8 Encore! F083A Series makes it suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices and sensors.

Features

Z8 Encore! F083A Series MCU include the following key features:

- 20MHz eZ8 CPU
- Up to 8KB Flash memory with in-circuit programming capability
- Up to 256 B register RAM
- 100 B nonvolatile data storage (NVDS)
- Up to 23 I/O pins depending upon package
- Internal precision oscillator (IPO)
- External crystal oscillator
- Two enhanced 16-bit timers with capture, compare and PWM capability
- Watchdog Timer (WDT) with dedicated internal RC oscillator
- Single-pin, On-Chip Debugger (OCD)
- Fast 8-channel, 10-bit Analog-to-Digital Converter (ADC)
- On-chip analog comparator
- Up to 17 interrupt sources
- Voltage Brown-Out protection (VBO)
- Power-On Reset (POR)
- 2.7V to 3.6V operating voltage
- Up to thirteen 5V-tolerant input pins
- 20-pin and 28-pin packages

- 0°C to +70°C standard temperature range and –40°C to +105°C extended temperature operating ranges

Part Selection Guide

Table 1 lists the basic features available for each device within the Z8 Encore! F083A Series product line. For details, see the Ordering Information chapter on page 199.

Table 1. Z8 Encore! F083A Series Family Part Selection Guide

Part Number	Flash (KB)	RAM (B)	NVDS (100B)	ADC	I/O Pins
Z8F083A	8	256	Yes	Yes	17/23
Z8F043A	4	256	Yes	Yes	17/23

Block Diagram

Figure 1 displays a block diagram of the Z8 Encore! F083A Series architecture.

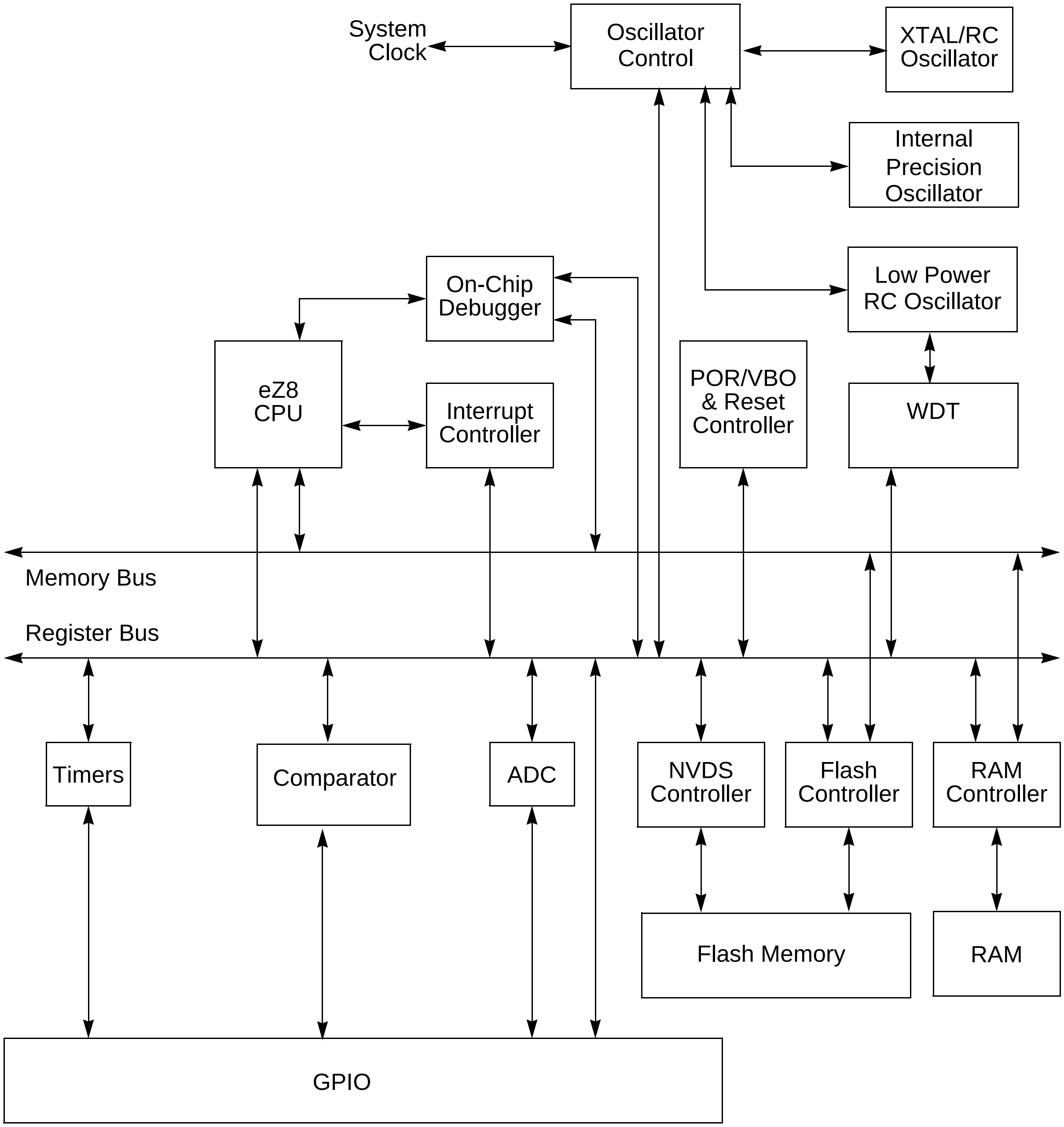


Figure 1. Z8 Encore! F083A Series Block Diagram

CPU and Peripheral Overview

Zilog's 8-bit eZ8 CPU meets the continuing demand for faster and more code-efficient microcontrollers. The eZ8 CPU executes a superset of the original Z8 instruction set. The eZ8 CPU features include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required program memory
- Software stack allows much greater depth in subroutine calls and interrupts than hardware stacks
- Compatible with existing Z8 CPU code
- Expanded internal Register File allows access up to 4KB
- New instructions improve execution efficiency for code developed using high-level programming languages, including C
- Pipelined instruction fetch and execute
- New instructions for improved performance including BIT, BSWAP, BTJ, CPC, LDC, LDCI, LEA, MULT and SRL
- New instructions support 12-bit linear addressing of the Register File
- Up to10 MIPS operation
- C Compiler-friendly
- Two to nine clock cycles per instruction

For more information regarding the eZ8 CPU, refer to eZ8 CPU Core User Manual (UM0128), available for download on www.zilog.com.

General Purpose Input/Output

The Z8 Encore! F083A Series features up to 23 port pins (Ports A–D) for general purpose input/output (GPIO). The number of GPIO pins available is a function of package. Each pin is individually programmable.

Flash Controller

The Flash Controller programs and erases Flash memory. It also supports protection against accidental programming and erasure.

Nonvolatile Data Storage

The nonvolatile data storage (NVDS) uses a hybrid hardware/software scheme to implement a byte programmable data memory and is capable of storing about 100,000 write cycles.

Internal Precision Oscillator

The internal precision oscillator (IPO) with accuracy of ±4% full voltage/temperature range is a trimable clock source that requires no external components.

External Crystal Oscillator

The external crystal oscillator circuit provides highly accurate clock frequencies with the use of an external crystal, ceramic resonator or RC network.

10-Bit Analog-to-Digital Converter

The analog-to-digital converter (ADC) converts an analog input signal to a 10-bit binary number. The ADC accepts inputs from eight different analog input pins. It has a fast 2.8 µs conversion speed.

Analog Comparator

The analog comparator compares the signal at an input pin with either an internal programmable reference voltage or with a signal at the second input pin. The comparator output is used either to drive a logic output pin or to generate an interrupt.

Timers

Two enhanced 16-bit reloadable timers are used for timing/counting events or for motor control operations. These timers provide a 16-bit programmable reload counter and operate in ONE-SHOT, CONTINUOUS, GATED, CAPTURE, CAPTURE RESTART, COMPARE, CAPTURE and COMPARE, PWM SINGLE OUTPUT and PWM DUAL OUTPUT modes.

Interrupt Controller

The Z8 Encore! F083A Series products support seventeen interrupt sources with sixteen interrupt vectors: up to five internal peripheral interrupts and up to twelve GPIO interrupts. These interrupts have three levels of programmable interrupt priority.

Reset Controller

The Z8 Encore! F083A Series products are reset using any one of the following: the $\overline{\text{RESET}}$ pin, POR, WDT time-out, STOP Mode exit or VBO warning signal. The $\overline{\text{RESET}}$ pin is bidirectional, that is, it functions as reset source as well as a reset indicator.

On-Chip Debugger

The Z8 Encore! F083A Series products feature an integrated OCD. The OCD provides a rich set of debugging capabilities, such as reading and writing registers, programming Flash memory, setting breakpoints and executing code. The OCD uses one single-pin interface for communication with an external host.

Acronyms and Expansions

This document references a number of acronyms; each is expanded in Table 2 for the reader's understanding.

Table 2. Acronyms and Expansions

Acronyms	Expansions
ADC	Analog-to-Digital Converter
NVDS	Nonvolatile Data Storage
WDT	Watchdog Timer
GPIO	General-Purpose Input/Output
OCD	On-Chip Debugger
POR	Power-On Reset
VBO	Voltage Brownout
IPO	Internal Precision Ocsillator
PDIP	Plastic Dual Inline Package
SOIC	Small Outline Integrated Circuit
SSOP	Small Shrink Outline Package
QFN	Quad Flat No Lead
IRQ	Interrupt request
ISR	Interrupt service routine
MSB	Most significant byte
LSB	Least significant byte
PWM	Pulse Width Modulation
SAR	Successive Approximation Register

Pin Description

The Z8 Encore! F083A Series products are available in variety of package styles and pin configurations. This chapter describes the signals and the pin configurations for each of the package styles. For information about the physical package specifications, see the Packaging chapter on page 198.

Available Packages

Table 3 lists the package styles that are available for each device in the Z8 Encore! F083A Series product line.

Table 3. Z8 Encore! F083A Series Package Options

Part Number	ADC	20-pin QFN	20-pin SOIC	20-pin SSOP	20-pin PDIP	28-pin QFN	28-pin SOIC	28-pin SSOP
Z8F083A	Yes	X	X	X	X	X	X	X
Z8F043A	Yes	X	X	X	X	X	X	X

Pin Configurations

Figures 2 through 5 display the pin configurations of all of the packages available in the Z8 Encore! F083A Series. For the description of the signals, see Table 4 on page 11.

The pin configurations listed are preliminary and subject to change based on manufacturing limitations.

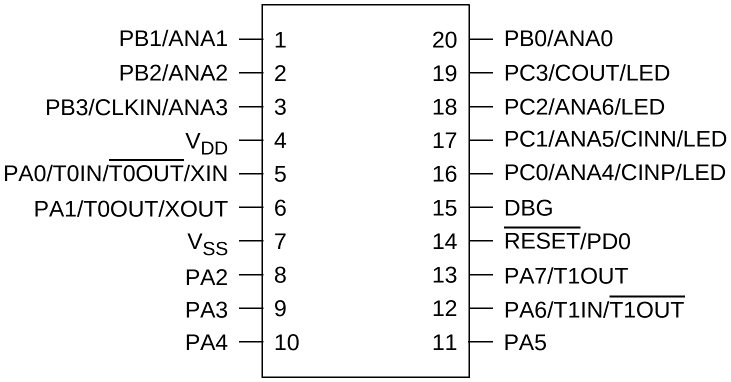


Figure 2. Z8F083A Series in 20-Pin SOIC, SSOP, PDIP Package

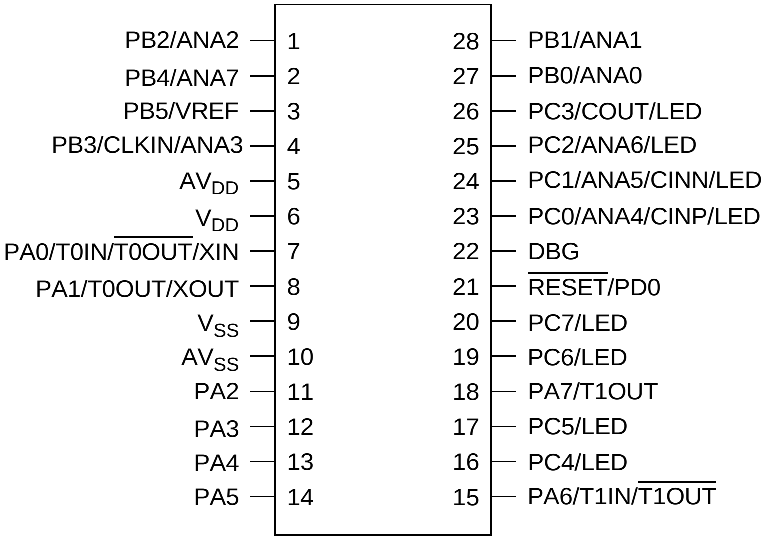


Figure 3. Z8F083A Series in 28-Pin SOIC and SSOP Packages

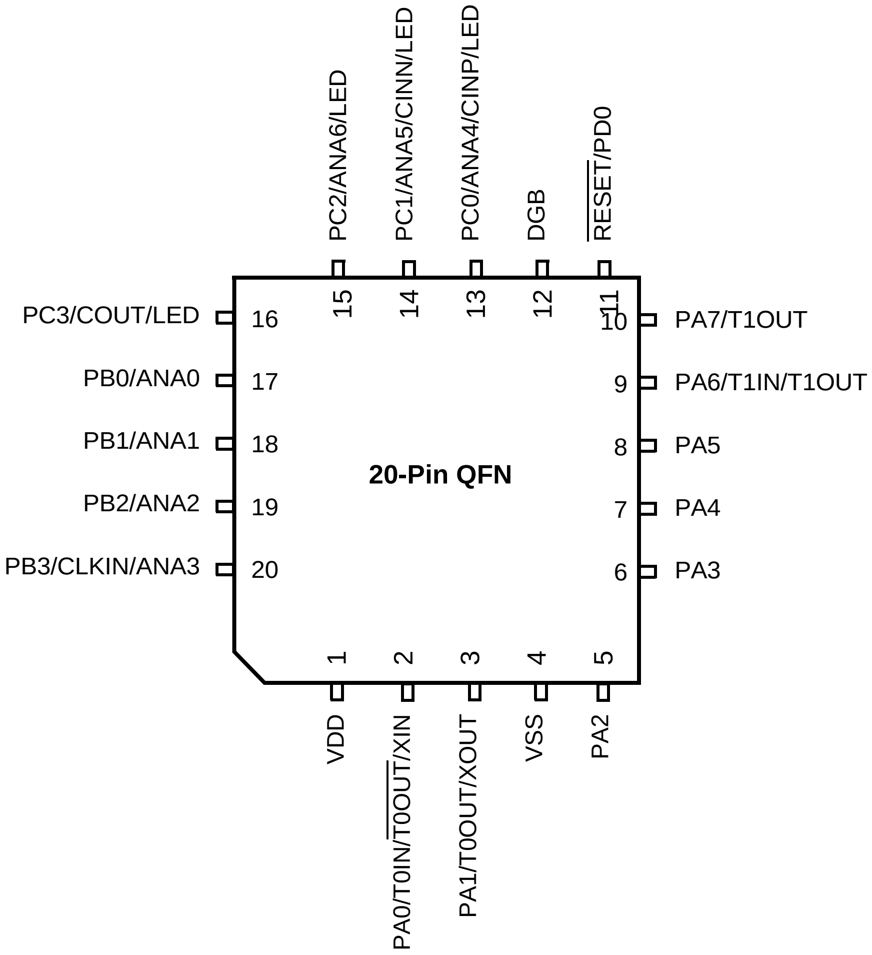


Figure 4. Z8F083A Series in 20-Pin QFN Package

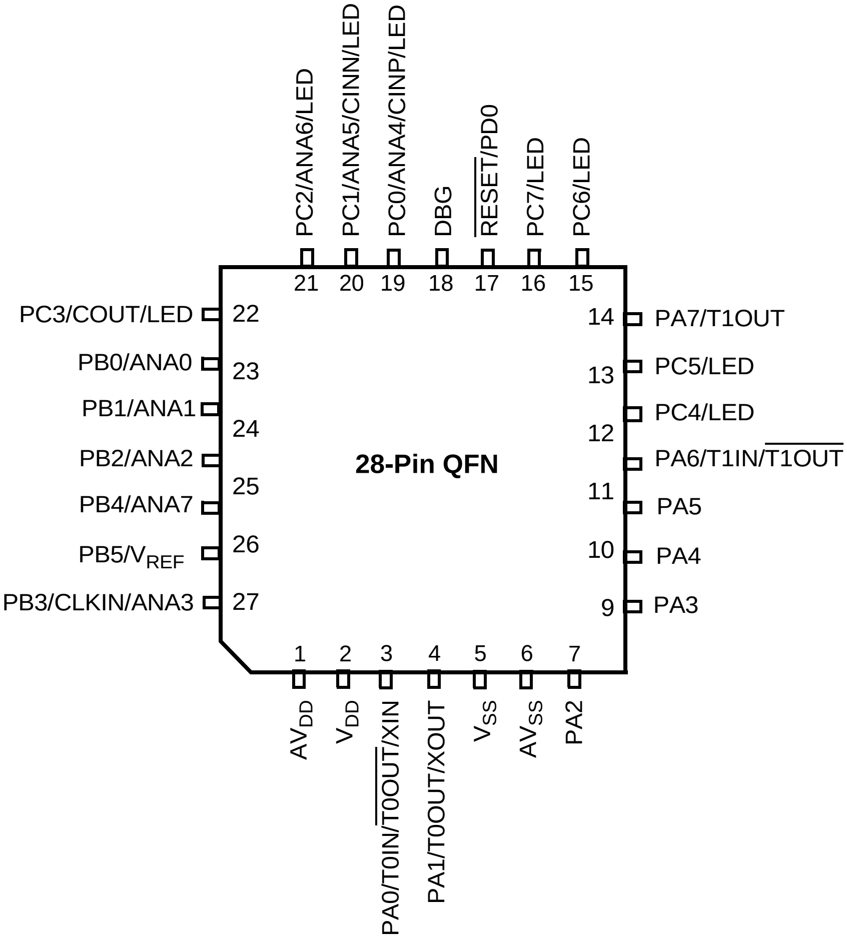


Figure 5. Z8F083A Series in 28-Pin QFN Package

Signal Descriptions

Table 4 describes the Z8 Encore! F083A Series signals. To determine the signals available for a specific package style, see the Pin Configurations section on page 7.

Table 4. Signal Descriptions

Signal Mnemonic	I/O	Description
General-Purpose Input/Output Ports A–D		
PA[7:0]	I/O	Port A. These pins are used for GPIO.
PB[5:0]	I/O	Port B. These pins are used for GPIO.
PC[7:0]	I/O	Port C. These pins are used for GPIO.
PD[0]	I/O	Port D. This pin is used for general purpose output only.
Timers		
T0OUT/T1OUT	O	Timer output 0–1. These signals are the output from the timers.
$\overline{\text{T0OUT}}$/$\overline{\text{T1OUT}}$	O	Timer complement output 0–1. These signals are output from the timers in PWM DUAL OUTPUT mode.
T0IN/T1IN	I	Timer Input 0–1. These signals are used as the capture, gating and counter inputs. The T0IN signal is multiplexed $\overline{\text{T0OUT}}$ signals.
Comparator		
CINP/CINN	I	Comparator inputs. These signals are the positive and negative inputs to the comparator.
COUT	O	Comparator output. This is the output of the comparator.
Analog		
ANA[7:0]	I	Analog port. These signals are used as inputs to the Analog-to-Digital Converter (ADC).
VREF	I/O	ADC reference voltage input. Note: When configuring ADC using external Vref, PB5 is used as VREF in 28-pin package.
Oscillators		
XIN	I	External crystal input. This is the input pin to the crystal oscillator. A crystal is connected between it and the **XOUT** pin to form the oscillator. In addition, this pin is used with external RC networks or external clock drivers to provide the system clock.
XOUT	O	External crystal output. This pin is the output of the crystal oscillator. A crystal is connected between it and the **XIN** pin to form the oscillator.
Clock Input		
CLKIN	I	Clock input signal. This pin can be used to input a TTL-level signal to be used as the system clock.

Table 4. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
LED Drivers		
LED	O	Direct LED drive capability. All Port C pins have the capability to drive an LED without any other external components. These pins have programmable drive strengths set by the GPIO block.
On-Chip Debugger		
DBG	I/O	Debug. This signal is the control and data input and output to and from the On-Chip Debugger. **Caution:** The DBG pin is open-drain and requires an external pull-up resistor to ensure proper operation.
Reset		
$\overline{\text{RESET}}$	I/O	RESET. Generates a reset when asserted (driven Low). Also serves as a reset indicator; the Z8 Encore! forces this pin low when in reset. This pin is open-drain and features an enabled internal pull-up resistor.
Power Supply		
V_{DD}	I	Digital power supply.
AV_{DD}	I	Analog power supply.
V_{SS}	I	Digital ground.
AV_{SS}	I	Analog ground.

Pin Characteristics

Table 5 provides detailed characteristics of each pin available on the Z8 Encore! F083A Series 20- and 28-pin devices. The data in Table 5 are sorted alphabetically by the pin symbol mnemonic.

Table 5. Pin Characteristics (20- and 28-pin Devices)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt Trigger Input	Open Drain Output	5V Tolerance
AV_{DD}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
AV_{SS}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	NA
DBG	I/O	I	N/A	Yes	No	Yes	Yes	Yes
PA[7:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable	PA[7:2] only
PB[5:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable	No
PC[7:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable	PC[7:3] only
RESET/ PD0	I/O	I/O (defaults to $\overline{\text{RESET}}$)	Low (in RESET mode)	Yes (PD0 only)	Programmable for PD0; always on for $\overline{\text{RESET}}$	Yes	Programmable for PD0; always on for $\overline{\text{RESET}}$	Yes
V_{DD}	N/A	N/A	N/A	N/A			N/A	N/A
V_{SS}	N/A	N/A	N/A	N/A			N/A	N/A

Address Space

The eZ8 CPU accesses three distinct address spaces as follows:

- The Register File addresses access for the general purpose registers and the eZ8 CPU, peripheral and GPIO port control registers
- Program Memory addresses access for all of the memory locations having executable code and/or data
- The Data Memory addresses access for all of the memory locations containing only the data

The following sections describe these three address spaces. For more detailed information about the eZ8 CPU and its address space, refer to the eZ8 CPU Core User Manual (UM0128), available for download on www.zilog.com.

Register File

The Register File address space in the Z8 Encore! MCU is 4KB (4096 bytes). The Register File consists of two sections: control registers and general purpose registers. When instructions are executed, registers defined as source are read and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general purpose registers to function as accumulators, address pointers, index registers, stack areas or scratch pad memory.

The upper 256 bytes of the 4KB Register File address space are reserved for control of the eZ8 CPU, on-chip peripherals and the I/O ports. These registers are located at addresses from `F00H` to `FFFH`. Some of the addresses within the 256 B Control Register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and produces unpredictable results.

The on-chip RAM always begins at address `000H` in the Register File address space. The Z8 Encore! F083A Series devices contain up to 256 B of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the Control Register address space), returns an undefined value. Writing to these Register File addresses has no effect.

Program Memory

The eZ8 CPU supports 64KB of program memory address space. The Z8 Encore! F083A Series devices contain 1KB to 12KB of on-chip Flash memory in the program memory address space, depending on the device. Reading from program memory addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented program memory addresses produces no effect. Table 6 describes the program memory maps for the Z8 Encore! F083A Series products.

Table 6. Z8 Encore! F083A Series Program Memory Maps

Program Memory Address (Hex)	Function
Z8F083A Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–003D	Interrupt Vectors*
003E–1FFF	Program Memory
Z8F043A Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–003D	Interrupt Vectors*
003E–0FFF	Program Memory

Note: *See Table 34 on page 55 for a list of interrupt vectors.

Data Memory

The Z8 Encore! F083A Series does not use the eZ8 CPU's 64KB data memory address space.

Flash Information Area

Table 7 indicates the Z8 Encore! F083A Series MCUs' Flash information area. This 128-byte information area is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash information area is mapped into program memory and overlays the 128 bytes at addresses FE00H to FE7FH. When information area access is enabled, all reads from these program memory addresses return information area data rather than program memory data. Access to the Flash information area is read-only.

Table 7. Z8 Encore! F083A Series Flash Memory Information Area Map

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog option bits
FE40–FE53	Part Number 20-character ASCII alphanumeric code Left-justified and filled with FH.
FE54–FE5F	Reserved.
FE60–FE7F	Zilog calibration data.
FE80–FFFF	Reserved.

Register Map

Table 8 provides an address map for the Z8 Encore! F083A Series register file. Consider registers for unimplemented peripherals to be reserved.

Table 8. Register File Address Map

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
General Purpose RAM				
000–0FF	General Purpose Register File RAM	—	XX	
100–EFF	Reserved	—	XX	
Timer 0				
F00	Timer 0 High Byte	T0H	00	84
F01	Timer 0 Low Byte	T0L	01	84
F02	Timer 0 Reload High Byte	T0RH	FF	85
F03	Timer 0 Reload Low Byte	T0RL	FF	85
F04	Timer 0 PWM High Byte	T0PWMH	00	86
F05	Timer 0 PWM Low Byte	T0PWML	00	86
F06	Timer 0 Control 0	T0CTL0	00	87
F07	Timer 0 Control 1	T0CTL1	00	88
Timer 1				
F08	Timer 1 High Byte	T1H	00	84
F09	Timer 1 Low Byte	T1L	01	84
F0A	Timer 1 Reload High Byte	T1RH	FF	85
F0B	Timer 1 Reload Low Byte	T1RL	FF	85
F0C	Timer 1 PWM High Byte	T1PWMH	00	86
F0D	Timer 1 PWM Low Byte	T1PWML	00	86
F0E	Timer 1 Control 0	T1CTL0	00	87
F0F	Timer 1 Control 1	T1CTL1	00	84
F10–F6F	Reserved	—	XX	
Analog-to-Digital Converter				
F70	ADC Control 0	ADCCTL0	00	102
F71	Reserved	—	XX	
F72	ADC Data High Byte	ADCD_H	XX	103
F73	ADC Data Low Bits	ADCD_L	XX	104

Note: XX = Undefined.

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
Analog-to-Digital Converter (cont'd.)				
F74	ADC Sample Settling Time	ADCSST	0F	105
F75	ADC Sample Time	ADCST	3F	106
F76	ADC Clock Prescale	ADCCP	00	107
F77–F7F	Reserved	—	XX	
Low Power Control				
F80	Power Control 0	PWRCTL0	88	32
F81	Reserved	—	XX	
LED Controller				
F82	LED Drive Enable	LEDEN	00	51
F83	LED Drive Level High	LEDLVLH	00	52
F84	LED Drive Level Low	LEDLVLL	00	53
F85	Reserved	—	XX	
Oscillator Control				
F86	Oscillator Control	OSCCTL	A0	154
F87–F8F	Reserved	—	XX	
Comparator 0				
F90	Comparator 0 Control	CMP0	14	109
F91–FBF	Reserved	—	XX	
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	59
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	62
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	62
FC3	Interrupt Request 1	IRQ1	00	60
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	63
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	64
FC6	Interrupt Request 2	IRQ2	00	61
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	65
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	65
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	67
FCE	Shared Interrupt Select	IRQSS	00	67

Note: XX = Undefined.

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
Interrupt Controller (cont'd.)				
FCF	Interrupt Control	IRQCTL	00	68
GPIO Port A				
FD0	Port A Address	PAADDR	00	39
FD1	Port A Control	PACTL	00	41
FD2	Port A Input Data	PAIN	XX	41
FD3	Port A Output Data	PAOUT	00	41
GPIO Port B				
FD4	Port B Address	PBADDR	00	39
FD5	Port B Control	PBCTL	00	41
FD6	Port B Input Data	PBIN	XX	41
FD7	Port B Output Data	PBOUT	00	41
GPIO Port C				
FD8	Port C Address	PCADDR	00	39
FD9	Port C Control	PCCTL	00	41
FDA	Port C Input Data	PCIN	XX	41
FDB	Port C Output Data	PCOUT	00	41
GPIO Port D				
FDC	Port D Address	PDADDR	00	39
FDD	Port D Control	PDCTL	00	41
FDE	Reserved	—	XX	
FDF	Port D Output Data	PDOUT	00	41
FE0–FEF	Reserved	—	XX	
Watchdog Timer				
FF0	Reset Status	RSTSTAT	XX	95
	WDT Control	WDTCTL	XX	95
FF1	WDT Reload upper byte	WDTU	FF	96
FF2	WDT Reload High Byte	WDTH	FF	96
FF3	WDT Reload Low Byte	WDTL	FF	97
FF4–FF5	Reserved	—	XX	

Note: XX = Undefined.

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
Trim Bit Control				
FF6	Trim Bit Address	TRMADR	00	126
FF7	Trim Data	TRMDR	XX	127
Flash Memory Controller				
FF8	Flash Control	FCTL	00	120
FF8	Flash Status	FSTAT	00	121
FF9	Flash Page Select	FPS	00	122
	Flash Sector Protect	FPROT	00	122
FFA	Flash Programming Frequency High Byte	FFREQH	00	123
FFB	Flash Programming Frequency Low Byte	FFREQL	00	123
eZ8 CPU				
FFC	Flags	—	XX	Refer to the eZ8 CPU Core User Manual (UM0128)
FFD	Register Pointer	RP	XX	
FFE	Stack Pointer High Byte	SPH	XX	
FFF	Stack Pointer Low Byte	SPL	XX	
Note: XX = Undefined.				

Reset and Stop Mode Recovery

The reset controller in the Z8 Encore! F083A Series controls Reset and Stop Mode Recovery operations. In a typical operation, the following events cause a Reset:

- Power-On Reset
- Voltage Brown-Out
- Watchdog Timer time-out (when configured by the WDT_RES Flash option bit to initiate a reset)
- External $\overline{\text{RESET}}$ pin assertion (when the alternate Reset function is enabled by the GPIO register)
- On-Chip Debugger initiated reset (OCDCTL[0] set to 1)

When the device is in STOP Mode, a Stop Mode Recovery is initiated by either of the following:

- WDT time-out
- GPIO port input pin transition on an enabled Stop Mode Recovery source

The VBO circuitry on the device generates the VBO reset when the supply voltage drops below a minimum safe level.

Reset Types

The Z8 Encore! F083A Series provides different types of Reset operation. Stop Mode Recovery is considered as a form of reset. Table 9 lists the types of reset and their operating characteristics. The system reset is longer, if the external crystal oscillator is enabled by the Flash option bits, allowing additional time for oscillator start-up.

Table 9. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	Control Registers	eZ8 CPU	Reset Latency (Delay)
System Reset	Reset (as applicable)	Reset	About 66 internal precision oscillator cycles.
System Reset with Crystal Oscillator Enabled	Reset (as applicable)	Reset	About 5000 internal precision oscillator cycles.
Stop Mode Recovery	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	About 66 internal precision oscillator cycles.
Stop Mode Recovery with crystal oscillator enabled	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	About 5000 internal precision oscillator cycles.

During a system Reset or Stop Mode Recovery, the Z8 Encore! F083A Series device is held in reset for about 66 cycles of the internal precision oscillator. If the crystal oscillator is enabled in the Flash option bits, the reset period is increased to about 5000 IPO cycles. When a reset occurs because of a low voltage condition or POR, the reset delay is measured from the time the supply voltage first exceeds the POR level (discussed later in this chapter). If the external pin reset remains asserted at the end of the reset period, the device remains in reset until the pin is deasserted.

At the beginning of reset, all GPIO pins are configured as inputs with pull-up resistor disabled, except PD0 which is shared with the reset pin. On Reset, the Port D0 pin is configured as a bidirectional open-drain reset. This pin is internally driven low during port reset, after which the user code reconfigures this pin as a general purpose output.

During reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer Oscillator continues to run.

On reset, control registers within the Register File that have a defined reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer and Flags) and general purpose RAM are undefined following the reset. The eZ8 CPU fetches the reset vector at program memory addresses `0002H` and `0003H` and loads that value into the program counter. Program execution begins at the reset vector address.

Because the control registers are reinitialized by a system reset, the system clock after reset is always the IPO. User software must reconfigure the oscillator control block, to enable and select the correct system clock source.

Reset Sources

Table 10 lists the possible sources of a system reset.

Table 10. Reset Sources and Resulting Reset Type

Operating Mode	Reset Source	Special Conditions
NORMAL or HALT modes	Power-On Reset / Voltage Brown-Out.	Reset delay begins after supply voltage exceeds POR level.
	WDT time-out when configured for reset.	None.
	$\overline{\text{RESET}}$ pin assertion.	All reset pulses less than four system clocks in width are ignored.
	On-Chip Debugger initiated reset (OCDCTL[0] set to 1).	System, except the On-Chip Debugger is unaffected by the reset.
STOP Mode	Power-On Reset / Voltage Brown-Out.	Reset delay begins after supply voltage exceeds POR level.
	$\overline{\text{RESET}}$ pin assertion.	All reset pulses less than 12 ns are ignored.
	DBG pin driven Low.	None.

Power-On Reset

Each device in the Z8 Encore! F083A Series contains an internal POR circuit. The POR circuit monitors the digital supply voltage and holds the device in the Reset state until the digital supply voltage reaches a safe operating level. After the supply voltage exceeds the POR voltage threshold (V_{POR}), the device is held in the Reset state until the POR counter has timed out. If the crystal oscillator is enabled by the option bits, the time-out is longer.

After the Z8 Encore! F083A Series device exits the POR state, the eZ8 CPU fetches the reset vector. Following the POR, the POR status bit in the Reset Status (RSTSTAT) Register is set to 1.

Figure 6 displays POR operation. For POR threshold voltage (V_{POR}), see the Electrical Characteristics chapter on page 184.

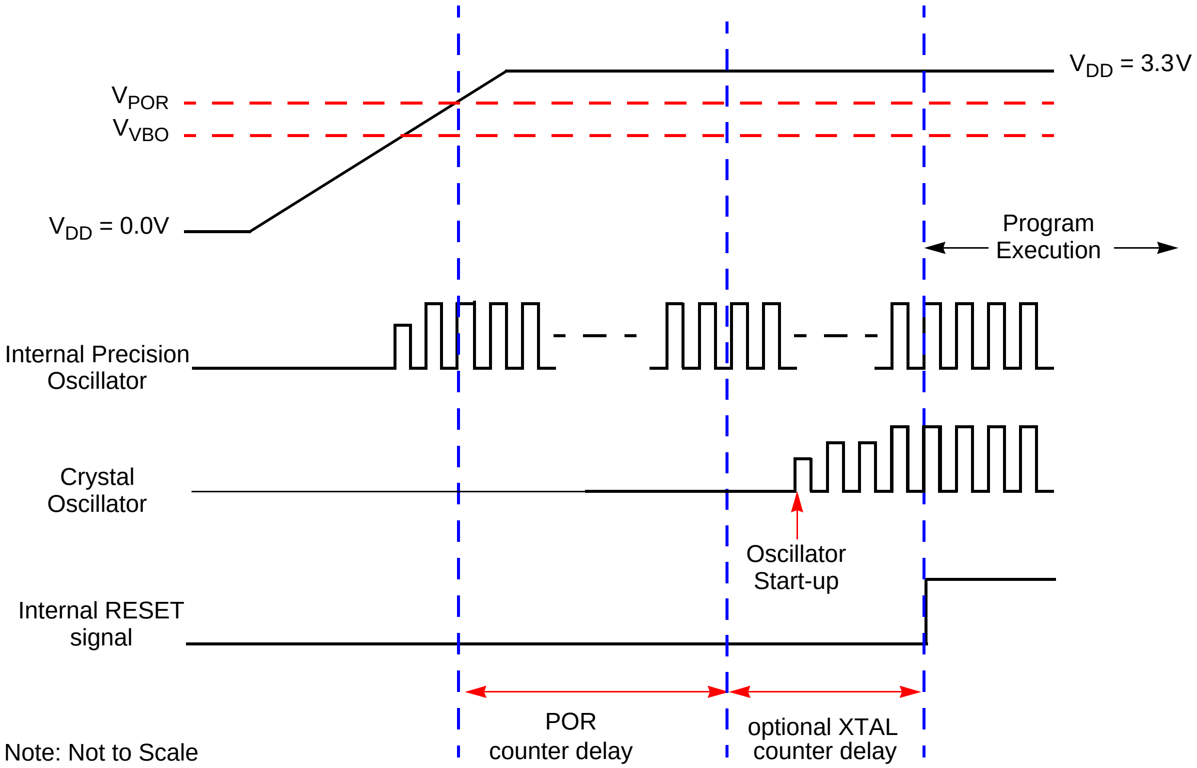


Figure 6. Power-On Reset Operation

Voltage Brown-Out Reset

The devices in the Z8 Encore! F083A Series provide low VBO protection. The VBO circuit forces the device to the Reset state, when the supply voltage drops below the VBO threshold voltage (unsafe level). While the supply voltage remains below the POR threshold voltage (V_{POR}), the VBO circuit holds the device in reset.

After the supply voltage exceeds the POR threshold voltage, the device progresses through a full system reset sequence, as described in the POR section. Following POR, the POR status bit in the reset status (RSTSTAT) Register is set to 1. Figure 7 displays VBO operation. For the VBO and POR threshold voltages (V_{VBO} and V_{POR}), see the Electrical Characteristics chapter on page 184.

The POR level is greater than the VBO level as determined by the specified hysteresis value. As a result, the devices is ensured to undergo a POR after recovering from a VBO condition.

The VBO circuit is either enabled or disabled during STOP Mode. Operations during STOP Mode are set by the VBO_AO Flash option bit. For more details about configuring VBO_AO, see the Flash Option Bits chapter on page 124.

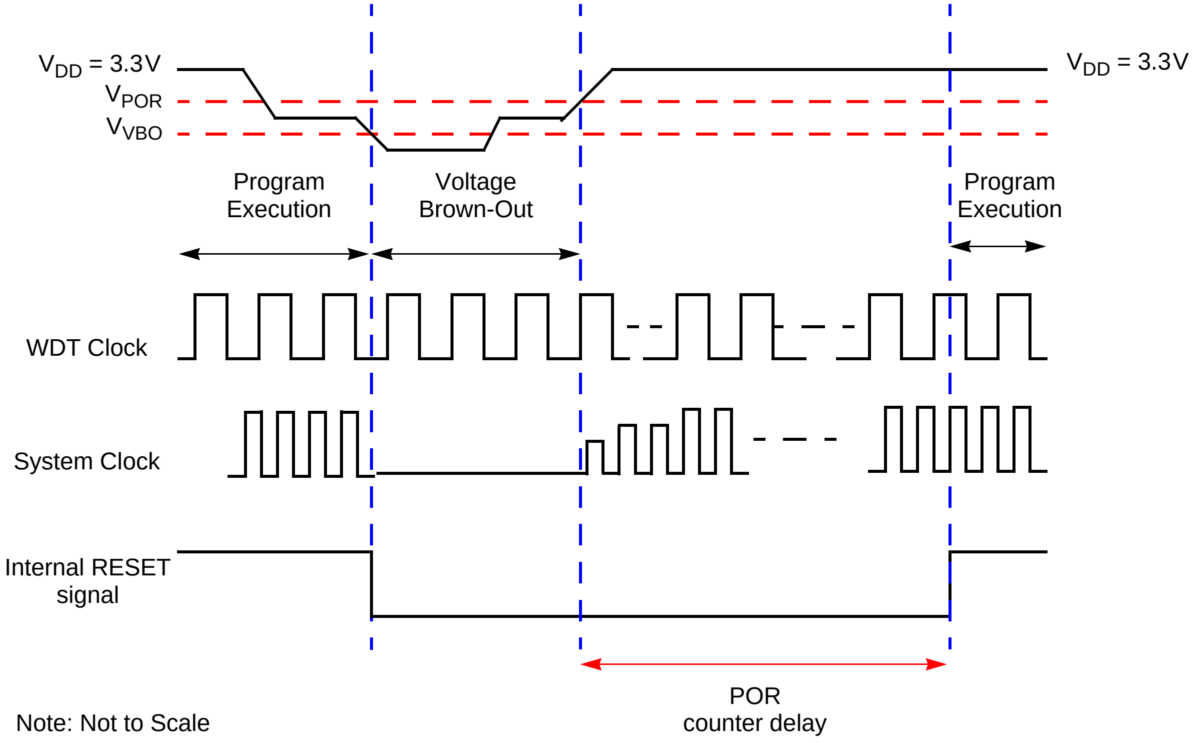


Figure 7. Voltage Brown-Out Reset Operation

Watchdog Timer Reset

If the device is in NORMAL or STOP Mode, the WDT initiates a system reset at time-out if the WDT_RES Flash option bit is programmed to 1. This is the unprogrammed state of the WDT_RES Flash option bit. If the bit is programmed to 0, it configures the WDT to cause an interrupt, not a system reset, at time-out. The WDT status bit in the Reset Status (RSTSTAT) Register is set to 1 to signify that the reset was initiated by the WDT.

External Reset Input

The $\overline{\text{RESET}}$ pin has a schmitt-triggered input and an internal pull-up resistor. Once the $\overline{\text{RESET}}$ pin is asserted for a minimum of four system clock cycles, the device progresses through the system reset sequence. Because of the possible asynchronicity of the system clock and reset signals, the required reset duration may be three or four clock periods. A reset pulse of three clock cycles in duration might trigger a reset and a reset pulse of four cycles in duration always triggers a reset.

While the $\overline{\text{RESET}}$ input pin is asserted low, the Z8 Encore! F083A Series devices remain in the Reset state. If the $\overline{\text{RESET}}$ pin is held low beyond the system reset time-out, the device exits the Reset state on the system clock rising edge following $\overline{\text{RESET}}$ pin deassertion. Following a system reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the Reset Status (RSTSTAT) Register is set to 1.

External Reset Indicator

During system reset or when enabled by the GPIO logic, the $\overline{\text{RESET}}$ pin functions as an open-drain (active low) RESET mode indicator in addition to the input functionality. This reset output feature allows an Z8 Encore! F083A Series device to reset other components to which it is connected, even if that reset is caused by internal sources such as POR, VBO or WDT events. See the Port A–D Control Registers section on page 41.

After an internal Reset event occurs, the internal circuitry begins driving the $\overline{\text{RESET}}$ pin low. The $\overline{\text{RESET}}$ pin is held low by the internal circuitry until the appropriate delay listed in Table 9 on page 22 has elapsed.

On-Chip Debugger Initiated Reset

A POR is initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The On-Chip Debugger block is not reset, but the remainder of the chip goes through a normal system reset. The RST bit automatically clears during the system reset. Following the system reset, the POR bit in the reset status (RSTSTAT) Register is set.

Stop Mode Recovery

The device enters the STOP Mode when the STOP instruction is executed by the eZ8 CPU. For more details about STOP Mode, see the Low-Power Modes chapter on page 30. During Stop Mode Recovery, the CPU is held in reset for about 66 IPO cycles if the crystal oscillator is disabled or about 5000 cycles if it is enabled.

Stop Mode Recovery does not affect the on-chip registers other than the Reset Status (RSTSTAT) Register and the Oscillator Control Register (OSCCTL). After any Stop Mode Recovery, the IPO is enabled and selected as the system clock. If another system clock source is required or IPO disabling is required, the Stop Mode Recovery code must reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

The eZ8 CPU fetches the reset vector at program memory addresses `0002H` and `0003H` and loads that value into the program counter. Program execution begins at the reset vector address. Following Stop Mode Recovery, the STOP bit in the Reset Status (RSTSTAT) Register is set to 1. Table 11 lists the Stop Mode Recovery sources and resulting actions. The following sections provide more detailed information about each of the Stop Mode Recovery sources.

Table 11. Stop Mode Recovery Sources and Resulting Action

Operating Mode	Stop Mode Recovery Source	Action
STOP Mode	WDT time-out when configured for Reset.	Stop Mode Recovery.
	WDT time-out when configured for interrupt.	Stop Mode Recovery followed by interrupt (if interrupts are enabled).
	Data transition on any GPIO port pin enabled as a Stop Mode Recovery source.	Stop Mode Recovery.
	Assertion of external $\overline{\text{RESET}}$ pin.	System reset.
	Debug pin driven Low.	System reset.

Stop Mode Recovery using Watchdog Timer Time-Out

If the WDT times out during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Reset Status (RSTSTAT) Register, the WDT and STOP bits are set to 1. If the WDT is configured to generate an interrupt upon time-out and the Z8 Encore! F083A Series device is configured to respond to interrupts, the eZ8 CPU services the WDT interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery using GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. If any GPIO pin is enabled as a Stop Mode Recovery source, a change in the input pin value (from high to low or from low to high) initiates Stop Mode Recovery. In the Reset Status (RSTSTAT) Register, the STOP bit is set to 1.

⚠ **Caution:** In STOP Mode, the GPIO port input data registers (PxIN) are disabled. The port input data registers record the port transition only if the signal stays on the port pin through the end of the Stop Mode Recovery delay. As a result, short pulses on the port pin initiates Stop Mode Recovery without being written to the port Input Data Register or without initiating an interrupt (if enabled for that pin).

Stop Mode Recovery Using the External $\overline{\text{RESET}}$ Pin

When the Z8 Encore! F083A Series device is in STOP Mode and the external $\overline{\text{RESET}}$ pin is driven low, a system reset occurs. Because of a glitch filter operating on the $\overline{\text{RESET}}$ pin, the low pulse must be greater than the minimum width specified about 12ns or it is ignored. The EXT bit in the Reset Status (RSTSTAT) Register is set.

Debug Pin Driven Low

Debug reset is initiated when the On-Chip Debugger detects any of the following error conditions on the DBG pin:

- Serial break (a minimum of nine continuous bits low)
- Framing error (received STOP bit is low)
- Transmit collision (OCD and host simultaneous transmission detected by the OCD)

When the Z8F083 is in STOP Mode, the debug reset will cause a system reset. The On-Chip Debugger block is not reset, but the remainder of the chip goes through a normal system reset. The POR bit in the reset (RSTSTAT) Register is set to 1.

Reset Register Definitions

The following sections define the Reset registers.

Reset Status Register

The Reset Status (RSTSTAT) Register detailed in Table 12 is a read-only register that indicates the source of the most recent Reset event, a Stop Mode Recovery event or a WDT time-out event. Reading this register resets the upper four bits to 0.

This register shares its address with the Watchdog Timer Control Register, which is write-only.

Table 12. Reset Status Register (RSTSTAT)

Bit	7	6	5	4	3	2	1	0
Field	POR	STOP	WDT	EXT	Reserved			
RESET	See Table 13			0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	FF0H							

Bit	Description
[7] POR	**Power-On Reset Indicator** This bit is set to 1 if a POR event occurs and is reset to 0, if a WDT time-out or Stop Mode Recovery occurs. Reading this register also resets this bit to 0.
[6] STOP	**Stop Mode Recovery Indicator** This bit is set to 1 if a Stop Mode Recovery occurs. If the STOP and WDT bits are both set to 1, the Stop Mode Recovery occurs because of a WDT time-out. If the STOP bit is 1 and the WDT bit is 0, the Stop Mode Recovery is not caused by a WDT time-out. This bit is reset by a POR or a WDT time-out that occurred while not in STOP Mode. Reading this register also resets this bit.
[5] WDT	**Watchdog Timer time-out Indicator** This bit is set to 1 if a WDT time-out occurs. A POR resets this pin. A Stop Mode Recovery from a change in an input pin also resets this bit. Reading this register resets this bit. This read must occur before clearing the WDT interrupt.
[4] EXT	**External reset Indicator** If this bit is set to 1, a reset initiated by the external $\overline{\text{RESET}}$ pin occurred. A POR or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit.
[3:0]	**Reserved** These bits are reserved and must be programmed to 0000.

Table 13. POR Indicator Values

Reset or Stop Mode Recovery Event	POR	STOP	WDT	EXT
Power-On Reset	1	0	0	0
Reset using $\overline{\text{RESET}}$ pin assertion	0	0	0	1
Reset using WDT time-out	0	0	1	0
Reset using the On-Chip Debugger (OCTCTL[1] set to 1)	1	0	0	0
Reset from STOP Mode using DBG pin driven Low	1	0	0	0
Stop Mode Recovery using GPIO pin transition	0	1	0	0
Stop Mode Recovery using WDT time-out	0	1	1	0

Low-Power Modes

The Z8 Encore! F083A Series products contain power saving features. The highest level of power reduction is provided by the STOP Mode. The next level of power reduction is provided by the HALT Mode.

Further power savings are implemented by disabling the individual peripheral blocks while in NORMAL Mode.

Caution: The user must not enable the pull-up register bits for unused GPIO pins, because these ports output by default to V_{SS}. Unused GPIO pins include those missing on 20-pin packages and ADC-enabled 28-pin packages.

STOP Mode

Executing the eZ8 CPU’s STOP instruction places the device into STOP Mode. In STOP Mode, the operating characteristics are:

- Primary crystal oscillator and internal precision oscillator are stopped; XIN and XOUT (if previously enabled) are disabled and PA0/PA1 revert to the states programmed by the GPIO registers
- System clock is stopped
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- WDT’s internal RC oscillator continues to operate if enabled by the Oscillator Control Register
- If enabled, the WDT logic continues to operate
- If enabled for operation in STOP Mode by the associated Flash option bit, the VBO protection circuit continues to operate
- All other on-chip peripherals are idle

To minimize the current in STOP Mode, all GPIO pins that are configured as digital inputs must be driven to V_{DD} when the pull-up register bit is enabled or to one of power rail (V_{DD} or GND) when the pull-up register bit is disabled. The device is brought out of STOP Mode using Stop Mode Recovery. For more information about Stop Mode Recovery, see the Reset and Stop Mode Recovery chapter on page 21.

HALT Mode

Executing the eZ8 CPU HALT instruction places the device into HALT Mode. In HALT Mode, the operating characteristics are:

- Primary oscillator is enabled and continues to operate
- System clock is enabled and continues to operate
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- WDT's internal RC oscillator continues to operate
- If enabled, the WDT continues to operate
- All other on-chip peripherals continue to operate

The eZ8 CPU is brought out of HALT Mode by any one of the following operations:

- Interrupt
- WDT time-out (interrupt or reset)
- POR
- VBO reset
- External $\overline{\text{RESET}}$ pin assertion

To minimize current in HALT Mode, all GPIO pins that are configured as digital inputs must be driven to V_{DD} when pull-up register bit is enabled or to one of power rail (V_{DD} or GND) when pull-up register bit is disabled.

Peripheral Level Power Control

In addition to the STOP and HALT modes, it is possible to disable each peripheral on each of the Z8 Encore! F083A Series devices. Disabling a given peripheral minimizes its power consumption.

Power Control Register Definitions

The following sections describe the power control registers.

Power Control Register 0

Each bit in this register disables a peripheral block, either by gating its system clock input or by removing power from the block.

Note: This register is only reset during a POR sequence. Other system Reset events do not affect it.

Table 14. Power Control Register 0 (PWRCTL0)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved			VBO	Reserved	Reserved	COMP	Reserved
RESET	1	0	0	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F80H							

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 000.
[4] VBO	**Voltage Brown-Out detector Disable** This bit is only effect when the VBO_AO Flash option bit is disabled. In STOP Mode, VBO is always disabled when the VBO_AO Flash option bit is disabled. For VBO_AO Flash option bit function, see the Flash Option Bits chapter on page 124. 0 = VBO enabled. 1 = VBO disabled.
[3]	**Reserved** This bit is reserved and must be programmed to 0.
[2]	**Reserved** This bit is reserved and must be programmed to 0.
[1] COMP	**Comparator Disable** 0 = Comparator is enabled. 1 = Comparator is disabled.
[0]	**Reserved** This bit is reserved and must be programmed to 0.

General Purpose Input/Output

The Z8 Encore! F083A Series products support a maximum of 23 port pins (Port A–D) for general purpose input/output (GPIO) operations. Each port contains control and data registers. The GPIO control registers determine data direction, open-drain, output drive current, programmable pull-ups, Stop Mode Recovery functionality and alternate pin functions. Each port pin is individually programmable. In addition, the Port C pins are capable of direct LED drive at programmable drive strengths.

GPIO Port Availability by Device

Table 15 lists the port pins available with each device and package type.

Table 15. Port Availability by Device and Package Type

Devices	Package	10-Bit ADC	Port A	Port B	Port C	Port D	Total I/O
Z8F083A, Z8F043A	20-pin	Yes	[7:0]	[3:0]	[3:0]	[0]	17
Z8F083A, Z8F043A	28-pin	Yes	[7:0]	[5:0]	[7:0]	[0]	23

Note: 20-pin and 28-pin and 10-bit ADC Enabled or Disabled can be selected via the option bits.

Architecture

Figure 8 displays a simplified block diagram of a GPIO port pin. In this figure, the ability to accommodate alternate functions and variable port current drive strength is not displayed.

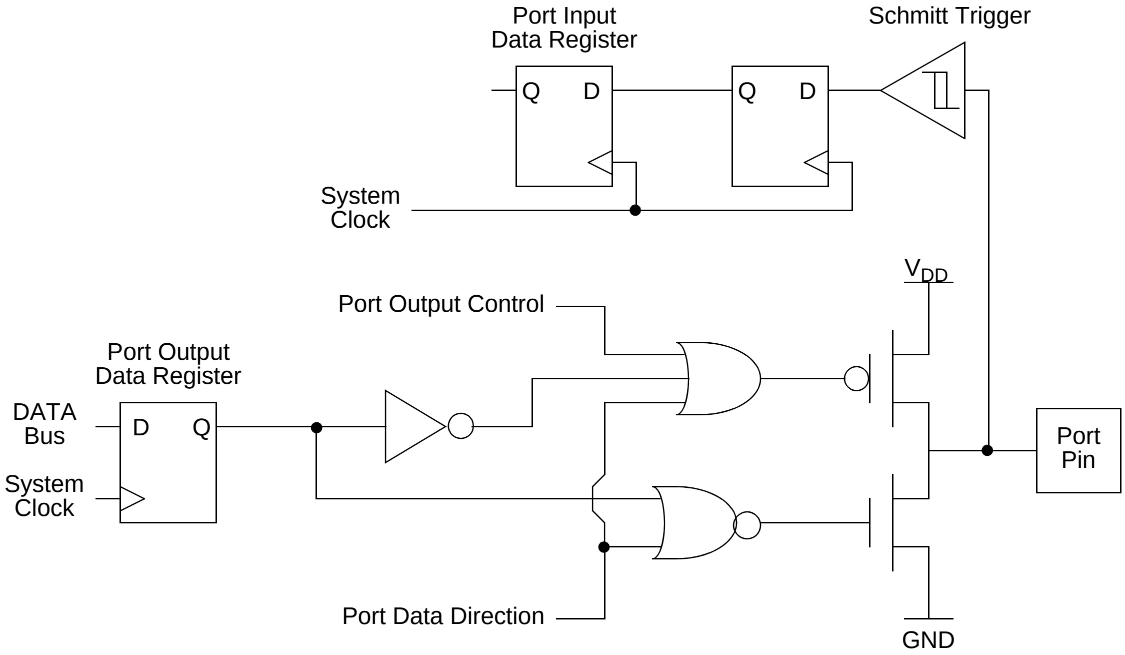


Figure 8. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many of the GPIO port pins are used for general purpose input/output and access to on-chip peripheral functions such as the timers and serial communication devices. The Port A–D alternate function subregisters configure these pins for either GPIO or Alternate Function operation. When a pin is configured for alternate function, control of the port pin direction (input/output) is passed from the Port A–D data direction registers to the alternate function assigned to this pin. Table 16 on page 36 lists the alternate functions possible with each port pin. The alternate function associated at a pin is defined through alternate function subregisters AFS1 and AFS2.

The crystal oscillator functionality is not controlled by the GPIO block. When the crystal oscillator is enabled in the oscillator control block, the GPIO functionality of PA0 and PA1 is overridden. In that case, pins PA0 and PA1 functions as input and output for the crystal oscillator.

PA0 and PA6 contain two different timer functions, a timer input and a complementary timer output. Both of these functions require the same GPIO configuration, the selection between the two is based on the TIMER mode. For more details, see the Timers chapter on page 69.

Direct LED Drive

The Port C pins provide a sinked current output, capable of driving an LED without requiring an external resistor. The output sinks current at programmable levels of 3mA, 7mA, 13mA and 20mA. This mode is enabled through the LED control registers.

For correct function, the LED anode must be connected to V_{DD} and the cathode to the GPIO pin.

Using all Port C pins in LED drive mode with maximum current may result in excessive total current. For details about maximum total current for the applicable package, see the Electrical Characteristics chapter on page 184.

Shared Reset Pin

On the 20- and 28-pin devices, the Port D0 pin shares function with a bidirectional reset pin. Unlike all other I/O pins, this pin does not default to GPIO function on power-up. This pin acts as a bidirectional input/output open-drain reset with an internal pull-up until the user software reconfigures it as a GPIO PD0. When in GPIO mode, the Port D0 pin functions as output only, and must be configured as an output. PD0 supports the high drive feature, but not the stop-mode recovery feature.

Crystal Oscillator Override

For systems using a crystal oscillator, the pins PA0 and PA1 are connected to the crystal. When the crystal oscillator is enabled, the GPIO settings are overridden and PA0 and PA1 are disabled. See the Oscillator Control Register Definitions section on page 154.

5V Tolerance

In the 20- and 28-pin versions of this device, any pin, which shares functionality with an ADC, crystal or comparator port is not 5V-tolerant, including PA[1:0], PB[5:0] and PC[2:0]. All other signal pins are 5V-tolerant and safely handles inputs higher than V_{DD} even with the pull-ups enabled, but with excess power consumption on pull-up resistor.

External Clock Setup

For systems using an external TTL drive, PB3 is the clock source for 20- and 28-pin devices. In this case, configure PB3 for alternate function CLKIN. Write to the Oscillator Control Register (see 154) to select the PB3 as the system clock.

Table 16. Port Alternate Function Mapping

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A	PA0	T0IN/$\overline{\text{T0OUT}}$	Timer 0 input/Timer 0 output complement	N/A
		Reserved		
	PA1	T0OUT	Timer 0 output	
		Reserved		
	PA2	Reserved	Reserved	
		Reserved		
	PA3	Reserved	Reserved	
		Reserved		
	PA4	Reserved	Reserved	
		Reserved		
	PA5	Reserved	Reserved	
		Reserved		
	PA6	T1IN/$\overline{\text{T1OUT}}$	Timer 1 input/Timer 1 output complement	
		Reserved		
	PA7	T1OUT	Timer 1 output	
		Reserved		

Note: Because there is only a single alternate function for each Port A and Port D (PD0) pin, the Alternate Function Set registers are not implemented for Port A and Port D (PD0). Enabling alternate function selections (as described in the Port A–D Alternate Function Subregisters section on page 42) automatically enables the associated alternate function.

Table 16. Port Alternate Function Mapping (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port B	PB0	Reserved		AFS1[0]: 0
		ANA0	ADC analog input	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1	ADC analog input	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2	ADC analog input	AFS1[2]: 1
	PB3	CLKIN	External input clock	AFS1[3]: 0
		ANA3	ADC analog input	AFS1[3]: 1
	PB4	Reserved		AFS1[4]: 0
		ANA7	ADC analog input	AFS1[4]: 1
	PB5	Reserved		AFS1[5]: 0
		VREF	ADC reference voltage	AFS1[5]: 1
	PB6	Reserved		AFS1[6]: 0
		Reserved		AFS1[6]: 1
	PB7	Reserved		AFS1[7]: 0
		Reserved		AFS1[7]: 1

Note: Because there is only a single alternate function for each Port A and Port D (PD0) pin, the Alternate Function Set registers are not implemented for Port A and Port D (PD0). Enabling alternate function selections (as described in the Port A–D Alternate Function Subregisters section on page 42) automatically enables the associated alternate function.

Table 16. Port Alternate Function Mapping (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port C	PC0	Reserved		AFS1[0]: 0
		ANA4/CINP	ADC or comparator input	AFS1[0]: 1
	PC1	Reserved		AFS1[1]: 0
		ANA5/CINN	ADC or comparator input	AFS1[1]: 1
	PC2	Reserved		AFS1[2]: 0
		ANA6	ADC analog input	AFS1[2]: 1
	PC3	COUT	Comparator output	AFS1[3]: 0
		Reserved		AFS1[3]: 1
	PC4	Reserved		AFS1[4]: 0
				AFS1[4]: 1
	PC5	Reserved		AFS1[5]: 0
				AFS1[5]: 1
	PC6	Reserved		AFS1[6]: 0
				AFS1[6]: 1
	PC7	Reserved		AFS1[7]: 0
				AFS1[7]: 1
Port D	PD0	RESET	Default to be Reset function	N/A

Note: Because there is only a single alternate function for each Port A and Port D (PD0) pin, the Alternate Function Set registers are not implemented for Port A and Port D (PD0). Enabling alternate function selections (as described in the Port A–D Alternate Function Subregisters section on page 42) automatically enables the associated alternate function.

GPIO Interrupts

Many of the GPIO port pins are used as interrupt sources. Some port pins are configured to generate an interrupt request on either the rising edge or falling edge of the input pin signal. Other port pin interrupt sources, generate an interrupt when any edge occurs (both rising and falling). For more details about interrupts using the GPIO pins, see the Interrupt Controller chapter on page 54.

GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data and output data. Table 17 lists these port registers. Use the Port A–D address and control registers together to provide access to subregisters for port configuration and control.

Table 17. GPIO Port Registers and Subregisters

Port Register Mnemonic	Port Register Name
P*x*ADDR	Port A–D Address Register (selects subregisters)
P*x*CTL	Port A–D Control Register (provides access to subregisters)
P*x*IN	Port A–D Input Data Register
P*x*OUT	Port A–D Output Data Register
Port Subregister Mnemonic	**Port Register Name**
P*x*DD	Data Direction
P*x*AF	Alternate Function
P*x*OC	Output Control (open-drain)
P*x*HDE	High Drive Enable
P*x*SMRE	Stop Mode Recovery source enable
P*x*PUE	Pull-up Enable
PxAFS1	Alternate Function Set 1
PxAFS2	Alternate Function Set 2

Port A–D Address Registers

The Port A–D Address registers select the GPIO port functionality by accessing the Port A–D Control registers. The Port A–D Address and Control registers combine to provide access to all GPIO port controls; see Tables 18 and 19.

Table 18. Port A–D GPIO Address Registers (P*x*ADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD0H, FD4H, FD8H, FDCH							

Bit	Description
[7:0] PADDR	**Port Address** The port address selects one of the subregisters accessible through the Port Control Register.

Table 19. Port Control Subregister Access

PADDR[7:0]	Port Control Subregister Accessible Using the Port A–D Control Registers
00H	No function. Provides some protection against accidental port reconfiguration.
01H	Data Direction
02H	Alternate function
03H	Output control (Open-Drain)
04H	High drive enable
05H	Stop Mode Recovery source enable.
06H	Pull-up enable
07H	Alternate function set 1
08H	Alternate function set 2
09H–FFH	No function

Port A–D Control Registers

The Port A–D control registers, shown in Table 20, set GPIO port operation. The value in the corresponding Port A–D Address Register determines which subregister is read from or written to by a Port A–D Control Register transaction.

Table 20. Port A–D Control Registers (P*x*CTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD1H, FD5H, FD9H, FDDH							

Bit	Description
[7:0] PCTL	**Port Control** The Port Control Register provides access to all subregisters that configure GPIO port operation.

Port A–D Data Direction Subregisters

The Port A–D Data Direction Subregister, shown in Table 21, is accessed through the Port A–D Control Register by writing `01H` to the Port A–D Address Register.

Table 21. Port A–D Data Direction Subregisters (PxDD)

Bit	7	6	5	4	3	2	1	0
Field	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 01H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] DDx	**Data Direction** These bits control the direction of the associated port pin. Port Alternate function operation overrides the Data Direction Register setting. 0 = Output. Data in the Port A–D Output Data Register is driven onto the port pin. 1 = Input. The port pin is sampled and the value written into the Port A–D Input Data Register. The output driver is tristated.
Note:	x indicates the specific GPIO port pin number (7–0).

Port A–D Alternate Function Subregisters

The Port A–D alternate function subregister is accessed through the Port A–D Control Register by writing `02H` to the Port A–D Address Register. See Table 22 on page 42. The Port A–D alternate function subregisters enable the alternate function selection on pins. If disabled, the pins functions as GPIO. If enabled, select one of four alternate functions using alternate function set subregisters 1 and 2 as described in the **the** Port A–D Alternate Function Set 1 Subregisters **section on page 47** and **the** Port A–D Alternate Function Set 2 Subregisters **section on page 48.** To determine the alternate functions associated with each port pin, see **the** GPIO Alternate Functions section on page 34.

Caution: Do not enable alternate functions for GPIO port pins for which there are no associated alternate functions. Failure to follow this guideline can result in unpredictable operation.

Table 22. Port A–D Alternate Function Subregisters (PxAF)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
RESET	00H (Ports A–C); 01H (Port D)							
R/W	R/W							
Address	If 02H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	**Description**
[7:0] AFx	**Port Alternate Function Enabled** 0 = The port pin is in NORMAL Mode and the `DDx` bit in the Port A–D Data Direction Subregister determines the direction of the pin. 1 = The alternate function selected through alternate function set subregisters is enabled. Port pin operation is controlled by the alternate function.
Note:	x indicates the specific GPIO port pin number (7–0).

Port A–D Output Control Subregisters

The Port A–D output control subregister is accessed through the Port A–D Control Register by writing `03H` to the Port A–D Address Register. See Table 23. Setting the bits in the Port A–D output control subregisters to 1, configures the specified port pins for open-drain operation. These subregisters affect the pins directly and, as a result, alternate functions are also affected.

Table 23. Port A–D Output Control Subregisters (PxOC)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 03H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	**Description**
[7:0] POCx	**Port Output Control** These bits function independently of the alternate function bit and always disable the drains, if set to 1. 0 = The drains are enabled for any output mode (unless overridden by the alternate function). 1 = The drain of the associated pin is disabled (OPEN-DRAIN mode).

Note: x indicates the specific GPIO port pin number (7–0).

Port A–D High Drive Enable Subregisters

The Port A–D High Drive Enable Subregister, shown in Table 24, is accessed through the Port A–D Control Register by writing `04H` to the Port A–D Address Register. Setting the bits in the Port A–D High Drive Enable subregisters to 1 configures the specified port pins for high-output current drive operation. The Port A–D High Drive Enable Subregister affects the pins directly and, as a result, alternate functions are also affected.

Table 24. Port A–D High Drive Enable Subregisters (PxHDE)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PHDE7	PHDE6	PHDE5	PHDE4	PHDE3	PHDE2	PHDE1	PHDE0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 04H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	**Description**
[7:0] PHDEx	**Port High Drive Enable** 0 = The port pin is configured for standard output current drive. 1 = The port pin is configured for high output current drive.

Note: x indicates the specific GPIO port pin number (7–0).

Port A–D Stop Mode Recovery Source Enable Subregisters

The Port A–D Stop Mode Recovery Source Enable Subregister, shown in Table 25, is accessed through the Port A–D Control Register by writing `05H` to the Port A–D Address Register. Setting the bits in the Port A–D Stop Mode Recovery Source Enable subregisters to1, configures the specified port pins as a Stop Mode Recovery source. During STOP Mode, any logic transition on a port pin enabled as a Stop Mode Recovery source initiates Stop Mode Recovery.

Table 25. Port A–D Stop Mode Recovery Source Enable Subregisters (PxSMRE)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PSMRE7	PSMRE6	PSMRE5	PSMRE4	PSMRE3	PSMRE2	PSMRE1	PSMRE0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 05H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	**Description**
[7:0] PSMREx	**Port Stop Mode Recovery Source Enable** 0 = The port pin is not configured as a Stop Mode Recovery source. Transitions on this pin during STOP Mode do not initiate Stop Mode Recovery. 1 = The port pin is configured as a Stop Mode Recovery source. Any logic transition on this pin during STOP Mode initiates Stop Mode Recovery.
Note: x indicates the specific GPIO port pin number (7–0).	

Port A–D Pull-up Enable Subregisters

The Port A–D pull-up enable subregister is accessed through the Port A–D Control Register by writing `06H` to the Port A–D Address Register. See Table 26. Setting the bits in the Port A–D pull-up enable subregisters, enables a weak internal resistive pull-up on the specified port pins.

Table 26. Port A–D Pull-Up Enable Subregisters (PxPUE)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PPUE7	PPUE6	PPUE5	PPUE4	PPUE3	PPUE2	PPUE1	PPUE0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 06H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] PPUEx	**Port Pull-Up Enable** 0 = The weak pull-up on the port pin is disabled. 1 = The weak pull-up on the port pin is enabled.
Note: x indicates the specific GPIO port pin number (7–0).	

Port A–D Alternate Function Set 1 Subregisters

The Port A–D Alternate Function Set 1 Subregister, shown in Table 27, is accessed through the Port A–D Control Register by writing `07H` to the Port A–D Address Register. The Alternate Function Set 1 subregisters select the alternate function available at a port pin. Alternate functions selected by setting or clearing bits of this register are defined in the Port Alternate Function Mapping table on page 36.

Note: Alternate function selection on the port pins must also be enabled, as described in the Port A–D Alternate Function Subregisters section on page 42.

Table 27. Port A–D Alternate Function Set 1 Subregisters (PxAFS1)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PAFS17	PAFS16	PAFS15	PAFS14	PAFS13	PAFS12	PAFS11	PAFS10
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 07H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	**Description**
[7:0] PAFS1x	**Port Alternate Function Set 1** 0 = Port alternate function selected as defined in the GPIO Alternate Functions section; see Table 16 on page 36. 1 = Port alternate function selected as defined in the GPIO Alternate Functions section; see Table 16 on page 36.

Note: x indicates the specific GPIO port pin number (7–0).

Port A–D Alternate Function Set 2 Subregisters

The Port A–D Alternate Function Set 2 Subregister, shown in Table 28, is accessed through the Port A–D Control Register by writing `08H` to the Port A–D Address Register. The Alternate Function Set 2 subregisters select the alternate function available at a port pin. Alternate functions selected by setting or clearing bits of this register are defined in the Port Alternate Function Mapping table on page 36.

Note: Alternate function selection on port pins must also be enabled, as described in the Port A–D Alternate Function Subregisters section on page 42.

Table 28. Port A–D Alternate Function Set 2 Subregisters (PxAFS2)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PAFS27	PAFS26	PAFS25	PAFS24	PAFS23	PAFS22	PAFS21	PAFS20
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 08H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	**Description**
[7:0] PAFS2x	**Port Alternate Function Set 2** 0 = Port alternate function selected, as defined in Table 16 on page 36. 1 = Port alternate function selected, as defined in Table 16 on page 36.

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Input Data Registers

Reading from the Port A–C Input Data registers, shown in Table 29, returns the sampled values from the corresponding port pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8- and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

Table 29. Port A–C Input Data Registers (PxIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	FD2H, FD6H, FDAH							

Bit	Description
[7:0] PxIN	**Port Input Data** Sampled data from the corresponding port pin input. 0 = Input data is logical 0 (Low). 1 = Input data is logical 1 (High).

Note: x indicates the specific GPIO port pin number (7–0).

Port A–D Output Data Register

The Port A–D Output Data Register, shown in Table 30, controls the output data to the pins.

Table 30. Port A–D Output Data Register (P*x*OUT)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD3H, FD7H, FDBH, FDFH							

Bit	**Description**
[7:0] PxOUT	**Port Output Data** These bits contain the data to be driven to the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation. 0 = Drive a logical 0 (Low). 1 = Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding port output Control Register bit to 1.

Note: x indicates the specific GPIO port pin number (7–0).

LED Drive Enable Register

The LED Drive Enable Register, shown in Table 31, activates the controlled current drive. The Alternate Function Register has no control over the LED function; therefore, setting the Alternate Function Register to select the LED function is not required. LEDEN bits [7:0] correspond to Port C bits [7:0], respectively.

Table 31. LED Drive Enable (LEDEN)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	LEDEN[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F82H							

Bit	**Description**
[7:0] LEDEN	**LED Drive Enable** These bits determine, which Port C pins are connected to an internal current sink. 0 = Tristate the Port C pin. 1 = Connect controlled current sink to the Port C pin.

LED Drive Level High Register

The LED Drive Level High Register, shown in Table 32, contains two control bits for each Port C pin. These two bits selects one of four programmable current drive levels for each Port C pin. Each pin is individually programmable.

Table 32. LED Drive Level High Register (LEDLVLH)

Bit	7	6	5	4	3	2	1	0
Field	LEDLVLH[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F83H							

Bit	Description
[7:0] LEDLVLH	**LED Level High Bits** {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA. 01 = 7mA. 10 = 13mA. 11 = 20mA.

LED Drive Level Low Register

The LED Drive Level Low Register, shown in Table 33, contains two control bits for each Port C pin. These two bits selects one of four programmable current drive levels for each Port C pin. Each pin is individually programmable.

Table 33. LED Drive Level Low Register (LEDLVLL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	LEDLVLL[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F84H							

Bit	**Description**
[7:0] LEDLVLL	**LED Level Low Bits** {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA. 01 = 7mA. 10 = 13mA. 11 = 20mA.

Interrupt Controller

The Interrupt Controller on the Z8 Encore! F083A Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of the Interrupt Controller include the following:

- Seventeen interrupt sources using sixteen unique interrupt vectors
  - Twelve GPIO port pin interrupt sources
  - Five on-chip peripheral interrupt sources (the comparator output interrupt shares one interrupt vector with PA6)
- Flexible GPIO interrupts
  - Eight selectable rising- and falling-edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually-programmable interrupt priority
- Watchdog Timer is configured to generate an interrupt

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually, this interrupt service routine is involved with the exchange of data, status information or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the Interrupt Controller has no effect on operation. For more information regarding interrupt servicing by the eZ8 CPU, refer to the eZ8 CPU Core User Manual (UM0128), available for download on www.zilog.com.

Interrupt Vector Listing

Table 34 lists the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even program memory address and the least-significant byte (LSB) at the odd program memory address.

Note: Some port interrupts are not available on the 20-pin and 28-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

Table 34. Trap and Interrupt Vectors in Order of Priority

Priority	Program Memory Vector Address	Interrupt or Trap Source
Highest	0002H	Reset (not an interrupt)
	0004H	Watchdog Timer (see Watchdog Timer chapter)
	003AH	Primary oscillator fail trap (not an interrupt)
	003CH	Watchdog oscillator fail trap (not an interrupt)
	0006H	Illegal instruction trap (not an interrupt)
	0008H	Reserved
	000AH	Timer 1
	000CH	Timer 0
	000EH	Reserved
	0010H	Reserved
	0012H	Reserved
	0014H	Reserved
	0016H	ADC
	0018H	Port A7, selectable rising or falling input edge
	001AH	Port A6, selectable rising or falling input edge or Comparator Output
	001CH	Port A5, selectable rising or falling input edge
	001EH	Port A4, selectable rising or falling input edge
	0020H	Port A3, selectable rising or falling input edge
	0022H	Port A2, selectable rising or falling input edge
	0024H	Port A1, selectable rising or falling input edge
	0026H	Port A0, selectable rising or falling input edge
	0028H	Reserved
	002AH	Reserved
	002CH	Reserved
	002EH	Reserved
	0030H	Port C3, both input edges
	0032H	Port C2, both input edges
	0034H	Port C1, both input edges
	0036H	Port C0, both input edges
Lowest	0038H	Reserved

Architecture

Figure 9 displays the Interrupt Controller block diagram.

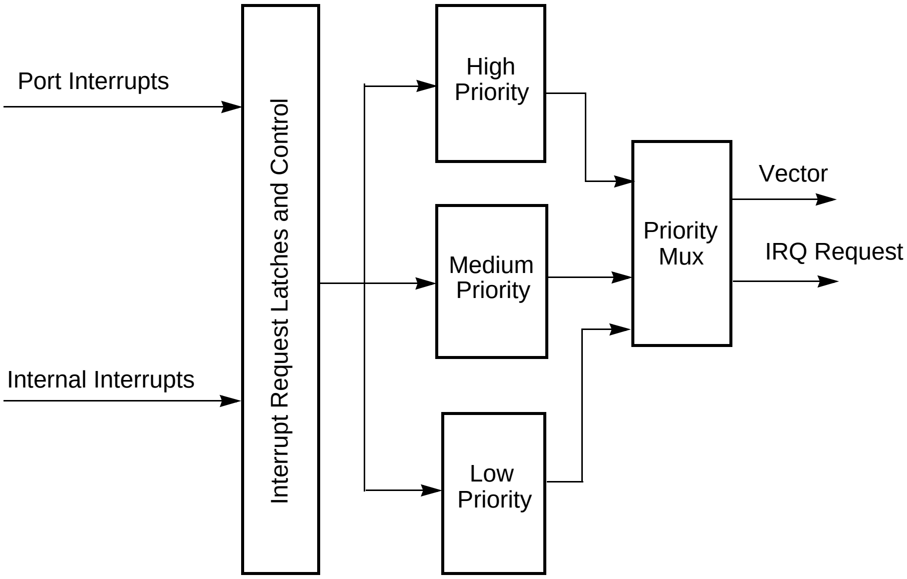


Figure 9. Interrupt Controller Block Diagram

Operation

This section describes the operational aspects of the following functions.

Master Interrupt Enable: see page 56

Interrupt Vectors and Priority: see page 57

Interrupt Assertion: see page 57

Software Interrupt Assertion: see page 58

Master Interrupt Enable

The master interrupt enable bit (`IRQE`) in the Interrupt Control Register globally enables and disables the interrupts.

Interrupts are globally enabled by any of the following actions:

- Execution of an enable interrupt (EI) instruction
- Execution of an return from interrupt (IRET) instruction

- Writing 1 to the IRQE bit in the Interrupt Control Register

Interrupts are globally disabled by any of the following actions:

- Execution of a DI (disable interrupt) instruction
- eZ8 CPU acknowledgement of an interrupt service request from the Interrupt Controller
- Writing a 0 to the IRQE bit in the Interrupt Control Register
- Reset
- Execution of a trap instruction
- Illegal instruction trap
- Primary oscillator fail trap
- Watchdog oscillator fail trap

Interrupt Vectors and Priority

The Interrupt Controller supports three levels of interrupt priority. Level 3 is the highest priority, level 2 is the second highest priority and level 1 is the lowest priority. If all of the interrupts are enabled with identical interrupt priority (all as level 2 interrupts, for example), the interrupt priority is assigned from highest to lowest as specified in Table 34 on page 55. Level 3 interrupts are always assigned higher priority than level 2 interrupts and level 2 interrupts are assigned higher priority than level 1 interrupts. Within each interrupt priority level (level 1, level 2 or level 3), priority is assigned as specified in Table 34. Reset, Watchdog Timer interrupts (if enabled), primary oscillator fail traps, Watchdog oscillator fail traps and illegal instruction traps always have highest (level 3) priority.

Interrupt Assertion

Interrupt sources assert their interrupt requests for only a single system clock period (single pulse). When the interrupt request is acknowledged by the eZ8 CPU, the corresponding bit in the interrupt request register is cleared. Writing 0 to the corresponding bit in the interrupt request register clears the interrupt request.

⚠ **Caution:** Zilog recommends not using a coding style that clears bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 1, which follows.

Example 1. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
AND r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 2 to clear bits in the Interrupt Request 0 Register:

Example 2. A good coding style that avoids lost interrupt requests:

```
ANDX IRQ0, MASK
```

Software Interrupt Assertion

Program code generates interrupts directly. Writing 1 to the correct bit in the interrupt request register triggers an interrupt (assuming that interrupt is enabled). When the interrupt request is acknowledged by the eZ8 CPU, the bit in the interrupt request register is automatically cleared to 0.

⚠ **Caution:** Zilog recommends not using a coding style to generate software interrupts by setting bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 3, which follows.

Example 3. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 4 to set bits in the Interrupt Request registers:

Example 4. A good coding style that avoids lost interrupt requests:

```
ORX IRQ0, MASK
```

Interrupt Control Register Definitions

The Interrupt Control registers enable individual interrupts, set interrupt priorities and indicate interrupt requests for all of the interrupts other than the Watchdog Timer interrupt, the primary oscillator fail trap and the Watchdog oscillator fail trap interrupts.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register, shown in Table 35, stores the interrupt requests for both vectored and polled interrupts. When a request is sent to the Interrupt Controller, the corresponding bit in the IRQ0 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the Interrupt Controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU reads the Interrupt Request 0 Register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 0 Register (IRQ0)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	T1I	T0I	Reserved				ADCI
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC0H							

Bit	**Description**
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1I	**Timer 1 Interrupt Request** 0 = No interrupt request is pending for Timer 1. 1 = An interrupt request from timer 1 is awaiting service.
[5] T0I	**Timer 0 Interrupt Request** 0 = No interrupt request is pending for Timer 0. 1 = An interrupt request from timer 0 is awaiting service.
[4:1]	**Reserved** These bits are reserved and must be programmed to 0000.
[0] ADCI	**ADC Interrupt Request** 0 = No interrupt request is pending for the ADC. 1 = An interrupt request from the ADC is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) Register, shown in Table 36, stores interrupt requests for both vectored and polled interrupts. When a request is sent to the Interrupt Controller, the corresponding bit in the IRQ1 Register becomes 1. If interrupts are globally enabled (i.e., vectored interrupts), the Interrupt Controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (i.e., polled interrupts), the eZ8 CPU reads the Interrupt Request 1 Register to determine if any interrupt requests are pending.

Table 36. Interrupt Request 1 Register (IRQ1)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PA7I	PA6CI	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC3H							

Bit	**Description**
[7] PA7I	**Port A7** 0 = No interrupt request is pending for GPIO Port A. 1 = An interrupt request from GPIO Port A.
[6] PA6CI	**Port A6 or Comparator Interrupt Request** 0 = No interrupt request is pending for GPIO Port A or comparator. 1 = An interrupt request from GPIO Port A or comparator.
[5] PAxI	**Port A Pin *x* Interrupt Request** 0 = No interrupt request is pending for GPIO Port A pin *x*. 1 = An interrupt request from GPIO Port A pin *x* is awaiting service.
Note: x indicates the specific GPIO port pin number (5–0).	

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) Register, shown in Table 37, stores interrupt requests for both vectored and polled interrupts. When a request is sent to the Interrupt Controller, the corresponding bit in the IRQ2 Register becomes 1. If interrupts are globally enabled (i.e., vectored interrupts), the Interrupt Controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (i.e., polled interrupts), the eZ8 CPU reads the Interrupt Request 2 Register to determine if any interrupt requests are pending.

Table 37. Interrupt Request 2 Register (IRQ2)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				PC3I	PC2I	PC1I	PC0I
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC6H							

Bit	**Description**
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:0] PCxI	**Port C pin *x* Interrupt Request** 0 = No interrupt request is pending for GPIO Port C pin *x*. 1 = An interrupt request from GPIO Port C pin *x* is awaiting service.
Note: x indicates the specific GPIO port pin number (3–0).	

IRQ0 Enable High and Low Bit Registers

Table 38 indicates priority control for the IRQ0 Register. The IRQ0 Enable High and Low Bit registers, shown in Tables 39 and 40, form a priority-encoded enabling service for interrupts in the Interrupt Request 0 Register. Priority is generated by setting the appropriate bits in each register.

Table 38. IRQ0 Enable and Priority Encoding

IRQ0ENH[*x*]	**IRQ0ENL[*x*]**	**Priority**	**Description**
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High

Note: x indicates register bits 7–0.

Table 39. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	T1ENH	T0ENH	Reserved				ADCENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC1H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1ENH	**Timer 1 Interrupt Request Enable High Bit**
[5] T0ENH	**Timer 0 Interrupt Request Enable High Bit**
[4:1]	**Reserved** These bits are reserved and must be programmed to 000.
[0] ADCENH	**ADC Interrupt Request Enable High Bit**

Table 40. IRQ0 Enable Low Bit Register (IRQ0ENL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	T1ENL	T0ENL	Reserved				ADCENL
RESET	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R	R	R/W
Address	FC2H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1ENL	**Timer 1 Interrupt Request Enable Low Bit**
[5] T0ENL	**Timer 0 Interrupt Request Enable Low Bit**
[4:1]	**Reserved** These bits are reserved and must be programmed to 000.
[0] ADCENL	**ADC Interrupt Request Enable Low Bit**

IRQ1 Enable High and Low Bit Registers

Table 41 indicates priority control for the IRQ1 Register. The IRQ1 Enable High and Low Bit registers, shown in Tables 42 and 43, form a priority-encoded enabling service for interrupts in the Interrupt Request 1 Register. Priority is generated by setting the appropriate bits in each register.

Table 41. IRQ1 Enable and Priority Encoding

IRQ1ENH[*x*]	IRQ1ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High
Note: x indicates register bits 7–0.			

Table 42. IRQ1 Enable High Bit Register (IRQ1ENH)

Bit	7	6	5	4	3	2	1	0
Field	PA7ENH	PA6CENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC4H							

Bit	Description
[7] PA7ENH	**Port A Interrupt Request Enable High Bit**
[6] PA6CENH	**Port A Comparator Interrupt Request Enable High Bit**
[5:0] PAxENH	**Port A Interrupt Request Enable High Bits** Refer to the Interrupt Port Select Register for selection of either Port A or Port D as the interrupt source.
Note: x indicates register bits 5–0.	

Table 43. IRQ1 Enable Low Bit Register (IRQ1ENL)

Bit	7	6	5	4	3	2	1	0
Field	PA7ENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC5H							

Bit	Description
[7] PA7ENL	**Port A Interrupt Request Enable Low Bit**
[6] PA6CENL	**Port A Comparator Interrupt Request Enable Low Bit**
[5:0] PAxENL	**Port A Interrupt Request Enable Low Bits**
Note: x indicates register bits 5–0.	

IRQ2 Enable High and Low Bit Registers

Table 44 indicates priority control for the IRQ2 Register. The IRQ2 Enable High and Low Bit registers, shown in Tables 45 and 46, form a priority-encoded enabling service for interrupts in the Interrupt Request 2 Register. Priority is generated by setting the appropriate bits in each register.

Table 44. IRQ2 Enable and Priority Encoding

IRQ2ENH[*x*]	IRQ2ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High
Note: *x* indicates register bits 7–0.			

Table 45. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				C3ENH	C2ENH	C1ENH	C0ENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC7H							

Bit	**Description**
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3] C3ENH	**Port C3 Interrupt Request Enable High Bit**
[2] C2ENH	**Port C2 Interrupt Request Enable High Bit**
[1] C1ENH	**Port C1 Interrupt Request Enable High Bit**
[0] C0ENH	**Port C0 Interrupt Request Enable High Bit**

Table 46. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				C3ENL	C2ENL	C1ENL	C0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC8H							

Bit	**Description**
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3] C3ENL	**Port C3 Interrupt Request Enable Low Bit**
[2] C2ENL	**Port C2 Interrupt Request Enable Low Bit**
[1] C1ENL	**Port C1 Interrupt Request Enable Low Bit**
[0] C0ENL	**Port C0 Interrupt Request Enable Low Bit**

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register, shown in Table 47, determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A or Port D input pin.

Table 47. Interrupt Edge Select Register (IRQES)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FCDH							

Bit	**Description**
[7:0] IESx	**Interrupt Edge Select** 0 = An interrupt request is generated on the falling edge of the PA*x* input or PDx. 1 = An interrupt request is generated on the rising edge of the PA*x* input or PDx.

Note: x indicates register bits 7–0.

Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) Register, shown in Table 48, determines the source of the PADxS interrupts. The Shared Interrupt Select Register selects between Port A and alternate sources for the individual interrupts.

Because these shared interrupts are edge-triggered, it is possible to generate an interrupt just by switching from one shared source to another. For this reason, an interrupt must be disabled before switching between sources.

Table 48. Shared Interrupt Select Register (IRQSS)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	PA6CS	Reserved					
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FCEH							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] PA6CS	**PA6/Comparator Selection** 0 = PA6 is used for the interrupt caused by PA6CS interrupt request. 1 = The comparator is used for the interrupt caused by PA6CS interrupt request.
[5:0]	**Reserved** These bits are reserved and must be programmed to 000000.

Interrupt Control Register

The Interrupt Control (IRQCTL) Register, shown in Table 49, contains the master enable bit for all interrupts.

Table 49. Interrupt Control Register (IRQCTL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	IRQE	Reserved						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R
Address	FCFH							

Bit	Description
[7] IRQE	**Interrupt Request Enable** This bit is set to 1 by executing an EI (enable interrupts) or IRET (interrupt return) instruction or by a direct register write of 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, reset or by a direct register write of a 0 to this bit. 0 = Interrupts are disabled. 1 = Interrupts are enabled.
[6:0]	**Reserved** These bits are reserved and must be programmed to 0000000.

Timers

Z8 Encore! F083A Series products contain up to two 16-bit reloadable timers that are used for timing, event counting or generation of pulse width modulated (PWM) signals. The timers features include:

- 16-bit reload counter
- Programmable prescaler with prescale values ranging from 1 to 128
- PWM output generation
- Capture and compare capability
- External input pin for timer input, clock gating or capture signal. External input pin signal frequency is limited to a maximum of one-fourth the system clock frequency
- Timer output pin
- Timer interrupt

Architecture

Figure 10 displays the architecture of the timers.

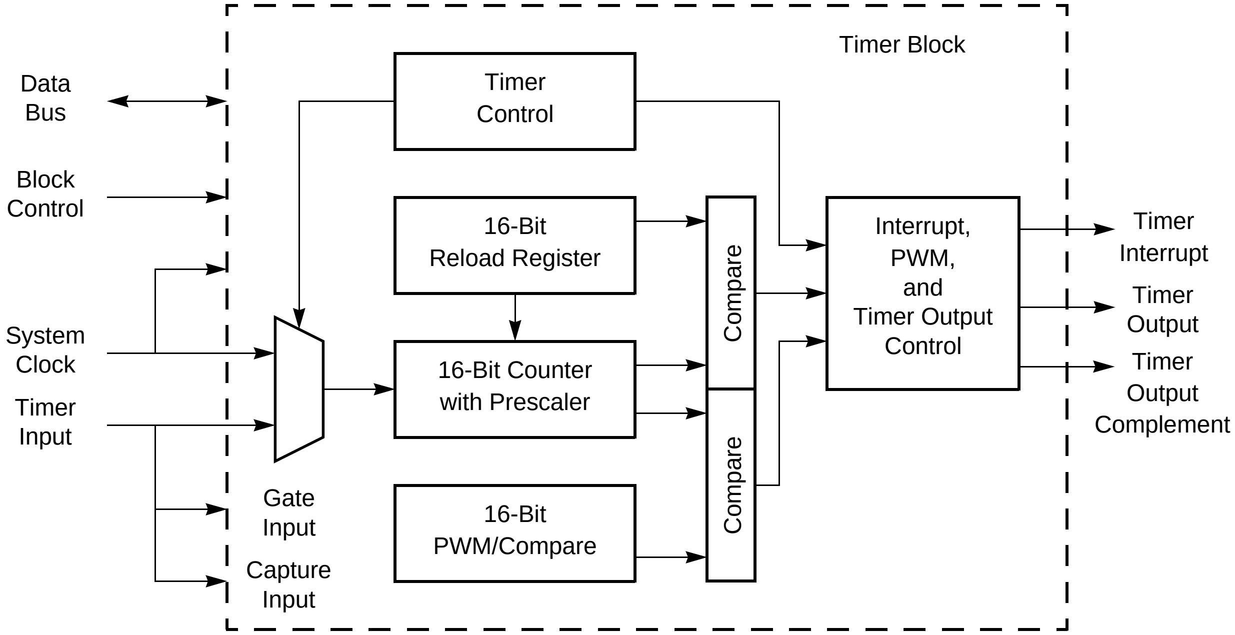


Figure 10. Timer Block Diagram

Operation

The timers are 16-bit upcounters. Minimum time-out delay is set by loading the value `0001H` into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value `0000H` into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the timer reaches `FFFFH`, the timer resets back to `0000H` and continues counting.

Timer Operating Modes

The timers are configured to operate in the following modes:

ONE-SHOT Mode

In ONE-SHOT Mode, the timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the reload value, the timer generates an interrupt and the count value in the Timer High and Low Byte registers is reset to `0001H`. The timer is automatically disabled and stops counting.

Additionally, if the timer output alternate function is enabled, the timer output pin changes state for one system clock cycle (from Low to High or from High to Low) upon timer reload. For the timer output to make a state change at a ONE-SHOT time-out (rather than a single cycle pulse), first set the TPOL bit in the Timer Control Register to the start value before enabling ONE-SHOT Mode. After starting the timer, set `TPOL` to the opposite bit value.

Observe the following steps to configure a timer for ONE-SHOT Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for ONE-SHOT Mode
   - Set the prescale value
   - Set the initial output level (High or Low) if using the timer output alternate function
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
6. Write to the Timer Control Register to enable the timer and initiate counting.

In ONE-SHOT Mode, the system clock always provides the timer input. The timer period is given by the following equation:

$$\text{One-Shot Mode Time-Out Period (s)} = \frac{(\text{Reload Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

CONTINUOUS Mode

In CONTINUOUS Mode, the timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. Additionally, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps to configure a timer for CONTINUOUS Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CONTINUOUS Mode
   - Set the prescale value
   - If using the timer output alternate function, set the initial output level ((High or Low))
2. Write to the Timer High and Low Byte registers to set the starting count value (usually `0001H`). This write only affects the first pass in CONTINUOUS Mode. After the first timer reload in CONTINUOUS Mode, counting always begins at the reset value of `0001H`.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Enable the timer interrupt (if appropriate) and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin (if using the timer output function) for the timer output alternate function.
6. Write to the Timer Control Register to enable the timer and initiate counting.

In CONTINUOUS Mode, the system clock always provides the timer input. The timer period is given by the following equation:

$$\text{Continuous Mode Time-Out Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than `0001H` is loaded into the Timer High and Low Byte registers, use the ONE-SHOT Mode equation to determine the first time-out period.

COUNTER Mode

In COUNTER Mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO port pin: timer input alternate function. The TPOL bit in the Timer Control Register determines whether the count occurs on the rising edge or the falling edge of the timer input signal. In COUNTER Mode, the prescaler is disabled.

Caution: The input frequency of the timer input signal must not exceed one-fourth the system clock frequency.

Upon reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. Additionally, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps to configure a timer for COUNTER Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COUNTER Mode
   - Select either the rising edge or falling edge of the timer input signal for the count. This selection also sets the initial logic level (High or Low) for the timer output alternate function. However, the timer output function is not required to be enabled
2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in COUNTER Mode. After the first timer reload in COUNTER Mode, counting always begins at the reset value `0001H`. In COUNTER Mode, the Timer High and Low Byte registers must be written with the value `0001H`.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin for the timer input alternate function.
6. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
7. Write to the Timer Control Register to enable the timer.

In COUNTER Mode, the number of timer input transitions is given by the following equation:

$$\text{Counter Mode Timer Input Transitions} = \text{Current Count Value} - \text{Start Value}$$

COMPARATOR COUNTER Mode

In COMPARATOR COUNTER Mode, the timer counts the input transitions from the analog comparator output. The TPOL bit in the Timer Control Register determines whether the count occurs on the rising edge or the falling edge of the comparator output signal. In COMPARATOR COUNTER Mode, the prescaler is disabled.

Caution: The frequency of the comparator output signal must not exceed one-fourth the system clock frequency.

After reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. Additionally, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps to configure a timer for COMPARATOR COUNTER Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COMPARATOR COUNTER Mode
   - Select either the rising edge or falling edge of the comparator output signal for the count. This also sets the initial logic level (High or Low) for the timer output alternate function. However, the timer output function is not required to be enabled
2. Write to the Timer High and Low Byte registers to set the starting count value. This write only affects the first pass in COMPARATOR COUNTER Mode. After the first timer reload in COMPARATOR COUNTER Mode, counting always begins at the reset value `0001H`. Generally, in COMPARATOR COUNTER Mode, the Timer High and Low Byte registers must be written with the value `0001H`.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
6. Write to the Timer Control Register to enable the timer.

In COMPARATOR COUNTER Mode, the number of comparator output transitions is given by the following equation:

$$\text{Comparator Output Transitions} = \text{Current Count Value} - \text{Start Value}$$

PWM SINGLE OUTPUT Mode

In PWM SINGLE OUTPUT Mode, the timer outputs a pulse width modulated (PWM) output signal through a GPIO port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the timer output toggles. The timer continues counting until it reaches the reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes.

If the TPOL bit in the Timer Control Register is set to 1, the timer output signal begins as a high (1) and transitions to a low (0) when the timer value matches the PWM value. The timer output signal returns to a high (1) after the timer reaches the reload value and is reset to `0001H`.

If the TPOL bit in the Timer Control Register is set to 0, the timer output signal begins as a low (0) and transitions to a high (1) when the timer value matches the PWM value. The timer output signal returns to a low (0) after the timer reaches the reload value and is reset to `0001H`.

Observe the following steps to configure a timer for PWM SINGLE OUTPUT Mode and for initiating the PWM operation are:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for PWM mode
   - Set the prescale value
   - Set the initial logic level (High or Low) and PWM high/low transition for the timer output alternate function
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of `0001H`.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the Timer Reload High and Low Byte registers to set the reload value (PWM period). The reload value must be greater than the PWM value.
5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the timer output alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than `0001H` is loaded into the Timer High and Low Byte registers, use the ONE-SHOT Mode equation to determine the first PWM time-out period.

If TPOL bit is set to 0, the ratio of the PWM output high time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL bit is set to 1, the ratio of the PWM output high time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

PWM DUAL OUTPUT Mode

In PWM DUAL OUTPUT Mode, the timer outputs a PWM output signal pair (i.e., a basic PWM signal and its complement) through two GPIO port pins. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the timer output toggles. The timer continues counting until it reaches the reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes.

If the TPOL bit in the Timer Control Register is set to 1, the timer output signal begins as a high (1) and transitions to low (0) when the timer value matches the PWM value. The timer output signal returns to high (1) after the timer reaches the reload value and is reset to `0001H`.

If the TPOL bit in the Timer Control Register is set to 0, the timer output signal begins as a low (0) and transitions to high (1) when the timer value matches the PWM value. The timer output signal returns to low (0) after the timer reaches the reload value and is reset to `0001H`.

The timer also generates a second PWM output signal: the timer output complement. The timer output complement is the complement of the timer output PWM signal. A programmable deadband delay is configured to time delay (0 to 128 system clock cycles) PWM output transitions on these two pins from a Low to a High (i.e., inactive to active) to ensure a time gap between the deassertion of one PWM output to the assertion of its complement.

Observe the following steps to configure a timer for PWM DUAL OUTPUT Mode and for initiating the PWM operation are:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for PWM DUAL OUTPUT Mode. Setting the mode also involves writing to TMODEHI bit in the TxCTL1 Register
   - Set the prescale value
   - Set the initial logic level (High or Low) and PWM high/low transition for the timer output alternate function
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of `0001H`.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the PWM Control Register to set the PWM deadband delay value. The deadband delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM High and Low Byte registers). It must also be less than the duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the timer reload registers).
5. Write to the Timer Reload High and Low Byte registers to set the reload value (PWM period). The reload value must be greater than the PWM value.
6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
7. Configure the associated GPIO port pin for the timer output and timer output complement alternate functions. The timer output complement function is shared with the timer input function for both timers. Setting the timer mode to dual PWM, will automatically switch the function from timer-in to timer-out complement.
8. Write to the Timer Control Register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than `0001H` is loaded into the Timer High and Low Byte registers, the ONE-SHOT Mode equation determines the first PWM time-out period.

If TPOL bit is set to 0, the ratio of the PWM output high time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL bit is set to 1, the ratio of the PWM output high time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

CAPTURE Mode

In CAPTURE Mode, the current timer count value is recorded when the appropriate external timer input transition occurs. The capture count value is written to the timer PWM High and Low Byte registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines if a capture event occurs on a rising edge or a falling edge of the timer input signal.

When a capture event occurs, an interrupt is generated and the timer continues counting. The INPCAP bit in the TxCTL1 Register is set to indicate the timer interrupt because of an input capture event.

The timer continues counting up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt and continues counting. The INPCAP bit in the TxCTL1 Register clears, indicating that the timer interrupt has not occurred because of an input capture event.

Observe the following steps to configure a timer for CAPTURE Mode and initiating the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE Mode
   - Set the prescale value
   - Set the capture edge (rising or falling) for the timer input

2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Clear the timer PWM High and Low Byte registers to `0000H`. Clearing these registers allows user software to determine if interrupts were generated either by a capture event or by a reload. If the PWM High and Low Byte registers still contain `0000H` after the interrupt, the interrupt were generated by a reload.

5. Enable the timer interrupt, if appropriate and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.
6. Configure the associated GPIO port pin for the timer input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time between the timer start and the capture event is calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

CAPTURE RESTART Mode

In CAPTURE RESTART Mode, the current timer count value is recorded when an acceptable external timer input transition occurs. The capture count value is written to the timer PWM High and Low Byte registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines whether the capture occurs on a rising edge or a falling edge of the timer input signal. When a capture event occurs, an interrupt is generated and the count value in the Timer High and Low Byte registers is reset to `0001H`; counting then resumes. The INPCAP bit in the TxCTL1 Register is set to indicate that the timer interrupt is caused by an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value that is stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The INPCAP bit in the TxCTL1 Register is cleared to indicate that the timer interrupt is not caused by an input capture event.

Observe the following steps to configure a timer for CAPTURE RESTART Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE RESTART Mode. Setting the mode also involves writing to TMODEHI bit in the TxCTL1 Register
   - Set the prescale value
   - Set the capture edge (rising or falling) for the timer input

2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).

3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Clear the timer PWM High and Low Byte registers to `0000H`. This allows user software to determine if interrupts are generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain `0000H` after the interrupt, the interrupt were generated by a reload.
5. Enable the timer interrupt, if appropriate and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input capture and reload events. The user must configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.
6. Configure the associated GPIO port pin for the timer input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time between the timer start and the capture event is calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

COMPARE Mode

In COMPARE Mode, the timer counts up to the 16-bit maximum compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the compare value, the timer generates an interrupt and counting continues (the timer value is not reset to `0001H`). Additionally, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) upon compare.

If the timer reaches `FFFFH`, the timer resets to `0000H` and continues counting.

Observe the following steps to configure a timer for COMPARE Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COMPARE Mode
   - Set the prescale value
   - Set the initial logic level (High or Low) for the timer output alternate function

2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the compare value.

4. Enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
6. Write to the Timer Control Register to enable the timer and initiate counting.

In COMPARE Mode, the system clock always provides the timer input. The compare time is calculated by the following equation:

$$\text{Compare Mode Time (s)} = \frac{(\text{Compare Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

GATED Mode

In GATED Mode, the timer counts only when the timer input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control Register. When the timer input signal is asserted, counting begins. A timer interrupt is generated when the timer input signal is deasserted or a timer reload occurs. To determine whether the timer input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes (assuming the timer input signal remains asserted). Additionally, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reset.

Observe the following steps to configure a timer for GATED Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for GATED Mode
   - Set the prescale value
2. Write to the Timer High and Low Byte registers to set the starting count value. Writing these registers only affects the first pass in GATED Mode. After the first timer reset in GATED Mode, counting always begins at the reset value of `0001H`.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input deasser-

tion and reload events. Therefore, configure the timer interrupt to be generated only at the input deassertion event or the reload event by setting TICONFIG bit of the TxCTL1 Register.

5. Configure the associated GPIO port pin for the timer input alternate function.
6. Write to the Timer Control Register to enable the timer.
7. Assert the timer input signal to initiate the counting.

CAPTURE/COMPARE Mode

In CAPTURE/COMPARE Mode, the timer begins counting on the first external timer input transition. The acceptable transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the system clock.

Every subsequent acceptable transition (after the first) of the timer input signal, captures the current count value. The capture value is written to the timer PWM High and Low Byte registers. When a capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to `0001H` and the counting resumes. The INPCAP bit in the TxCTL1 Register is set to indicate that the timer interrupt is caused by an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The INPCAP bit in the TxCTL1 Register is cleared to indicate that the timer interrupt is not caused by an input capture event.

Observe the following steps to configure a timer for CAPTURE/COMPARE Mode and to initiate the count.

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE/COMPARE Mode
   - Set the prescale value
   - Set the capture edge (rising or falling) for the timer input

2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
3. Write to the Timer Reload High and Low Byte registers to set the compare value.
4. Enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.By default, the timer interrupt is generated for both input capture and reload events. The user must configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG bit of the TxCTL1 Register.

5. Configure the associated GPIO port pin for the timer input alternate function.
6. Write to the Timer Control Register to enable the timer.
7. Counting begins on the first appropriate transition of the timer input signal. No interrupt is generated by the first edge.

In CAPTURE/COMPARE Mode, the elapsed time from timer start to capture event is calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

Reading the Timer Count Values

The current count value in the timers are read (i.e., enabled) while counting. This capability has no effect on Timer operation. When the timer is enabled and the Timer High Byte Register is read, the contents of the Timer Low Byte Register are placed in a holding register. A subsequent read from the Timer Low Byte Register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value when enabled. When the timers are not enabled, a read from the Timer Low Byte Register returns the actual value in the counter.

Timer Pin Signal Operation

Timer output is a GPIO port pin alternate function. The timer output is toggled every time the counter is reloaded.

The timer input is used as a selectable counting source. It shares the same pin as the complementary timer output. When selected by the GPIO alternate function registers, this pin functions as a timer input in all modes except for the DUAL PWM OUTPUT Mode. For this mode, there is no timer input available.

Timer Control Register Definitions

This section defines the features of the following Timer Control registers.

Timer 0–1 High and Low Byte Registers: see page 84

Timer Reload High and Low Byte Registers: see page 84

Timer 0–1 PWM High and Low Byte Registers: see page 86

Timer 0–1 Control Registers: see page 87

Timer 0–1 High and Low Byte Registers

The Timer 0–1 High and Low Byte (TxH and TxL) registers, shown in Tables 50 and 51, contain the current 16-bit timer count value. When the timer is enabled, a read from TxH causes the value in TxL to be stored in a temporary holding register. A read from TxL always returns this temporary register content when the timer is enabled; however, when the timer is disabled, a read from the TxL reads the TxL Register content directly.

Writing to the Timer High and Low Byte registers while the timer is enabled is not recommended. There are no temporary holding registers available for write operations; therefore, simultaneous 16-bit writes are not possible. If either of the Timer High or Low Byte registers are written to during counting, the 8-bit written value is placed in the counter (high or low byte) at the next clock edge. The counter continues counting from the new value.

Table 50. Timer 0–1 High Byte Register (TxH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F00H, F08H							

Table 51. Timer 0–1 Low Byte Register (TxL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TL							
RESET	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F01H, F09H							

Bit	**Description**
[7:0] TH, TL	**Timer High and Low Bytes** These 2 bytes, {TH[7:0], TL[7:0]}, contain the current 16-bit timer count value.

Timer Reload High and Low Byte Registers

The timer 0–1 reload High and Low Byte (TxRH and TxRL) registers, shown in Tables 52 and 53, store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the timer reload high byte register are stored in a temporary holding register. When a write to the timer reload low byte register occurs, the temporary holding register value is written to the Timer High Byte Register. This operation allows simultaneous updates of the 16-bit timer reload value. In COMPARE Mode, the Timer Reload High and Low Byte registers store the 16-bit compare value.

Table 52. Timer 0–1 Reload High Byte Register (TxRH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F02H, F0AH							

Table 53. Timer 0–1 Reload Low Byte Register (TxRL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TRL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F03H, F0BH							

Bit	**Description**
[7:0] TRH, TRL	**Timer Reload Register High and Low Bytes** These two bytes form the 16-bit reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value, which initiates a timer reload to `0001H`. In COMPARE Mode, these two bytes form the 16-bit compare value.

Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers, shown in Tables 54 and 55, control PWM operations. These registers also store the capture values for the CAPTURE and CAPTURE/COMPARE modes.

Table 54. Timer 0–1 PWM High Byte Register (TxPWMH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PWMH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F04H, F0CH							

Table 55. Timer 0–1 PWM Low Byte Register (TxPWML)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PWML							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F05H, F0DH							

Bit	**Description**
[7:0] PWMH, PWML	**Pulse width modulator High and Low Bytes** These two bytes, {PWMH[7:0], PWML[7:0]}, form a 16-bit value that is compared to the current 16-bit timer count. When a match occurs, the PWM output changes state. The PWM output value is set by the TPOL bit in the Timer Control Register (TxCTL1). The TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in capture or CAPTURE/COMPARE modes.

Timer 0–1 Control Registers

The Timer Control registers are 8-bit read/write registers that control the operation of their associated counter/timers.

Time 0–1 Control Register 0

The Timer Control Register 0 (TxCTL0) and the Timer Control Register 1 (TxCTL1) determine the timer operating mode. It also includes a programmable PWM deadband delay, two bits to configure timer interrupt definition and a status bit to identify, if the most recent timer interrupt is caused by an input capture event.

Table 56. Timer 0–1 Control Register 0 (TxCTL0)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TMODEHI	TICONFIG		Reserved	PWMD			INPCAP
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F06H, F0EH							

Bit	Description
[7] TMODEHI	**Timer Mode High Bit** This bit, along with the TMODE field in the TxCTL1 Register, determines the operating mode of the timer; it is the most significant bit of the timer mode selection value. For more details, see Timer 0–1 Control Register 1 on page 88.
[6:5] TICONFIG	**Timer Interrupt Configuration** This field configures timer interrupt definition. 0x = Timer interrupt occurs on all of the defined reload, compare and input events. 10 = Timer interrupt occurs only on defined input Capture/Deassertion events. 11 = Timer interrupt occurs only on defined Reload/Compare events.
[4]	**Reserved** This bit is reserved and must be programmed to 0.
[3:1] PWMD	**PWM Delay Value** This field is a programmable delay to control the number of system clock cycles delay before the timer output and the timer output complement are forced to their active state. 000 = No delay. 001 = 2-cycle delay. 010 = 4-cycle delay. 011 = 8-cycle delay. 100 = 16-cycle delay. 101 = 32-cycle delay. 110 = 64-cycle delay. 111 = 128-cycle delay.

Bit	Description (Continued)
[0] INPCAP	**Input Capture Event** This bit indicates whether the most recent timer interrupt is caused by a timer input capture event. 0 = Previous timer interrupt is not caused by timer input capture event. 1 = Previous timer interrupt is caused by timer input capture event .

Timer 0–1 Control Register 1

The Timer 0–1 Control (TxCTL1) Register, shown in Table 57, enables/disables the timers, set the prescaler value and determines the timer operating mode.

Table 57. Timer 0–1 Control Register 1 (TxCTL1)

Bit	7	6	5	4	3	2	1	0
Field	TEN	TPOL	PRES			TMODE		
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F07H, F0FH							

Bit	Description
[7] TEN	**Timer Enable** 0 = Timer is disabled. 1 = Timer enabled to count.

Bit	Description (Continued)
[6] TPOL	**Timer Input/Output Polarity** Operation of this bit is a function of the current operating mode of the timer. **ONE-SHOT Mode** When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled, the timer output signal is complemented on timer reload. **CONTINUOUS Mode** When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled and reloaded, the timer output signal is complemented. **COUNTER Mode** If the timer is disabled, the timer output signal is set to the value of this bit. If the timer is enabled the timer output signal is complemented after timer reload. 0 = Count occurs on the rising edge of the timer input signal. 1 = Count occurs on the falling edge of the timer input signal. **PWM SINGLE OUTPUT Mode** 0 = Timer output is forced low (0), when the timer is disabled. The timer output is forced high (1), when the timer is enabled and the PWM count matches and the timer output is forced low (0), when the timer is enabled and reloaded. 1 = Timer output is forced high (1), when the timer is disabled. The timer output is forced low(0), when the timer is enabled and the PWM count matches and forced high (1) when the timer is enabled and reloaded. **CAPTURE Mode** 0 = Count is captured on the rising edge of the timer input signal. 1 = Count is captured on the falling edge of the timer input signal. **COMPARE Mode** When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled and reloaded, the timer output signal is complemented. **GATED Mode** 0 = Timer counts when the timer input signal is high (1) and interrupts are generated on the falling edge of the timer input. 1 = Timer counts when the timer input signal is low (0) and interrupts are generated on the rising edge of the timer input. **CAPTURE/COMPARE Mode** 0 = Counting is started on the first rising edge of the timer input signal. The current count is captured on subsequent rising edges of the timer input signal. 1 = Counting is started on the first falling edge of the timer input signal. The current count is captured on subsequent falling edges of the timer input signal.

Bit	Description (Continued)
[6] TPOL (cont'd.)	**PWM DUAL OUTPUT Mode** 0 = Timer output is forced low (0) and timer output complement is forced high (1), when the timer is disabled. When enabled and the PWM count matches, the timer output is forced high (1) and forced low (0) when enabled and reloaded. When enabled and the PWM count matches, the timer output complement is forced low (0) and forced high (1) when enabled and reloaded. 1 = Timer output is forced high (1) and timer output complement is forced low (0) when the timer is disabled. When enabled and the PWM count matches, the timer output is forced low (0) and forced high (1) when enabled and reloaded.When enabled and the PWM count matches, the timer output complement is forced high (1) and forced low (0) when enabled and reloaded. The PWMD field in the TxCTL0 register determiners an optional added delay on the assertion (low to high) transition of both timer output and timer output complement for deadband generation. **CAPTURE RESTART Mode** 0 = Count is captured on the rising edge of the timer input signal. 1 = Count is captured on the falling edge of the timer input signal. **COMPARATOR COUNTER Mode** When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled, the timer output signal is complemented on timer reload. When the timer output alternate function TxOUT on a GPIO port pin is enabled, TxOUT will change to whatever state the TPOL bit is in. The timer is not required to be enabled for that to happen. Additionally, the port data direction sub register is not needed to be set to output on TxOUT. Changing the TPOL bit when the timer is enabled and running does not immediately change the polarity TxOUT.
[5:3] PRES	**Prescale Value** The timer input clock is divided by 2^{PRES}, where PRES is set from 0 to 7. The prescaler is reset each time the timer is disabled. This reset ensures proper clock division each time the timer is restarted. 000 = Divide by 1. 001 = Divide by 2. 010 = Divide by 4. 011 = Divide by 8. 100 = Divide by 16. 101 = Divide by 32. 110 = Divide by 64. 111 = Divide by 128.

Bit	Description (Continued)
[2:0] TMODE	**TIMER Mode** This field along with the TMODEHI bit in TxCTL0 register determines the operating mode of the timer. TMODEHI is the most significant bit of the timer mode selection value. 0000 = ONE-SHOT Mode. 0001 = CONTINUOUS Mode. 0010 = COUNTER Mode. 0011 = PWM SINGLE OUTPUT Mode. 0100 = CAPTURE Mode. 0101 = COMPARE Mode. 0110 = GATED Mode. 0111 = CAPTURE/COMPARE Mode. 1000 = PWM DUAL OUTPUT Mode. 1001 = CAPTURE RESTART Mode. 1010 = COMPARATOR COUNTER Mode.

Watchdog Timer

The Watchdog Timer (WDT) protects from corrupted or unreliable software, power faults and other system-level problems, which may place the Z8 Encore! F083A Series devices into unsuitable operating states. The Watchdog Timer includes the following features:

- On-chip RC oscillator
- A selectable time-out response: reset or interrupt
- 24-bit programmable time-out value

Operation

The WDT is a retriggerable one-shot timer that resets or interrupts Z8 Encore! F083A Series devices when the WDT reaches its terminal count. The WDT uses a dedicated on-chip RC oscillator as its clock source. The WDT operates only in two modes: ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. Perform an enable by executing the WDT instruction or by setting the WDT_AO Flash option bit. The WDT_AO bit forces the WDT to operate immediately upon reset, even if a WDT instruction has not been executed.

The WDT is a 24-bit reloadable downcounter that uses three 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is calculated by the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{10}$$

where the WDT reload value is the 24-bit decimal value furnished by {WDTU[7:0], WDTH[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10kHz. The Watchdog Timer cannot be refreshed after it reaches `000002H`. The WDT reload value must not be set to values below `000004H`. Table 58 provides information about approximate time-out delays for the minimum and maximum WDT reload values.

Table 58. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value (Hex)	WDT Reload Value (Decimal)	Approximate Time-Out Delay (with 10kHz Typical WDT Oscillator Frequency)	
		Typical	**Description**
000004	4	400µs	Minimum time-out delay
000400	1024	102 ms	Default time-out delay
FFFFFF	16,777,215	28 minutes	Maximum time-out delay

Watchdog Timer Refresh

Upon first enable, the Watchdog Timer is loaded with the value in the WDT reload registers. The Watchdog Timer counts down to 000000H unless a WDT instruction is executed by the eZ8 CPU. Execution of the WDT instruction causes the downcounter to be reloaded with the WDT reload value stored in the WDT reload registers. Counting resumes following the reload operation.

When Z8 Encore! F083A Series devices are operating in DEBUG Mode (using the On-Chip Debugger), the WDT must be continuously refreshed to prevent any WDT Timer time-outs.

Watchdog Timer Time-Out Response

The WDT times out when the counter reaches 000000H. A time-out of the WDT generates either an interrupt or a system reset. The WDT_RES Flash option bit determines the time-out response of the WDT. For more details about programming of WDT_RES Flash option bit, see *the* Flash Option Bits chapter on page 124.

WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the Watchdog Timer issues an interrupt request to the Interrupt Controller and sets the WDT status bit in the reset status register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executing code from the vector address. After time-out and interrupt generation, the Watchdog Timer counter resets to its maximum value of FFFFFH and continues counting. The Watchdog Timer counter will not automatically return to its reload value.

The Reset Status Register, shown in Table 12 on page 29, must be read before clearing the WDT interrupt. This read clears the WDT time-out flag and prevents further WDT interrupts occurring immediately.

WDT Interrupt in STOP Mode

If configured to generate an interrupt when a time-out occurs and the Z8 Encore! F083A Series devices are in STOP Mode, the Watchdog Timer automatically initiates a Stop Mode Recovery and generates an interrupt request. Both the WDT status bit and the STOP bit in the Watchdog Timer Control Register are set to 1 following a WDT time-out in STOP Mode. For more details about Stop Mode Recovery, see *the* Reset and Stop Mode Recovery chapter on page 21.

If interrupts are enabled, following completion of the Stop Mode Recovery, the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executes the code from the vector address.

WDT Reset in Normal Operation

If configured to generate a reset when a time-out occurs, the Watchdog Timer forces the device into the system Reset state. The WDT status bit in the Watchdog Timer Control Register is set to 1. For more details about system reset, see *the* Reset and Stop Mode Recovery chapter on page 21.

WDT Reset in STOP Mode

If configured to generate a reset when a time-out occurs and the device is in STOP Mode, the Watchdog Timer initiates a Stop Mode Recovery. Both the WDT status bit and the STOP bit in the Watchdog Timer Control Register are set to 1 following WDT time-out in STOP Mode. For more details, see *the* Reset and Stop Mode Recovery chapter on page 21.

Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watchdog Timer (WDTCTL) Control Register address, unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL Register address produce no effect on the bits in the WDTCTL Register. The locking mechanism prevents spurious writes to the reload registers. The following sequence is required to unlock the Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) for write access.

1. Write 55H to the Watchdog Timer Control Register (WDTCTL).
2. Write AAH to the Watchdog Timer Control Register (WDTCTL).
3. Write the Watchdog Timer reload upper byte register (WDTU).
4. Write the Watchdog Timer reload high byte register (WDTH).
5. Write the Watchdog Timer reload low byte register (WDTL).

All three Watchdog Timer Reload registers must be written in the order listed above. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes occurs unless the sequence is restarted. The value in the Watchdog Timer reload registers is loaded into the counter when the Watchdog Timer is first enabled and every time a WDT instruction is executed.

Watchdog Timer Control Register Definitions

This section defines the features of the following Watchdog Timer Control registers.

Watchdog Timer Control Register (WDTCTL): see page 95

Watchdog Timer Reload Upper Byte Register (WDTU): see page 96

Watchdog Timer Reload High Byte Register (WDTH): see page 96

Watchdog Timer Reload Low Byte Register (WDTL): see page 97

Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) Register, shown in Table 59, is a write-only control register. Writing the unlock sequence: 55H, AAH to the WDTCTL Register address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL Register address have no effect on the bits in the WDTCTL Register. The locking mechanism prevents spurious writes to the reload registers.

This register address is shared with the read-only Reset Status Register.

Table 59. Watchdog Timer Control Register (WDTCTL)

Bit	7	6	5	4	3	2	1	0
Field	WDTUNLK							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
Address	FF0H							

Bit	Description
[7:0] WDTUNLK	**Watchdog Timer Unlock** The user software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer reload registers.

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTH, WDTL) registers, shown in Tables 60 through 62, form the 24-bit reload value{WDTU[7:0], WDTH[7:0], WDTL[7:0]} that is loaded into the Watchdog Timer when a WDT instruction is executed. Writing to these registers sets the appropriate reload value; reading from these registers returns the current Watchdog Timer count value.

⚠ **Caution:** The 24-bit WDT reload value must not be set to a value less than `000004H`.

Table 60. Watchdog Timer Reload Upper Byte Register (WDTU)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	WDTU							
RESET	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF1H							
Note: *R/W = A read returns the current WDT count value; a write sets the appropriate reload value.								

Bit	**Description**
[7:0] WDTU	**WDT Reload Upper Byte** MSB, Bits[23:16], of the 24-bit WDT reload value.

Table 61. Watchdog Timer Reload High Byte Register (WDTH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	WDTH							
RESET	0	0	0	0	0	1	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF2H							
Note: *R/W = A read returns the current WDT count value; a write sets the appropriate reload value.								

Bit	**Description**
[7:0] WDTH	**WDT Reload High Byte** Middle byte, bits[15:8] of the 24-bit WDT reload value.

Table 62. Watchdog Timer Reload Low Byte Register (WDTL)

Bit	7	6	5	4	3	2	1	0
Field	WDTL							
RESET	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF3H							
Note: *A read returns the current WDT count value; a write sets the appropriate reload value.								

Bit	Description
[7:0] WDTL	**WDT Reload Low Byte** LSB, bits[7:0] of the 24-bit WDT reload value.

Analog-to-Digital Converter

Z8 Encore! F083A Series devices include an eight-channel Successive Approximation Register (SAR) analog-to-digital converter (ADC). The ADC converts an analog input signal to a 10-bit binary number. The features of the SAR ADC include:

- Eight analog input sources multiplexed with general purpose I/O ports
- Fast conversion time, as low as 2.8 µs (ADC conversion clock should be less than 10 MHz)
- Programmable timing controls
- Interrupt on conversion complete
- Internal voltage reference generator
- Ability to select external reference voltage.
- When configuring ADC using external Vref, PB5 is used as Vref in 28-pin package

Architecture

The ADC architecture, shown in Figure 11, consists of an 8-input multiplexer, sample-and-hold amplifier and 10-bit SAR ADC. The ADC digitizes the signal on a selected channel and stores the digitized data in the ADC data registers. In an environment with high electrical noise, an external RC filter must be added at the input pins to reduce high-frequency noise.

$T_{CONV} = T_{S/H} + T_{CON}$

$T_{CONV} = T_S + T_H + 13 * \text{SCLK} * \text{ADC Prescaler}$

where:

SCLK = System clock

T_{CONV} = Total conversion time

Ts = Sample time (SCLK*ADCST)

T_{CON} = Conversion time (13*SCLK*ADCCP)

T_H = Hold time (SCLK*ADCSST)

DIV = ADC Prescaler register number

Example: For an F083A device operating at 10 MHz:

$T_{CONV} = 1\,\mu s + 0.5\,\mu s + 13 * \text{SCLK} * \text{DIV}$

$T_{CONV} = 1\,\mu s + 0.5\,\mu s + 13 * (1/10\,\text{MHz}) * 1 = 2.8\,\mu s$

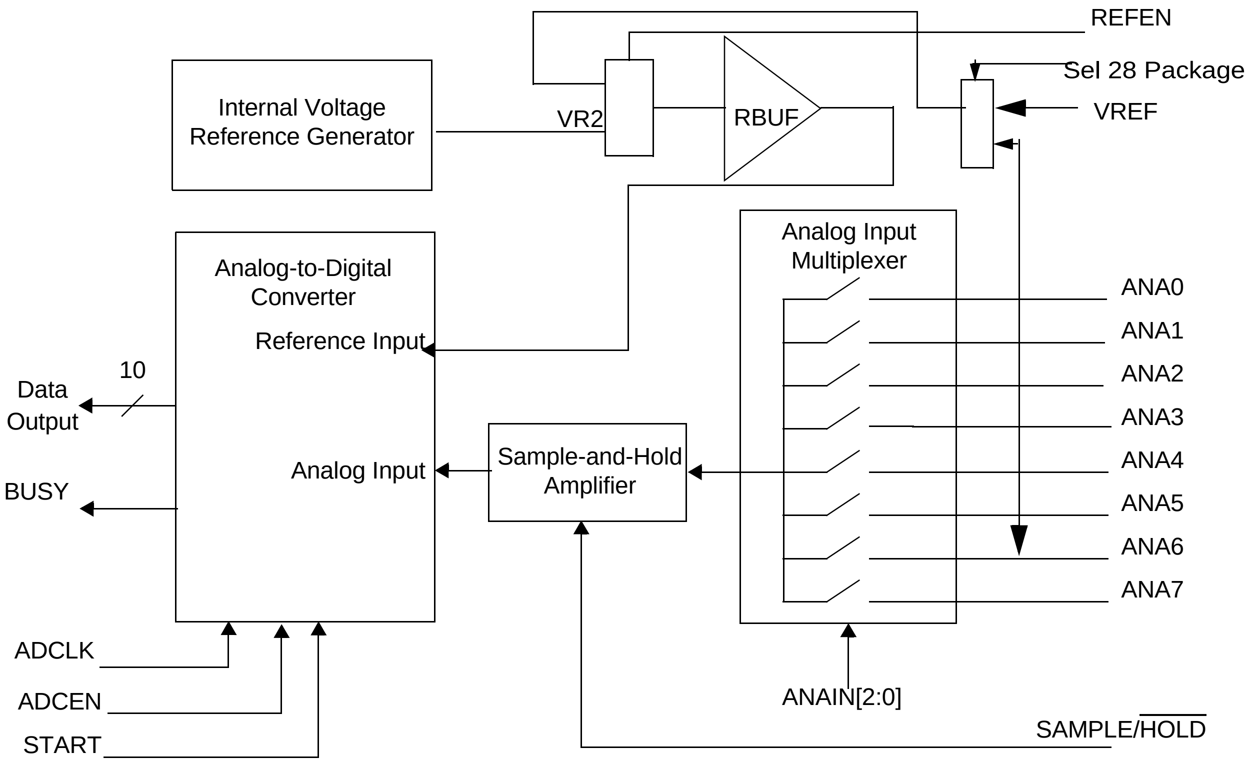


Figure 11. Analog-to-Digital Converter Block Diagram

Operation

The ADC converts the analog input, ANA_X, to a 10-bit digital representation. The equation for calculating the digital value is represented by:

$$\text{ADCOutput} = 1024 \times (\text{ANA}_x \div V_{REF})$$

Assuming zero gain and offset errors, any voltage outside the ADC input limits of AV_{SS} and V_{REF} returns all 0s or 1s, respectively. A new conversion is initiated by a software to the ADC Control Register’s start bit.

Initiating a new conversion stops any conversion currently in progress and begins a new conversion. To avoid disrupting a conversion already in progress, the START bit is read to determine ADC operation status (i.e., busy or available).

ADC Timing

Each ADC measurement consists of three phases:

1. Input sampling (programmable, minimum of 1.0 µs).
2. Sample-and-hold amplifier settling (programmable, minimum of 0.5 µs).
3. Conversion is 13 ADCLK cycles.

Figure 12 displays the timing of an ADC conversion.

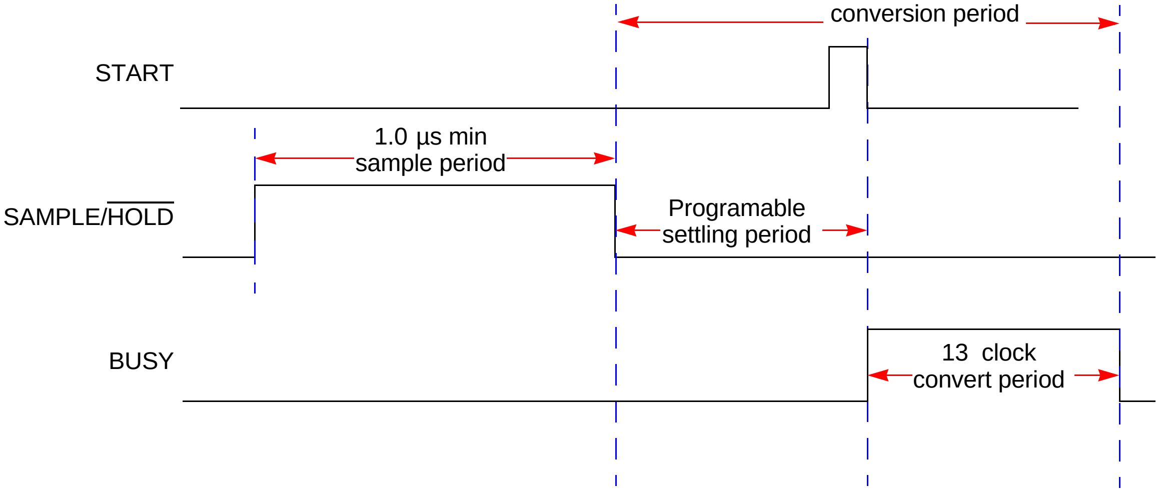


Figure 12. ADC Timing Diagram

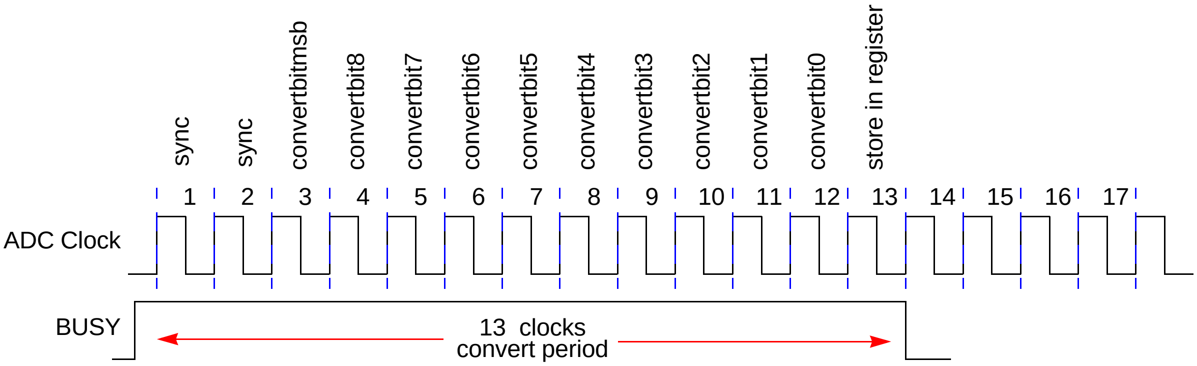


Figure 13. ADC Convert Timing

ADC Interrupt

The ADC generates an interrupt request when a conversion has been completed. An interrupt request that is pending when the ADC is disabled is not cleared automatically.

Reference Buffer

The reference buffer, RBUF, supplies the reference voltage for the ADC. When enabled, the internal voltage reference generator supplies the ADC. When RBUF is disabled, the ADC must have the reference voltage supplied externally through the V_{REF} pin in 28-pin package. RBUF is controlled by the REFEN bit in the ADC Control Register.

Internal Voltage Reference Generator

The internal voltage reference generator provides the RBUF voltage, VR2; this VR2 value is 2 V.

Calibration and Compensation

The user can perform calibration and store the values into Flash; conversely, the user code can perform a manual offset calibration. There is no provision for manual gain calibration.

ADC Control Register Definitions

The ADC control registers are defined in this section.

ADC Control Register 0

The ADC Control Register 0, shown in Table 63, initiates the A/D conversion and provides ADC status information.

Table 63. ADC Control Register 0 (ADCCTL0)

Bit	7	6	5	4	3	2	1	0
Field	START	Reserved	REFEN	ADCEN	Reserved	ANAIN[2:0]		
RESET	0	0	0	0	0	0	0	0
R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F70h							

Bit Position	Description
[7] START	**ADC Start/Busy** 0 = Writing to 0 has no effect. Reading a 0 indicates that the ADC is available to begin a conversion. 1 = Writing to 1 starts a conversion. Reading a 1 indicates that a conversion is currently in progress.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] REFEN	**Reference Enable** 0 = Internal reference voltage is disabled allowing an external reference voltage to be used by the ADC. 1 = Internal reference voltage for the ADC is enabled. The internal reference voltage is measured on the VREF pin.
[4] ADCEN	**ADC Enable** 0 = ADC is disabled for Low Power operation. 1 = ADC is enabled for normal use.
[3]	**Reserved** This bit is reserved and must be programmed to 0.

Bit Position	Description (Continued)
[2:0] ANAIN	**Analog Input Select** 000 = ANA0 input is selected for analog-to-digital conversion. 001 = ANA1 input is selected for analog-to-digital conversion. 010 = ANA2 input is selected for analog-to-digital conversion. 011 = ANA3 input is selected for analog-to-digital conversion. 100 = ANA4 input is selected for analog-to-digital conversion. 101 = ANA5 input is selected for analog-to-digital conversion. 110 = ANA6 input is selected for analog-to-digital conversion. 111 = ANA7 input is selected for analog-to-digital conversion.

ADC Data High Byte Register

The ADC Data High Byte Register, shown in Table 64, contains the upper eight bits of the ADC output. Access to the ADC Data High Byte Register is read-only. Reading the ADC Data High Byte Register latches data in the ADC Low Bits Register.

Table 64. ADC Data High Byte Register (ADCD_H)

Bit	7	6	5	4	3	2	1	0
Field	ADCDH							
RESET	X							
R/W	R							
Address	F72H							

Bit Position	Value (H)	Description
[7:0]	00h–FFh	ADC high byte The last conversion output is held in the data registers until the next ADC conversion is completed.

ADC Data Low Bits Register

The ADC Data Low Bits Register, shown in Table 65, contain the lower bits of the ADC output as well as an overflow status bit. Access to the ADC Data Low Bits Register is read-only. Reading the ADC Data High Byte Register latches lower bits of the ADC in the ADC Low Bits Register.

Table 65. ADC Data Low Bits Register (ADCD_L)

Bit	7	6	5	4	3	2	1	0
Field	ADCDL		Reserved					
RESET	X		X					
R/W	R		R					
Address	F73H							

Bit	Description
[7:6]	**ADC Low Bits** 00–11b = These bits are the two least significant bits of the 10-bit ADC output. These bits are undefined after a reset. The low bits are latched into this register whenever the ADC Data High Byte Register is read.
[5:0]	**Reserved** These bits are reserved and must be programmed to 000000.

Sample Settling Time Register

The Sample Settling Time Register, shown in Table 66, is used to program the delay after the SAMPLE/HOLD signal is asserted and before the START signal is asserted; the conversion then begins. The number of clock cycles required for settling will vary from system to system depending on the system clock period used. The system designer must program this register to contain the number of clocks required to meet a 0.5 µs minimum settling time.

Table 66. Sample Settling Time (ADCSST)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				SST			
RESET	0				1	1	1	1
R/W	R				R/W			
Address	F74H							

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:0] SST	0h–Fh: Sample settling time in number of system clock periods to meet 0.5 µs minimum.

Sample Time Register

The sample time register, shown in Table 67, is used to program the length of active time for the sample after a conversion has begun by setting the START bit in the ADC Control Register. The number of system clock cycles required for the sample time varies from system to system, depending on the clock period used. The system designer must program this register to contain the number of system clocks required to meet a 1 µs minimum sample time.

Table 67. Sample Time (ADCST)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved		ST					
RESET	0		1	1	1	1	1	1
R/W	R/W		R/W					
Address	F75H							

Bit	**Description**
[7:6]	**Reserved** These bits are reserved and must be programmed to 00.
[5:0] ST	**Sample/Hold Time** Measured in number of system clock periods to meet 1 µs minimum.

ADC Clock Prescale Register

The ADC Clock Prescale Register, shown in Table 68, is used to provide a divided system clock to the ADC. When this register is programmed with `0h`, the system clock is used for the ADC clock. DIV8 has the highest priority, DIV2 has the lowest priority.

Table 68. ADC Clock Prescale Register (ADCCP)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved					DIV8	DIV4	DIV2
RESET	0					0	0	0
R/W	R/W							
Address	F76H							

Bit	**Description**
[0] DIV2	**DIV2** 0 = Clock is not divided. 1 = System clock is divided by 2 for ADC clock.
[1] DIV4	**DIV4** 0 = Clock is not divided. 1 = System clock is divided by 4 for ADC clock
[2] DIV8	**DIV8** 0 = Clock is not divided. 1 = System clock is divided by 8 for ADC clock.
[7:3]	**Reserved** These bits are reserved and must be programmed to 00000.

Comparator

Z8 Encore! F083A Series devices feature a general purpose comparator that compares two analog input signals. A GPIO (`CINP`) pin provides the positive comparator input. The negative input (`CINN`) is taken from either an external GPIO pin or from an internal reference. The output is available as an interrupt source or is routed to an external pin using the GPIO multiplex. The comparator includes the following features:

- Positive input is connected to a GPIO pin
- Negative input is connected to either a GPIO pin or an programmable internal reference
- Output is either an interrupt source or an output to an external pin

Operation

One of the comparator inputs is connected to an internal reference, which is a user selectable reference and is user programmable with 200 mV resolution.

The comparator may be powered down to save supply current. For more details, see *the* Power Control Register 0 *section on page 32*.

⚠ **Caution:** As a result of the propagation delay of the comparator, Zilog does not recommend enabling the comparator without first disabling interrupts and waiting for the comparator output to settle. This delay prevents spurious interrupts after comparator enabling.

The following sample code shows how to safely enable the comparator:

```
di
ld cmp0
nop
nop      ; wait for output to settle
clr irq0 ; clear any spurious interrupts pending
ei
```

Comparator Control Register Definition

The Comparator Control Register (CMPCTL) configures the comparator inputs and sets the value of the internal voltage reference. The GPIO pin always used as positive comparator input.

Table 69. Comparator Control Register (CMP0)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	INNSEL	REFLVL				Reserved	
RESET	0	0	0	1	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F90H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] INNSEL	**Signal Select for Negative Input** 0 = internal reference disabled, GPIO pin used as negative comparator input. 1 = internal reference enabled as negative comparator input.
[5:2] REFLVL	**Internal Reference Voltage Level** This reference is independent of the ADC voltage reference. 0000 = 0.0V. 0001 = 0.2V. 0010 = 0.4V. 0011 = 0.6V. 0100 = 0.8V. 0101 = 1.0V (Default). 0110 = 1.2V. 0111 = 1.4V. 1000 = 1.6V. 1001 = 1.8V. 1010–1111 = Reserved.
[1:0]	**Reserved** These bits are reserved and must be programmed to 00.

Flash Memory

The products in the Z8 Encore! F083A Series features either 4KB (4096 bytes with NVDS) or 8KB (8192 bytes with NVDS) of nonvolatile Flash memory with read/write/erase capability. The Flash memory is programmed and erased in-circuit by either user code or through the On-Chip Debugger.

The Flash memory array is arranged in pages with 512-bytes per page. The 512-byte page is the minimum Flash block size that is erased. Each page is divided into eight rows of 64 bytes.

For program/data protection, Flash memory is also divided into sectors. In the Z8 Encore! F083A Series, each sector maps to one page for 4KB devices and two pages for 8KB devices.

The first two bytes of the Flash program memory are used as Flash option bits. For details, see *the* Flash Option Bits chapter on page 124.

Table 70 describes the Flash memory configuration for each device in the Z8 Encore! F083A Series. Figures 14 and 15 display the Flash memory arrangement.

Table 70. Z8 Encore! F083A Series Flash Memory Configurations

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash Sector Size (bytes)
Z8F083A	8 (8196)	16	0000H–1FFFH	1024
Z8F043A	4 (4096)	8	0000H–0FFFH	512

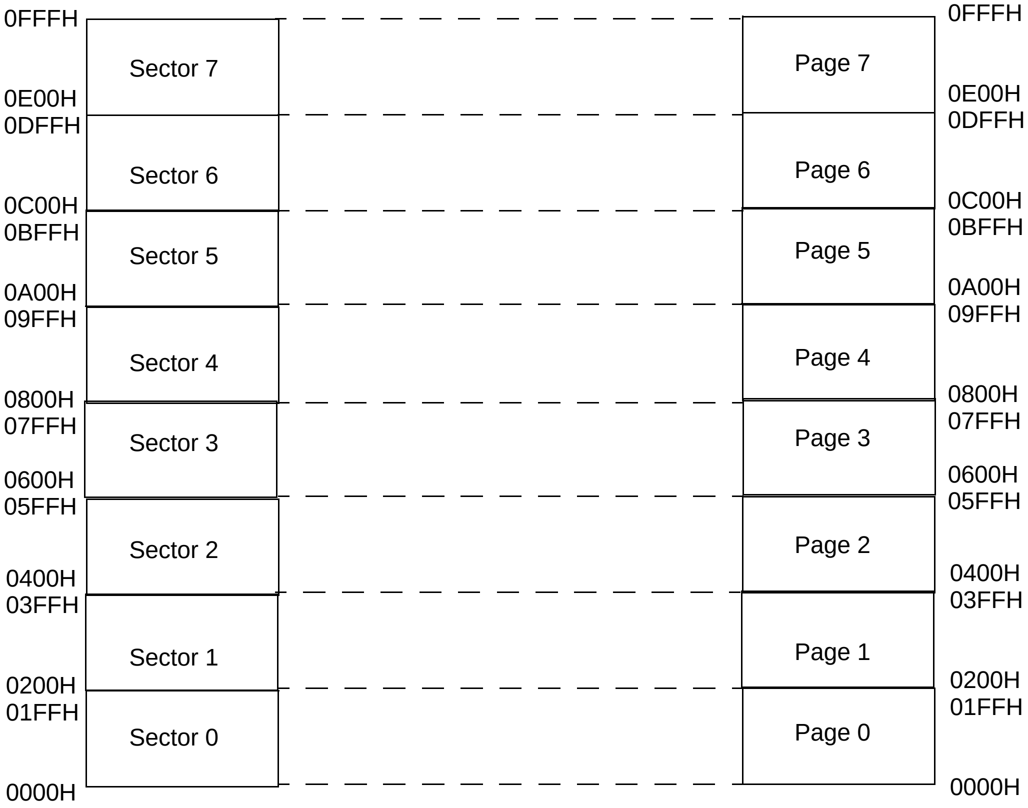


Figure 14. 4K Flash with NVDS

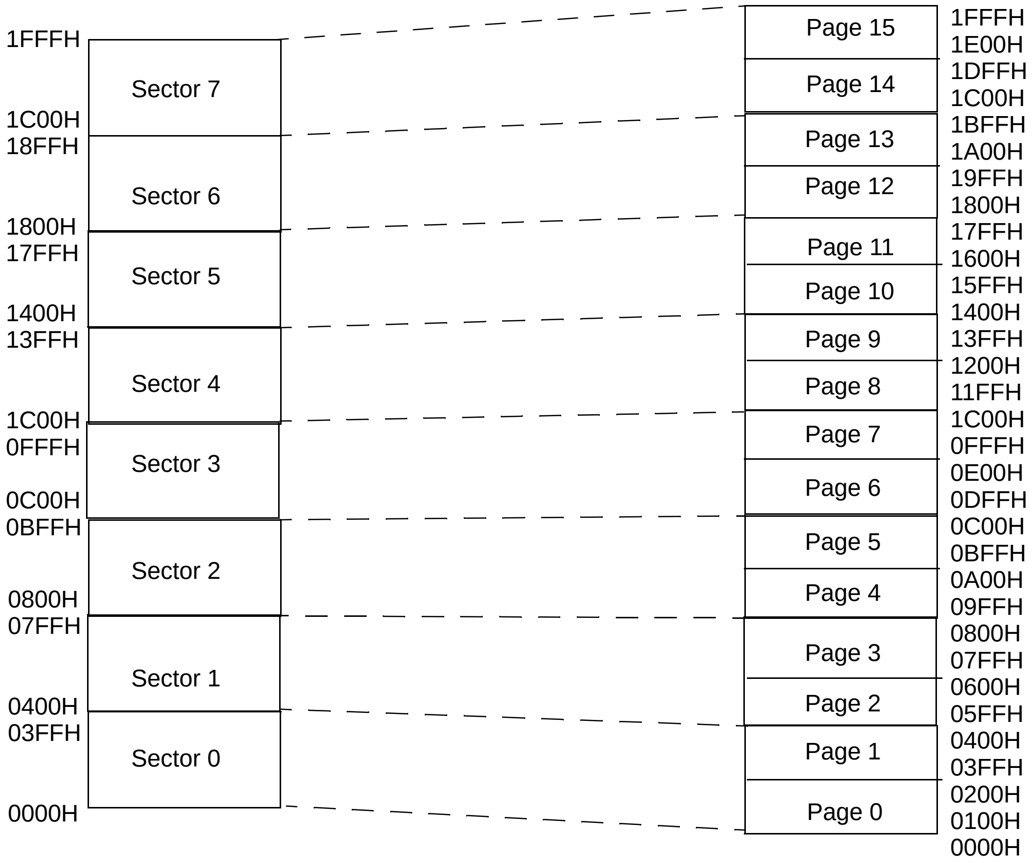


Figure 15. 8K Flash with NVDS

Data Memory Address Space

The Flash information area, including the Zilog Flash option bits, is located in the data memory address space. The Z8 Encore! F083A Series devices are configured by the Zilog Flash option bits to prevent the user from writing to the eZ8 CPU data memory address space.

Flash Information Area

The Flash information area is physically separate from program memory and is mapped to the address range `FE00H` to `FE7FH`. Not all of these addresses are user-accessible. Factory trim values for the VBO and internal precision oscillator, and factory calibration data for the ADC, are stored here.

Table 71 describes the Flash information area. This 128-byte information area is accessed by setting the bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash information area is mapped into program memory and overlays the 128-bytes at the addresses FE00H to FE7FH. When the information area access is enabled, all reads from these program memory addresses return the information area data rather than the program memory data. Access to the Flash information area is read-only.

The trim bits are handled differently than the other Zilog Flash option bits. The trim bits are the hybrid of the user option bits and the standard Zilog option bits. These trim bits must be user accessible for reading at all times using external registers, regardless of the state of bit 7 in the Flash Page Select Register. Writes to the trim space change the value of the option bit holding register, but does not affect the Flash bits, which remain as read-only.

Table 71. Z8F083 Flash Memory Area Map

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog option bits
FE40–FE53	Part number 20-character ASCII alphanumeric code Left-justified and filled with FH
FE54–FE5F	Reserved
FE60–FE7F	Zilog calibration data

Operation

The Flash Controller programs and erases Flash memory, and provides the proper Flash controls and timing for the byte programming, page erase and mass erase operations performed in Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels.

The flow chart in Figure 16 displays basic Flash Controller operation. The subsections that follow provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select Page Erase and Mass Erase) shown in Figure 16.

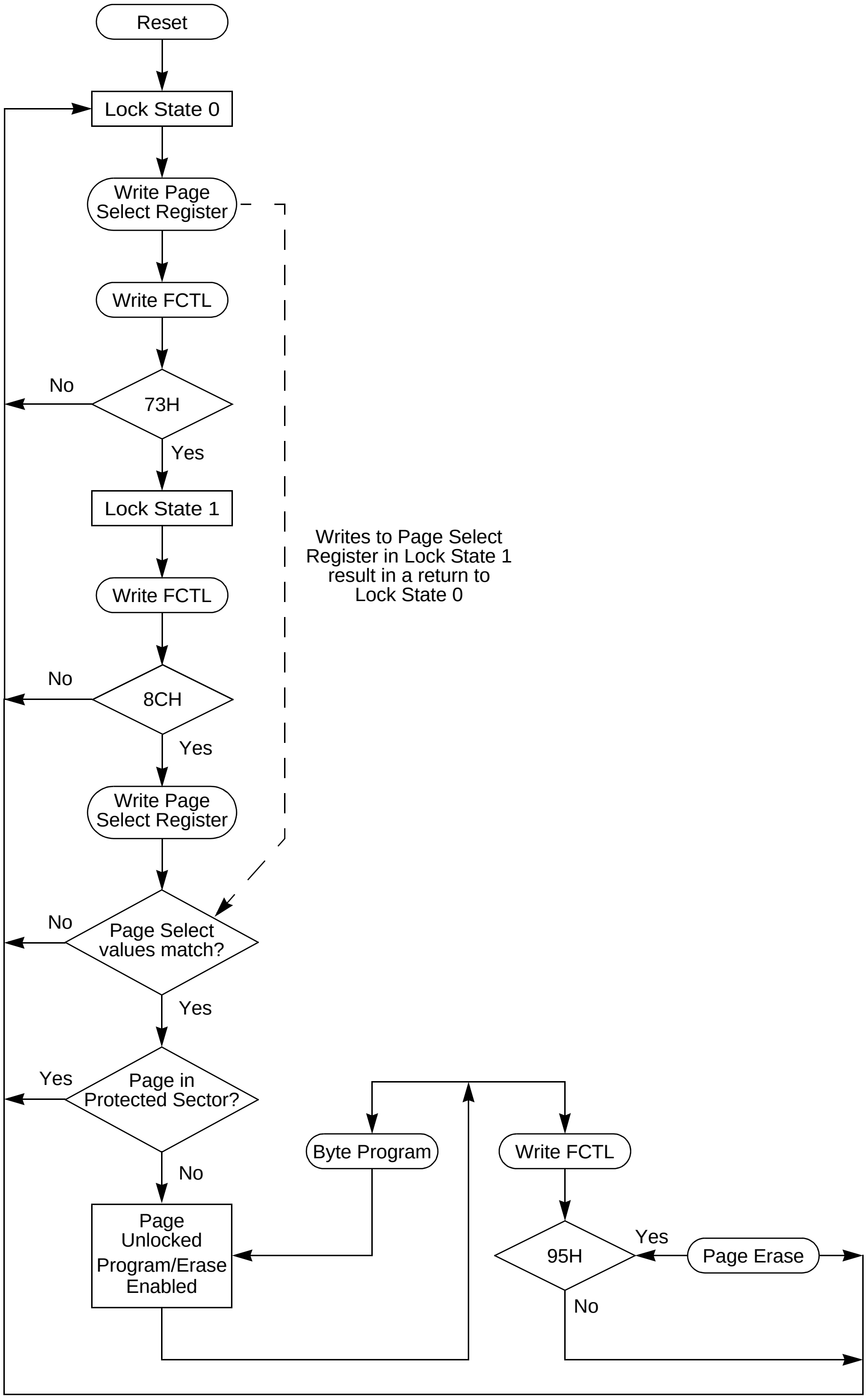

Reset
Lock State 0
Write Page Select Register
Write FCTL
73H
No
Yes
Lock State 1
Write FCTL
8CH
No
Yes
Write Page Select Register
Page Select values match?
No
Yes
Page in Protected Sector?
Yes
No
Page Unlocked Program/Erase Enabled
Byte Program
Write FCTL
95H
Yes
No
Page Erase
Writes to Page Select Register in Lock State 1 result in a return to Lock State 0

Figure 16. Flash Controller Operation Flow Chart

Flash Operation Timing Using the Flash Frequency Registers

Before performing either a Program or Erase operation on Flash memory, the user must first configure the Flash frequency High and Low Byte registers. The Flash frequency registers allow programming and erasing of the Flash with system clock frequencies ranging from 10 kHz to 20 MHz.

The Flash frequency High and Low Byte registers combine to form a 16-bit value, `FFREQ`, to control the timing for Flash Program and Erase operations. The 16-bit binary Flash frequency value must contain the system clock frequency (in kHz). This value is calculated using the following equation:

$$\text{FFREQ[15:0]} = \frac{\text{System Clock Frequency (Hz)}}{1000}$$

Caution: Flash programming and erasure are not supported for system clock frequencies below 10 kHz or above 20 MHz. The Flash frequency High and Low Byte registers must be loaded with the correct value to ensure operation of the Z8 Encore! F083A Series devices.

Flash Code Protection Against External Access

The user code contained within Flash memory is protected against external access by using the On-Chip Debugger. Programming the `FRP` Flash option bit prevents reading of the user code using the On-Chip Debugger. For more details, see the Flash Option Bits chapter on page 124 and the On-Chip Debugger chapter on page 139.

Flash Code Protection Against Accidental Program and Erasure

Z8 Encore! F083A Series devices provide several levels of protection against accidental program and erasure of the Flash memory contents. This protection is provided by a combination of the Flash option bits, the register locking mechanism, the page select redundancy and the sector level protection control of the Flash Controller.

Flash Code Protection using the Flash Option Bits

The FHSWP and FWP Flash option bits combine to provide three levels of Flash program memory protection as listed in Table 72. For more details, see the Flash Option Bits chapter on page 124.

Table 72. Flash Code Protection Using Flash Option Bits

FHSWP	FWP	Flash Code Protection Description
0	0	Programming and erasing disabled for all Flash program memory. In user code programming, page erase and mass erase are all disabled. Mass erase is available through the On-Chip Debugger.
0 or 1	1	Programming, page erase and mass erase are enabled for all of the Flash program memory.

At reset, the Flash Controller is locked to prevent accidental program or erasure of Flash memory. To program or erase Flash memory, first write the target page to the Page Select Register. Unlock the Flash Controller by making two consecutive writes to the Flash Control Register with the values 73H and 8CH, sequentially. The Page Select Register must be rewritten with the same page previously stored there. If the two page select writes do not match, the controller reverts to a Locked state. If the two writes match, the selected page becomes active. For details, see Figure 16 on page 114.

After unlocking a specific page, the user must enable either page program or erase. Writing the value 95H causes a page erase only if the active page resides in a sector that is not protected. Any other value written to the Flash Control Register locks the Flash Controller. Mass erase is not allowed in the user code, but is allowed through the debug port.

After unlocking a specific page, the user must also write to any byte on that page. After a byte is written, the page remains unlocked, allowing for subsequent writes to other bytes on the same page. Further writes to the Flash Control Register causes the active page to revert to a Locked state.

Sector-Based Flash Protection

The final protection mechanism is implemented on a per-sector basis. The Flash memories of Z8 Encore! devices are divided into maximum number of eight sectors. A sector is one-eighth of the total size of Flash memory, unless this value is smaller than the page size, in which case the sector and page sizes are equal. On the Z8 Encore! F083A Series devices, the sector size is varied; see Table 70 and Figures 14 and 15.

The Flash Sector Protect Register can be configured to prevent sectors from being programmed or erased. After a sector is protected, it cannot be unprotected by user code. The Flash Sector Protect Register is cleared after reset and any previously written protection values is lost. User code must write this register in their initialization routine if they want to enable sector protection.

The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register is accessed by writing the Flash Control Register with 5EH. After the Flash Sector Protect Register is selected, it can be accessed at the Page Select Register address. When user code writes the Flash Sector Protect Register, bits can only be set to 1. Thus, sectors can be protected, but not unprotected, via register write operations. Writing a value other than 5EH to the Flash Control Register deselects the Flash Sector Protect Register and reenables access to the Page Select Register. Observe the following procedure to setup the Flash Sector Protect Register from user code:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write 5EH to the Flash Control Register to select the Flash Sector Protect Register.
3. Read and/or write the Flash Sector Protect Register which is now at Register File address FF9H.
4. Write 00H to the Flash Control Register to return the Flash Controller to its reset state.

The Sector Protect Register is initialized to 0 upon reset, putting each sector into an unprotected state. When a bit in the Sector Protect Register is written to 1, the corresponding sector is no longer be written or erased. After setting a bit in the Sector Protect Register, the bit cannot be cleared by the user.

Byte Programming

The Flash memory is enabled for byte programming after unlocking the Flash Controller and successfully enabling either mass erase or page erase. When the Flash Controller is unlocked and mass erase is successfully enabled, all of the program memory locations are available for byte programming. In contrast, when the Flash Controller is unlocked and page erase is successfully enabled, only the locations of the selected page are available for byte programming. An erased Flash byte contains all 1's (FFH). The programming operation is used only to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires execution of either the page erase or mass erase commands.

Byte programming is accomplished using the On-Chip Debugger's write memory command or eZ8 CPU execution of the LDC or LDCI instructions. For the description of the LDC and LDCI instructions, refer to the eZ8 CPU Core User Manual (UM0128), available for download on www.zilog.com. While the Flash Controller programs Flash memory, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. To exit programming mode and lock the Flash, write any value to the Flash Control Register, except the mass erase or page erase commands.

⚠ **Caution:** The byte at each Flash memory address cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs.

Page Erase

The Flash memory is erased one page (512 bytes) at a time. Page erasing Flash memory sets all bytes in that page to the value FFH. The Flash Page Select Register identifies the page to be erased. Only a page residing in an unprotected sector is erased. With the Flash Controller unlocked and the active page set, writing the value 95h to the Flash Control Register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the Page Erase operation completes. If the Page Erase operation is performed using the On-Chip Debugger, poll the Flash status register to determine when the Page Erase operation is complete. When the page erase is complete, the Flash Controller returns to its Locked state.

Mass Erase

The Flash memory is also mass erased using the Flash Controller, but only by using the On-Chip Debugger. Mass erasing Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked and the mass erase successfully enabled, writing the value 63H to the Flash Control Register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. Using the On-Chip Debugger, poll the Flash status register to determine when the Mass Erase operation is complete. When the mass erase is complete, the Flash Controller returns to its Locked state.

Flash Controller Bypass

The Flash Controller is bypassed and the control signals for Flash memory are brought out to the GPIO pins. Bypassing the Flash Controller allows faster row programming algorithms by controlling the Flash programming signals directly.

Row programing is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of Flash memory. Mass Erase and Page Erase operations are also supported, when the Flash Controller is bypassed.

For more information about bypassing the Flash Controller, refer to the Third Party Flash Programming Support for Z8 Encore! MCUs Application Note. This document is available for download at www.zilog.com.

Flash Controller Behavior in Debug Mode

The following behavioral changes are observed in the Flash Controller when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash write protect option bit is ignored.

- The Flash Sector Protect Register is ignored for programming and Erase operations.
- Programming operations are not limited to the page selected in the Page Select Register.
- Bits in the Flash Sector Protect Register are written to one or zero.
- The second write of the Page Select Register to unlock the flash controller is not necessary.
- The Page Select Register is written when the Flash Controller is unlocked.
- The mass erase command is enabled through the Flash Control Register

Caution: For security reasons, Flash Controller allows only a single page to be opened for write/ erase. When writing multiple Flash pages, the Flash Controller must go through the unlock sequence again to select another page.

NVDS Operational Requirements

The device uses a 12KB Flash memory, despite the maximum specified Flash of 8KB size (except 12KB mode with non-NVDS). User code accesses the lower 8KB of flash, leaving the upper 4 K for Zilog memory. The NVDS is implemented by using Zilog memory for special purpose routines and for the data required by the routines. These routines are factory programmed and cannot be altered by the user. The NVDS operation is described in detail. See the Nonvolatile Data Storage chapter on page 134.

The NVDS routines are triggered by a user code: CALL into Zilog memory. Code executing from Zilog memory must be able to read and write other locations within Zilog memory. User code must not be able to read or write Zilog memory.

Flash Control Register Definitions

This section defines the features of the following Flash Control registers.

Flash Control Register: see page 120

Flash Status Register: see page 121

Flash Page Select Register: see page 121

Flash Sector Protect Register: see page 122

Flash Frequency High and Low Byte Registers: see page 123

Flash Control Register

The Flash Controller must be unlocked using the Flash Control Register before programming or erasing Flash memory. Writing the sequence `73H 8CH`, sequentially, to the Flash Control Register unlocks the Flash Controller. When the Flash Controller is unlocked, Flash memory is enabled for mass erase or page erase by writing the appropriate enable command to the FCTL. *Page erase* applies only to the active page selected in the Flash Page Select Register. Mass erase is enabled only through the On-Chip Debugger. Writing an invalid value or an invalid sequence returns the Flash Controller to its Locked state. The write-only Flash Control Register, shown in Table 73, shares its Register File address with the read-only Flash Status Register.

Table 73. Flash Control Register (FCTL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	FCMD							
RESET	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	FF8H							

Bit	**Description**
[7:0] FCMD	**Flash Command** 73H = First unlock command. 8CH = Second unlock command. 95H = Page erase command (must be third command in sequence to initiate page erase). 63H = Mass erase command (must be third command in sequence to initiate mass erase). 5EH = Enable Flash Sector Protect Register access.

Flash Status Register

The Flash Status Register, shown in Table 74, indicates the current state of the Flash Controller. This register is read at any time. The read-only Flash status register shares its Register File address with the write-only Flash Control Register.

Table 74. Flash Status Register (FSTAT)

Bit	7	6	5	4	3	2	1	0
Field	Reserved		FSTAT					
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	FF8H							

Bit	Description
[7:6]	**Reserved** These bits are reserved and must be programmed to 00.
[5:0] FSTAT	**Flash Controller Status** 000000 = Flash Controller locked. 000001 = First unlock command received (73H written). 000010 = Second unlock command received (8CH written). 000011 = Flash Controller unlocked. 000100 = Sector protect register selected. 001xxx = Program operation in progress. 010xxx = Page Erase operation in progress. 100xxx = Mass Erase operation in progress.

Flash Page Select Register

The Flash Page Select Register, shown in Table 75, shares address space with the Flash Sector Protect Register. Unless the Flash Controller is locked and written with 5EH, any writes to this address will target the Flash Page Select Register.

This register selects one of the eight available Flash memory pages to be programmed or erased. Each Flash page contains 512-bytes of Flash memory. During a Page Erase operation, all Flash memory having addresses with the most significant 7 bits given by FPS[6:0] are chosen for Program/Erase operation.

Table 75. Flash Page Select Register (FPS)

Bit	7	6	5	4	3	2	1	0
Field	INFO_EN	PAGE						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF9H							

Bit	Description
[7] INFO_EN	**Information Area Enable** 0 = Information area is not selected. 1 = Information area is selected. The information area is mapped into the program memory address space at addresses FE00H through FFFFH.
[6:0] PAGE	**Page Select** This 7-bit field identifies the Flash memory page for page erase and page unlocking. Program memory address[15:9] = PAGE[6:0]. For Z8F04xx devices, the upper four bits must always be 0.

Flash Sector Protect Register

The Flash Sector Protect Register, shown in Table 76, is shared with the Flash Page Select Register. When the Flash Control Register is locked and written with 5EH, the next write to this address targets the Flash Sector Protect Register. In all other cases, it targets the Flash Page Select Register.

This register selects one of the eight available Flash memory sectors to be protected. The Reset state of each sector protect bit is a zero (unprotected) state. After a sector is protected by setting its corresponding register bit, the register bit cannot be cleared by the user.

Table 76. Flash Sector Protect Register (FPROT)

Bit	7	6	5	4	3	2	1	0
Field	SPROT7	SPROT6	SPROT5	SPROT4	SPROT3	SPROT2	SPROT1	SPROT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF9H							

Bit	Description
[7:0] SPROTx	**Sector Protection** To determine the appropriate Flash memory sector address range and sector number for your F083A Series product, please refer to Table 70 and to Figures 14 and 15.

Note: x indicates register bits 7–0.

Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte registers, shown in Tables 77 and 78, combine to form a 16-bit value, FFREQ, to control timing for Flash Program and Erase operations. The 16-bit binary Flash frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation.

$$\text{FFREQ[15:0]} = \{\text{FFREQH[7:0]},\text{FFREQL[7:0]}\} = \frac{\text{System Clock Frequency}}{1000}$$

Caution: Flash programming and erasure is not supported for system clock frequencies below 10 kHz or above 20 MHz. The Flash frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device.

Table 77. Flash Frequency High Byte Register (FFREQH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	FFREQH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FFAH							

Bit	**Description**
[7:0] FFREQH	**Flash Frequency High Byte** The high byte of the 16-bit Flash frequency value.

Table 78. Flash Frequency Low Byte Register (FFREQL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	FFREQL							
RESET	0							
R/W	R/W							
Address	FFBH							

Bit	**Description**
[7:0] FFREQL	**Flash Frequency Low Byte** The low byte of the 16-bit Flash frequency value.

Flash Option Bits

Programmable Flash option bits allow users to configure certain aspects of Z8 Encore! F083A Series operation. The configuration data is stored in Flash program memory and read during reset. These Flash option bits control the following functions:

- Watchdog Timer time-out response selection at interrupt or system reset
- Watchdog Timer enable at reset
- The ability to prevent unwanted read access to user code in program memory
- The ability to prevent accidental programming and erasure of all or a portion of the user code in program memory
- VBO configuration, which is always enabled or disabled during STOP Mode to reduce STOP Mode power consumption
- OSCILLATOR mode selection for high, medium and low power crystal oscillators or external RC oscillator
- Factory trimming information for the internal precision oscillator and VBO voltage

Operation

This section describes the type and configuration of the programmable Flash option bits.

Option Bit Configuration by Reset

Each time the Flash option bits are programmed or erased, the device must be reset for the change to be effective. During any Reset operation (system reset or Stop Mode Recovery), the Flash option bits are automatically read from the Flash program memory and written to option configuration registers. The option configuration registers, control the operation of the devices within the Z8 Encore! F083A Series. Option bit control is established before the device exits reset and the eZ8 CPU begins code execution. The option configuration registers are not part of the Register File and are not accessible for read or write access.

Option Bit Types

This section describes the two types of Flash option bits offered in the F083A Series.

User Option Bits

The user option bits are contained in the first two bytes of program memory. User access to these bits is provided because these locations contain application-specific device configurations. The information contained in these first two bytes is lost when page 0 of program memory is erased.

Trim Option Bits

The trim option bits are contained in the information page of Flash memory. These bits are factory programmed values required to optimize the operation of onboard analog circuitry and cannot be permanently altered by the user. Program memory may be erased without endangering these values. It is possible to alter working values of these bits by accessing the trim bit address and data registers, but these working values are lost after a power loss.

There are 32-bytes of trim data. To modify one of these values, the user code must first write a value between `00H` and `1FH` into the trim bit Address Register. The next write to the trim bit data register changes the working value of the target trim data byte.

Reading the trim data requires the user code to write a value between `00H` and `1FH` into the trim bit Address Register. The next read from the trim bit data register returns the working value of the target trim data byte.

Note: The trim address range is from information address `20–3F` only. The remainder of the information page is not accessible through the trim bit address and data registers.

During reset, the first 43 system clock cycles perform 43 Flash accesses. The six bits of the counter provide the lower six bits of the Flash memory address. All other address bits are set to 0. The option bit registers use the 6-bit address from the counter as an address and latch the data from the Flash on the positive edge of the IPO clock, allowing for a maximum of 344 bits (43 bytes) of option information to be read from Flash memory.

Because option information is stored in both the first two bytes of program memory and in the information area of Flash memory, the data must be placed in specific locations to be read correctly. In this case, the first two bytes at address 0 and 1 in program memory are read and the remainder of the bytes are read from the Flash information area.

Flash Option Bit Control Register Definitions

This section briefly describes the features of the Trim Bit Address and Data registers.

Trim Bit Address Register

The Trim Bit Address Register, shown in Table 79, contains the target address to access the trim option bits.Trim bit addresses in the range (`00H-1FH`) map to the information area addresses (`20H` to `3FH`) listed in Table 80.

Table 79. Trim Bit Address Register (TRMADR)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TRMADR: Trim Bit Address (00H to 1FH)							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF6H							

Table 80. Trim Bit Address Map

Trim Bit Address	Information Area Address
00H	20H
01H	21H
02H	22H
03H	23H
:	:
1FH	3FH

Trim Bit Data Register

The Trim Bit Data Register, shown in Table 81, contains the read or write data to access the trim option bits.

Table 81. Trim Bit Data Register (TRMDR)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	TRMDR: Trim Bit Data							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF7H							

Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses `0000H` and `0001H` are reserved for the user-programmable Flash option bits.

Flash Program Memory Address 0000H

Table 82. Flash Option Bits at Program Memory Address 0000H

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	WDT_RES	WDT_AO	OSC_SEL[1:0]		VBO_AO	FRP	Reserved	FWP
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Program Memory 0000H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	**Description**
[7] WDT_RES	**Watchdog Timer Reset** 0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request. 1 = Watchdog Timer time-out causes a system reset. This is the default setting for unprogrammed (erased) Flash.
[6] WDT_AO	**Watchdog Timer Always On** 0 = On application of system power, Watchdog Timer is automatically enabled. Watchdog Timer cannot be disabled. 1 = Watchdog Timer is enabled on execution of the WDT instruction. Once enabled, the Watchdog Timer is disabled only by a reset. This is the default setting for unprogrammed (erased) Flash.

Bit	Description (Continued)
[5:4] OSC_SEL	**OSCILLATOR Mode Selection** 00 = On-chip oscillator configured for use with external RC networks (<4 MHz). 01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz). 10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 5.0 MHz). 11 = Maximum power for use with high frequency crystals (5.0 MHz to 20.0 MHz). This is the default setting for unprogrammed (erased) Flash.
[3] VBO_AO	**Voltage Brown-Out Protection Always On** 0 = VBO protection is disabled in STOP Mode to reduce total power consumption. 1 = VBO protection is always enabled, even during STOP Mode. This is the default setting for unprogrammed (erased) Flash.
[2] FRP	**Flash Read Protect** 0 = User program code is inaccessible. Limited control features are available through the On-Chip Debugger. 1 = User program code is accessible. All On-Chip Debugger commands are enabled. This is the default setting for unprogrammed (erased) Flash.
[1]	**Reserved** This bit is reserved and must be programmed to 1.
[0] FWP	**Flash Write Protect** This option bit provides Flash program memory protection. 0 = Programming and erasure disabled for all Flash program memory. Programming, page erase and mass erase through user code is disabled. Mass erase is available using the On-Chip Debugger. 1 = Programming, page erase and mass erase are enabled for all Flash program memory.

Flash Program Memory Address 0001H

Table 83. Flash Options Bits at Program Memory Address 0001H

Bit	7	6	5	4	3	2	1	0
Field	VBO_RES	Reserved		XTLDIS	Reserved			
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Program Memory 0001H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description
[7] VBO_RES	**Voltage Brown-Out Reset** 1 = VBO detection causes a system reset. This setting is the default setting for unprogrammed (erased) Flash.
[6:5]	**Reserved** These bits are reserved and must be programmed to 11.
[4] XTLDIS	**State of the Crystal Oscillator at Reset** This bit only enables the crystal oscillator. Its selection as a system clock must be performed manually. 0 = The crystal oscillator is enabled during reset, resulting in longer reset timing. 1 = The crystal oscillator is disabled during reset, resulting in shorter reset timing.
[3:0]	**Reserved** These bits are reserved and must be programmed to 1111.

Trim Bit Address Space

Table 84. Trim Bit Address Space

Address	Function
00h	ADC reference voltage
01h	ADC and comparator
02h	Internal precision oscillator
03h	Oscillator and VBO
06h	ClkFltr

Trim Bit Address 0000H

Table 85. Trim Option Bits at 0000H (ADCREF)

Bit	7	6	5	4	3	2	1	0
Field	ADCREF_TRIM					Reserved		
RESET	U					U		
R/W	R/W					R/W		
Address	Information Page Memory 0020H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description
[7:3] ADCREF_TRIM	**ADC Reference Voltage Trim Byte** Contains trimming bits for the ADC reference voltage.
[2:0]	**Reserved** These bits are reserved and must be programmed to 111.

Note: The bit values indicated in Table 85 are set at the factory; no calibration is required.

Trim Bit Address 0001H

Table 86. Trim Option Bits at 0001H (TADC_COMP)

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0021H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description
[7:0]	**Reserved** These bits are reserved. Altering this register can result in incorrect device operation

Note: The bit values used in Table 86 are set at the factory; no calibration is required.

Trim Bit Address 0002H

Table 87. Trim Option Bits at 0002H (TIPO)

Bit	7	6	5	4	3	2	1	0
Field	IPO_TRIM							
RESET	U							
R/W	R/W							
Address	Information Page Memory 0022H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description
[7:0] IPO_TRIM	**Internal Precision Oscillator Trim Byte** Contains trimming bits for internal precision oscillator

Note: The bit values used in Table 87 are set at the factory; no calibration is required.

Trim Bit Address 0003H

Table 88. Trim Option Bits at 0003H (TVBO)

Bit	7	6	5	4	3	2	1	0
Field	Reserved					VBO_TRIM		
RESET	U					U		
R/W	R/W					R/W		
Address	Information Page Memory 0023H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description
[7:3]	**Reserved** These bits are reserved and must be programmed to 1111.
[2:0] VBO_TRIM	**VBO Trim Values** Contains factory-trimmed values for the oscillator and the VBO. See Table 89.

Note: The bit values used in Table 88 are set at the factory; no calibration is required.

Table 89. VBO Trim Definition

VBO_TRIM	Trigger Voltage Level
000	1.7
001	1.6
101	2.2
110	2.0
100	2.4
111	1.8

The F083A Series' on-chip Flash memory only guarantees write operations with a voltage supply over 2.7V. Write operations below 2.7V may cause unpredictable results.

Trim Bit Address 0006H

Table 90. Trim Option Bits at 0006H (TCLKFLT)

Bit	7	6	5	4	3	2	1	0
Field	DivBy4	Reserved	DlyCtl1	DlyCtl2	DlyCtl3	Reserved	FilterSel1	FilterSel0
RESET	0	1	0	0	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0026H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description
[7] DivBy4	**Output Frequency Selection** 0 = Output frequency is input frequency. 1 = Output frequency is 1/4 of the input frequency.
[6]	**Reserved** This bit is reserved and must be programmed to 1.
[5:3] DlyCtl*x*	**Delay Control** Filtered 3-bit pulse width selection. For 3.3V operation, see Table 91.
Notes: x indicates bit values 3–1; y indicates bit values 1–0.	

Bit	Description (Continued)
[2]	**Reserved** This bit is reserved and must be programmed to 1.
[1:0] FilterSel*y*	**Filter Selection** 2-bit Clock Filter Mode selection. 00 = No filter. 01 = Filter low level noise on high level signal. 10 = Filter high level noise on low level signal. 11 = Filter both.
Notes: x indicates bit values 3–1; y indicates bit values 1–0.	

Note: The bit values used in Table 90 are set at the factory; no calibration is required.

Table 91. ClkFlt Delay Control Definition

DlyCtl3, DlyCtl2, DlyCtl1	Low-Noise Pulse on High Signal (ns)	High-Noise Pulse on Low Signal (ns)
000	5	5
001	7	7
010	9	9
011	11	11
100	13	13
101	17	17
110	20	20
111	25	25
Note: The variation is about 30%.		

Nonvolatile Data Storage

Z8 Encore! F083A Series devices contain a Nonvolatile Data Storage (NVDS) element of up to 100 bytes. This type of memory can perform over 100,000 write cycles.

Operation

The NVDS is implemented by special-purpose Zilog software, which is stored in areas of program memory that are not accessible to the user. These special-purpose routines use Flash memory to store data. These routines incorporate a dynamic addressing scheme to maximize the write/erase endurance of Flash memory.

Note: Different members of the Z8 Encore! F083A Series feature multiple NVDS array sizes. For more details, see the Z8 Encore! F083A Series Family Part Selection Guide section on page 2.

NVDS Code Interface

Two routines are required to access the NVDS: a write routine and a read routine. Both of these routines are accessed with a `CALL` instruction to a predefined address outside of program memory that is accessible to the user. Both the NVDS address and data are single-byte values. In order to not disturb the user code, these routines save the working register set before using it; therefore, 16 bytes of stack space are required to preserve the site. After finishing the call to these routines, the working register set of the user code is recovered.

During both read and write accesses to the NVDS, interrupt service is not disabled. Any interrupts that occur during NVDS execution must not disturb the working register and existing stack contents; otherwise, the array can become corrupted. Zilog recommends the user disable interrupts before executing NVDS operations.

Use of the NVDS requires 16 bytes of available stack space. The contents of the working register set are saved before calling NVDS read or write routines.

For correct NVDS operation, the Flash Frequency registers must be programmed based on the system clock frequency. See the Flash Operation Timing Using the Flash Frequency Registers section on page 115.

Byte Write

To write a byte to the NVDS array, the user code must first push the address, then the data byte onto the stack. The user code issues a `CALL` instruction to the address of the Byte Write routine (`0x20B3`). At the return from the sub-routine, the write status byte resides in working register R0. The bit fields of this status byte are defined in Table 92. Additionally, user code should pop the address and data bytes off the stack.

The write routine uses 16 bytes of stack space in addition to the two bytes of address and data pushed by the user code. Sufficient memory must be available for this stack usage.

Because of the flash memory architecture, NVDS writes exhibit a nonuniform execution time. In general, a write takes 136 µs (assuming a 20MHz system clock). For every 200 writes, however, a maintenance operation is necessary. In this rare occurrence, the write takes up to 58 ms to complete. Slower system clock speeds result in proportionally higher execution times.

NVDS byte writes to invalid addresses (those exceeding the NVDS array size) have no effect. Illegal write operations have a 7µs execution time.

Table 92. Write Status Byte

Bit	7	6	5	4	3	2	1	0
Field	Reserved					FE	IGADDR	WE
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:3]	**Reserved** These bits are reserved and must be programmed to 00000.
[2] FE	**Flash Error** If a Flash error is detected, this bit is set to 1.
[1] IGADDR	**Illegal Address** When NVDS byte writes to invalid addresses occur (those exceeding the NVDS array size), this bit is set to 1.
[0] WE	**Write Error** A failure occurs during writing data into Flash. When writing data into a certain address, a read back operation is performed. If the read back value is not the same as the value written, this bit is set to 1.

Byte Read

To read a byte from the NVDS array, user code must first push the address onto the stack. User code issues a `CALL` instruction to the address of the byte-read routine (`0x2000`). At the return from the sub-routine, the read byte resides in working register R0 and the read status byte resides in working register R1. The bit fields of this status byte are defined in Table 93. Additionally, the user code should pop the address byte off the stack.

The read routine uses 16 bytes of stack space in addition to the one byte of address pushed by the user code. Sufficient memory must be available for this stack usage.

Because of the Flash memory architecture, NVDS reads exhibit a nonuniform execution time. A read operation takes between 71 µs and 258 µs (assuming a 20MHz system clock). Slower system clock speeds result in proportionally higher execution times.

NVDS byte reads from invalid addresses (those exceeding the NVDS array size) return `0xff`. Illegal read operations have a 6 µs execution time.

The status byte returned by the NVDS read routine is zero for successful read. If the status byte is nonzero, there is a corrupted value in the NVDS array at the location being read. In this case, the value returned in R0 is the byte most recently written to the array that does not have an error.

Table 93. Read Status Byte

Bit	7	6	5	4	3	2	1	0
Field	Reserved			DE	Reserved	FE	IGADDR	Reserved
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 000.
[4] DE	**Data Error** When reading a NVDS address, if an error is found in the latest data corresponding to this NVDS address, this bit is set to 1. NVDS source code steps forward until finding a valid data at this address.
[3]	**Reserved** This bit is reserved and must be programmed to 0.
[2] FE	**Flash Error** If a Flash error is detected, this bit is set to 1.

Bit	Description (Continued)
[1] IGADDR	**Illegal Address** When NVDS byte reads from invalid addresses occur (those exceeding the NVDS array size), this bit is set to 1.
[0]	**Reserved** This bit is reserved and must be programmed to 0.

Power Failure Protection

The NVDS routines employ error-checking mechanisms to ensure that any power failure will only endanger the most recently written byte. Bytes previously written to the array are not perturbed. For this protection to function, the VBO must be enabled (See *the* Low-Power Modes chapter on page 30) and configured for a threshold voltage of 2.4V or greater (See *the* Trim Bit Address Space *section on page 129*).

A system reset (such as a pin reset or Watchdog Timer reset) that occurs during a write operation also perturbs the byte currently being written. All other bytes in the array are unperturbed.

Optimizing NVDS Memory Usage for Execution Speed

As Table 94 shows, NVDS read times vary drastically, this discrepancy being a trade-off for minimizing the frequency of writes that require post-write page erases. The NVDS read time of address N is a function of the number of writes to addresses other than N since the most recent write to address N, as well as the number of writes since the most recent page erase. Neglecting effects caused by page erases and results caused by the initial condition in which the NVDS is blank, a rule of thumb is that every write since the most recent page erase causes read times of unwritten addresses to increase by 0.8µs, up to a maximum of 258µs.

Table 94. NVDS Read Time

Operation	Minimum Latency (µs)	Maximum Latency (µs)
Read	71	258
Write	126	136
Illegal Read	6	6
Illegal Write	7	7

Note: For every 200 writes, a maintenance operation is necessary. In this rare occurrence, the write takes up to 58ms to complete.

If NVDS read performance is critical to your software architecture, you can optimize your code for speed by using either of the two methods listed below.

- Periodically refresh all addresses that are used; this is the more useful method. The optimal use of NVDS in terms of speed is to rotate the writes evenly among all addresses planned for use, thereby bringing all reads closer to the minimum read time. Because the minimum read time is much less than the write time, however, actual speed benefits are not always realized.
- Use as few unique addresses as possible to optimize the impact of refreshing.

On-Chip Debugger

Z8 Encore! devices contain an integrated On-Chip Debugger (OCD) that provides the following advanced debugging features:

- Reading and writing of the Register File
- Reading and writing of program and data memory
- Setting of breakpoints and watchpoints
- Executing eZ8 CPU instructions

Architecture

The On-Chip Debugger consists of four primary functional blocks: transmitter, receiver, autobaud detector/generator and debug controller. Figure 17 displays the architecture of the On-Chip Debugger.

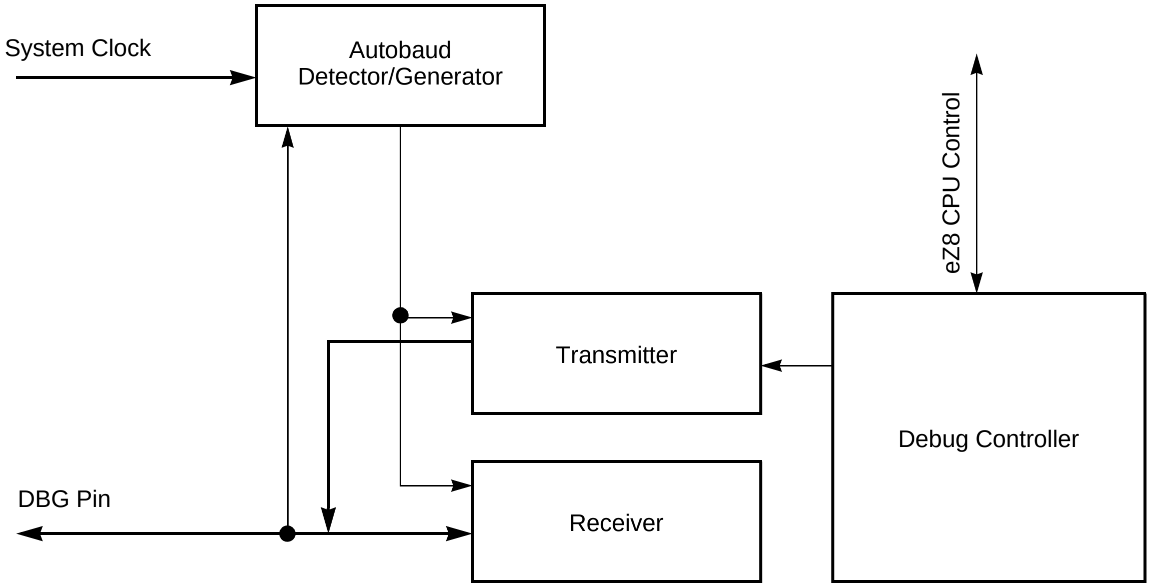


Figure 17. On-Chip Debugger Block Diagram

Operation

The following section describes the operation of the On-Chip Debugging function.

OCD Interface

The On-Chip Debugger uses the DBG pin for communication with an external host. This one-pin interface is a bidirectional open-drain interface that transmits and receives data. Data transmission is half-duplex, which means transmission and data retrieval cannot occur simultaneously. The serial data on the DBG pin is sent using the standard asynchronous data format defined in RS-232.This pin creates an interface between the Z8 Encore! F083A Series products and the serial port of a host PC using minimal external hardware.Two different methods for connecting the DBG pin to an RS-232 interface are displayed in Figures 18 and 19. The recommended method is the buffered implementation depicted in Figure 19. The DBG pin must always be connected to V_{DD} through an external pull-up resistor.

Caution: For operation of the On-Chip Debugger, all power pins (V_{DD} and AV_{DD}) must be supplied with power and all ground pins (V_{SS} and AV_{SS}) must be properly grounded. The DBG pin is open-drain and must always be connected to V_{DD} through an external pull-up resistor to ensure proper operation.

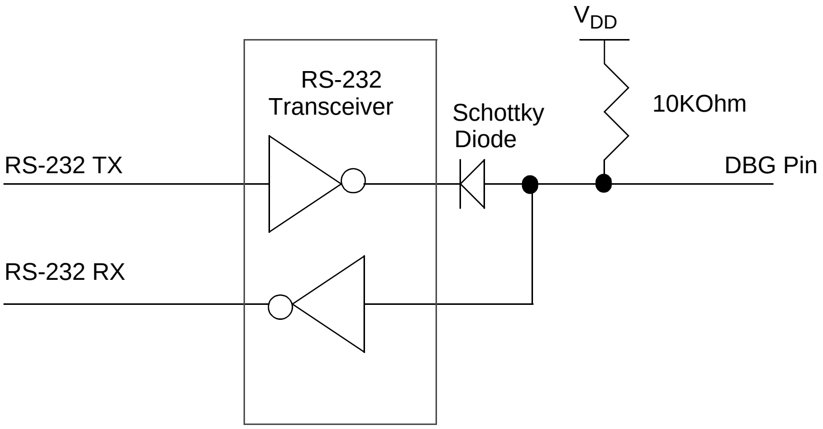


Figure 18. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 1 of 2

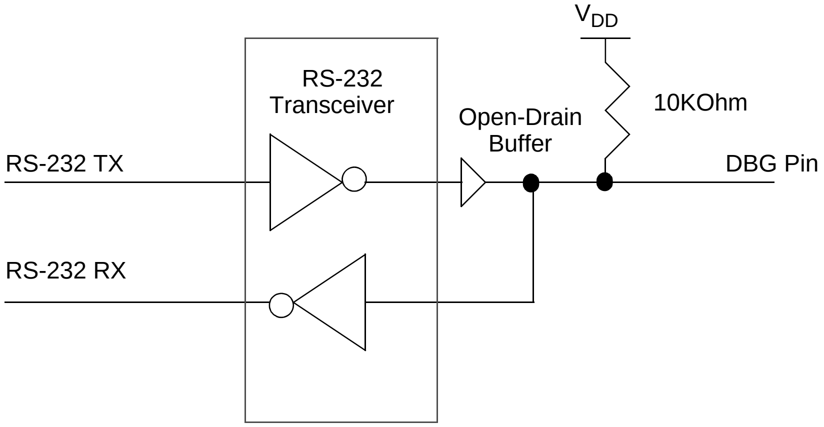


Figure 19. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #2 of 2

DEBUG Mode

F083A Series devices, when in DEBUG Mode, feature the following operating characteristics:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions
- The system clock operates, unless the device is in STOP Mode
- All enabled on-chip peripherals operate, unless the device is in STOP Mode
- Automatically exits HALT Mode
- Constantly refreshes the Watchdog Timer, if enabled

Entering DEBUG Mode

- The device enters DEBUG Mode after the eZ8 CPU executes a Breakpoint (BRK) instruction
- If the DBG pin is held Low during the most recent system reset clock cycle, the device enters DEBUG Mode upon exiting system reset

Exiting DEBUG Mode

The device exits DEBUG Mode upon any of the following operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset

- Watchdog Timer reset
- Asserting the $\overline{\text{RESET}}$ pin Low to initiate a reset
- Driving the DBG pin Low while the device is in STOP Mode initiates a system reset

OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 start bit, 8 data bits (least-significant bit first) and 1 stop bit. See Figure 20.

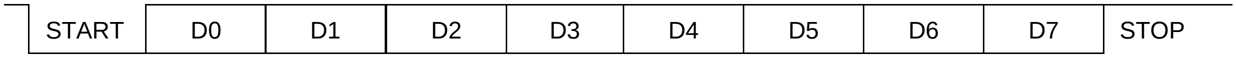


Figure 20. OCD Data Format

OCD Autobaud Detector/Generator

To run over a range of baud rates (data bits per second) with various system clock frequencies, the On-Chip Debugger contains an autobaud detector/generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the host is the character `80H`. The character `80H` has eight continuous bits low (one Start bit plus 7 data bits), framed between high bits. The autobaud detector measures this period and sets the OCD baud rate generator accordingly.

The autobaud detector/generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. For optimal operation with asynchronous datastreams, the maximum recommended baud rate is the system clock frequency divided by 8. The maximum possible baud rate for asynchronous datastreams is the system clock frequency divided by 4, but this theoretical maximum is possible only for low noise designs with clean signals. Table 95 lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 95. OCD Baud-Rate Limits

System Clock Frequency (MHz)	Recommended Maximum Baud Rate (kbps)	Recommended Standard PC Baud Rate (bps)	Minimum Baud Rate (kbps)
20.0	2500.0	1,843,200	39
1.0	125.0	115,200	1.95
0.032768 (32 kHz)	4.096	2400	0.064

If the OCD receives a serial break (nine or more continuous bits low), the autobaud detector/generator resets. Reconfigure the autobaud detector/generator by sending 80H.

OCD Serial Errors

The On-Chip Debugger detects any of the following error conditions on the DBG pin:

- Serial break (a minimum of nine continuous bits Low)
- Framing error (received Stop bit is Low)
- Transmit collision (simultaneous transmission by OCD and host detected by the OCD)

When the OCD detects one of these errors, it aborts any command currently in progress, transmits a four character long serial break back to the host and resets the autobaud detector/generator. A framing error or transmit collision may be caused by the host sending a serial break to the OCD. As a result of the open-drain nature of the interface, returning a serial break back to the host only extends the length of the serial break if the host releases the serial break early.

The host transmits a serial break on the DBG pin when first connecting to the Z8 Encore! F083A Series devices or when recovering from an error. A serial break from the host resets the autobaud generator/detector, but does not reset the OCD Control Register. A serial break leaves the device in DEBUG Mode, if that is the current mode. The OCD is held in reset until the end of the serial break when the DBG pin returns high. Because of the open-drain nature of the DBG pin, the host sends a serial break to the OCD even if the OCD is transmitting a character.

Breakpoints

Execution breakpoints are generated using the BRK instruction (Opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. If breakpoints are enabled, the OCD enters DEBUG Mode and idles the eZ8 CPU. If breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

Breakpoints in Flash Memory

The BRK instruction is Opcode 00H, which corresponds to the fully programmed state of a byte in Flash memory. To implement a breakpoint, write 00H to the required break address overwriting the current instruction. To remove a breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

Runtime Counter

The On-Chip Debugger contains a 16-bit runtime counter. It counts system clock cycles between breakpoints. The counter starts counting when the On-Chip Debugger leaves DEBUG Mode and stops counting when it enters DEBUG Mode again or when it reaches the maximum count of FFFFH.

On-Chip Debugger Commands

The host communicates to the On-Chip Debugger by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG Mode, all OCD commands become available unless the user code and control registers are protected by programming the Flash read protect option bit (FRP). The FRP prevents the code in memory from being read out of the Z8 Encore! F083A Series products. When this option is enabled, several of the OCD commands are disabled. Table 96 offers a summary of the On-Chip Debugger commands that operate when the device is not in DEBUG Mode (normal operation) and the commands that are disabled by programming the FRP.

Table 96. On-Chip Debugger Command Summary

Debug Command	Command Byte	Enabled when not in DEBUG Mode?	Disabled by Flash Read Protect Option Bit
Read OCD Revision	00H	Yes	–
Reserved	01H	–	–
Read OCD Status Register	02H	Yes	–
Read Runtime Counter	03H	–	–
Write OCD Control Register	04H	Yes	Cannot clear DBGMODE bit
Read OCD Control Register	05H	Yes	–
Write Program Counter	06H	–	Disabled
Read Program Counter	07H	–	Disabled
Write Register	08H	–	Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control register.
Read Register	09H	–	Disabled
Write Program Memory	0AH	–	Disabled
Read Program Memory	0BH	–	Disabled
Write Data Memory	0CH	–	Yes
Read Data Memory	0DH	–	–

Table 96. On-Chip Debugger Command Summary (Continued)

Debug Command	Command Byte	Enabled when not in DEBUG Mode?	Disabled by Flash Read Protect Option Bit
Read Program Memory CRC	0EH	–	–
Reserved	0FH	–	–
Step Instruction	10H	–	Disabled
Stuff Instruction	11H	–	Disabled
Execute Instruction	12H	–	Disabled
Reserved	13H–FFH	–	–

In the following list of OCD commands, data and commands sent from the host to the On-Chip Debugger are identified by `DBG ← Command/Data`. Data sent from the On-Chip Debugger back to the host is identified by `DBG → Data`.

Read OCD Revision (00H). The read OCD revision command determines the version of the On-Chip Debugger. If OCD commands are added, removed or changed, this revision number changes.

```
DBG ← 00H
DBG → OCDRev[15:8] (Major revision number)
DBG → OCDRev[7:0] (Minor revision number)
```

Read OCD Status Register (02H). The read OCD Status Register command reads the OCDSTAT register.

```
DBG ← 02H
DBG → OCDSTAT[7:0]
```

Read Runtime Counter (03H). The runtime counter counts system clock cycles in between breakpoints. The 16-bit runtime counter counts from `0000H` and stops at the maximum count of `FFFFH`. The runtime counter is overwritten during the write memory, read memory, write register, read register, read memory CRC, step instruction, stuff instruction and execute instruction commands.

```
DBG ← 03H
DBG → RuntimeCounter[15:8]
DBG → RuntimeCounter[7:0]
```

Write OCD Control Register (04H). The write OCD Control Register command writes the data that follows to the OCDCTL register. When the Flash read protect option bit is enabled, the DBGMODE bit (`OCDCTL`[7]) is set to 1 only, it cannot be cleared to 0. To return the device to normal operating mode, the device must be reset.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

Read OCD Control Register (05H). The read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

Write Program Counter (06H). The write program counter command, writes the data that follows to the eZ8 CPU's program counter (PC). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, the program counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

Read Program Counter (07H). The read program counter command, reads the value in the eZ8 CPUs program counter (PC). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, this command returns `FFFFH`.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

Write Register (08H). The write register command, writes data to the Register File. Data is written 1–256 bytes at a time (256 bytes are written by setting size to 0). If the device is not in DEBUG Mode, the address and data values are discarded. If the Flash read protect option bit is enabled, only writes to the Flash control registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0,Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```

Read Register (09H). The read register command, reads data from the Register File. Data is read 1–256 bytes at a time (256 bytes are read by setting size to 0). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, this command returns `FFH` for all of the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

Write Program Memory (0AH). The write program memory command, writes data to program memory. This command is equivalent to the LDC and LDCI instructions. Data is written 1–65536 bytes at a time (65536 bytes are written by setting size to 0). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Read Program Memory (0BH). The read program memory command, reads data from program memory. This command is equivalent to the LDC and LDCI instructions. Data is read 1–65536 bytes at a time (65536 bytes are read by setting size to 0). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, this command returns `FFH` for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

Write Data Memory (0CH). The write data memory command, writes data to data memory. This command is equivalent to the LDE and LDEI instructions. Data is written 1–65536 bytes at a time (65536 bytes are written by setting size to 0). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Data Memory (0DH)**—The read data memory command, reads from data memory. This command is equivalent to the LDE and LDEI instructions. Data is read from 1 to 65536 bytes at a time (65536 bytes are read by setting size to 0). If the device is not in DEBUG Mode, this command returns `FFH` for the data.

```
DBG ← 0DH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

Read Program Memory CRC (0EH). The read program memory CRC command, computes and returns the cyclic redundancy check (CRC) of program memory using the 16-bit CRC-CCITT polynomial. If the device is not in DEBUG Mode, this command returns `FFFFH` for the CRC value. Unlike the other OCD read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads program memory, calculates the CRC value and returns the result. The delay is a function of program memory size and is approximately equal to the system clock period multiplied by the number of bytes in program memory.

```
DBG ← 0EH
DBG → CRC[15:8]
DBG → CRC[7:0]
```

Step Instruction (10H). The step instruction command, steps one assembly instruction at the current program counter (PC) location. If the device is not in DEBUG Mode or the Flash read protect option bit is enabled, the OCD ignores this command.

```
DBG ← 10H
```

Stuff Instruction (11H). The stuff instruction command, steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0-4 bytes of the instruction are read from program memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a breakpoint. If the device is not in DEBUG Mode or the Flash read protect option bit is enabled, the OCD ignores this command.

```
DBG ← 11H
DBG ← opcode[7:0]
```

Execute Instruction (12H). The execute instruction command allows sending an entire instruction to be executed to the eZ8 CPU. This command also steps over breakpoints. The number of bytes to send for the instruction depends on the Opcode. If the device is not in DEBUG Mode or the Flash read protect option bit is enabled, this command reads and discards one byte.

```
DBG ← 12H
DBG ← 1-5 byte opcode
```

On-Chip Debugger Control Register Definitions

This section describes the features of the On-Chip Debugger Control and Status registers.

OCD Control Register

The OCD Control Register controls the state of the On-Chip Debugger. This register is used to enter or exit DEBUG Mode and to enable the `BRK` instruction. It also resets the Z8 Encore! F083A Series device.

A reset and stop function is achieved by writing `81H` to this register. A reset and `Go` function is achieved by writing `41H` to this register. If the device is in DEBUG Mode, a run function is implemented by writing `40H` to this register.

Table 97. OCD Control Register (OCDCTL)

Bit	7	6	5	4	3	2	1	0
Field	DBGMODE	BRKEN	DBGACK	Reserved				RST
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W

Bit	Description
[7] DBGMODE	**DEBUG Mode** The device enters DEBUG Mode when this bit is 1. When in DEBUG Mode, the eZ8 CPU stops fetching new instructions. Clearing this bit causes the eZ8 CPU to restart. This bit is automatically set when a BRK instruction is decoded and breakpoints are enabled. If the Flash read protect option bit is enabled, this bit is cleared only by resetting the device. It cannot be written to 0. 0 = The Z8 Encore! F083A Series device is operating in NORMAL Mode. 1 = The Z8 Encore! F083A Series device is in DEBUG Mode.
[6] BRKEN	**Breakpoint Enable** This bit controls the behavior of the BRK instruction (Opcode 00H). By default, breakpoints are disabled and the BRK instruction behaves similar to an NOP instruction. If this bit is 1 when a BRK instruction is decoded, the DBGMODE bit of the OCDCTL register is automatically set to 1. 0 = Breakpoints are disabled. 1 = Breakpoints are enabled.
[5] DBGACK	**Debug Acknowledge** This bit enables the debug acknowledge feature. If this bit is set to 1, the OCD sends a Debug acknowledge character (FFH) to the host when a breakpoint occurs. 0 = Debug acknowledge is disabled. 1 = Debug acknowledge is enabled.
[4:1]	**Reserved** These bits are reserved and must be programmed to 0000.
[0] RST	**Reset** Setting this bit to 1 resets the Z8F04xA family device. The device goes through a normal POR sequence with the exception that the On-Chip Debugger is not reset. This bit is automatically cleared to 0 at the end of reset. 0 = No effect. 1 = Reset the Flash read protect option bit device.

OCD Status Register

The OCD Status Register reports status information about the current state of the debugger and the system.

Table 98. OCD Status Register (OCDSTAT)

Bit	7	6	5	4	3	2	1	0
Field	DBG	HALT	FRPENB	Reserved				
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Description
[7] DBG	**Debug Status** 0 = NORMAL Mode. 1 = DEBUG Mode.
[6] HALT	**HALT Mode** 0 = Not in HALT Mode. 1 = In HALT Mode.
[5] FRPENB	**Flash Read Protect Option Bit Enable** 0 = FRP bit enabled, that allows disabling of many OCD commands. 1 = FRP bit has no effect.
[4:0]	**Reserved** These bits are reserved and must be programmed to 00000.

Oscillator Control

The Z8 Encore! F083A Series device uses five possible clocking schemes; each one of these clocking schemes is user-selectable.

- On-chip precision trimmed RC oscillator
- On-chip oscillator using off-chip crystal or resonator
- On-chip oscillator using external RC network
- External clock drive
- On-chip low precision Watchdog Timer Oscillator

In addition, Z8 Encore! F083A Series devices contain clock failure detection and recovery circuitry to allow continued operation despite any potential failure of the primary oscillator.

Operation

This chapter discusses the logic used to select the system clock and handle primary oscillator failures. A description of the specific operation of each oscillator is outlined elsewhere in this document: see the Watchdog Timer chapter on page 92 and the Crystal Oscillator chapter on page 157.

System Clock Selection

The oscillator control block selects from the available clocks. Table 99 describes each clock source and its usage.

Table 99. Oscillator Configuration and Selection

Clock Source	Characteristics	Required Setup
Internal precision RC oscillator	• 119 kHz or 20 MHz • ± 4% accuracy when trimmed • No external components required	• Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator at either 20 MHz or 119 kHz
External crystal/ resonator	• 32 kHz to 20 MHz • Very high accuracy (dependent on crystal or resonator used) • Requires external components	• Configure Flash option bits for correct external OSCILLATOR mode • Unlock and write OSCCTL to enable crystal oscillator, wait for it to stabilize and select as system clock (if the XTLDIS option bit has been deasserted, no waiting is required)
External RC oscillator	• 32 kHz to 4 MHz • Accuracy dependent on external components	• Configure Flash option bits for correct external OSCILLATOR Mode • Unlock and write OSCCTL to enable crystal oscillator and select as system clock
External clock drive	• 0 to 20 MHz • Accuracy dependent on external clock source	• Write GPIO registers to configure PB3 pin for external clock function • Unlock and write OSCCTL to select external system clock • Apply external clock signal to GPIO
Internal Watchdog Timer Oscillator	• 10 kHz nominal • ± 40% accuracy; no external components required • Low power consumption	• Enable WDT if not enabled and wait until WDT oscillator is operating. • Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator

⚠ **Caution:** Unintentional accesses to the Oscillator Control Register actually stop the chip by switching to a nonfunctioning oscillator. To prevent this condition, the oscillator control block employs a register-unlocking/locking scheme.

OSC Control Register Unlocking/Locking

To write the Oscillator Control Register, unlock it by making two writes to the OSCCTL Register with the values `E7H` followed by `18H`. A third write to the OSCCTL Register changes the value of the actual register and returns the register to a Locked state. Any other sequence of Oscillator Control Register writes has no effect. The values written to unlock the register must be ordered correctly, but are not necessarily consecutive. It is possible to write to or read from other registers within the unlocking/locking operation.

When selecting a new clock source, the primary oscillator failure detection circuitry and the Watchdog Timer Oscillator failure circuitry must be disabled. If POFEN and WOFEN are not disabled prior to a clock switch-over, it is possible to generate an interrupt for a failure of either oscillator. The failure detection circuitry is enabled anytime after a successful write of OSCSEL in the Oscillator Control Register.

The internal precision oscillator is enabled by default. If the user code changes to a different oscillator, it may be appropriate to disable the IPO for power savings. Disabling the IPO does not occur automatically.

Clock Failure Detection and Recovery

Should an oscillator or timer fail, there are methods of recovery, as this section describes.

Primary Oscillator Failure

The Z8F04xA family devices generates nonmaskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer Oscillator to drive the system clock. The Watchdog Timer Oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer is the primary oscillator. It is also unavailable if the Watchdog Timer Oscillator is disabled, though it is not necessary to enable the Watchdog timer reset function outlined in the Watchdog Timer chapter of this document.

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below 1kHz ±50%. If an external signal is selected as the system oscillator, it is possible that a very slow but nonfailing clock generates a failure condition. Under these conditions, do not enable the clock failure circuitry (POFEN must be deasserted in the OSCCTL Register).

Watchdog Timer Failure

In the event of a Watchdog Timer Oscillator failure, a similar nonmaskable interrupt-like event is issued. This event does not trigger an attendant clock switch-over, but alerts the CPU of the failure. After a Watchdog Timer failure, it is no longer possible to detect a primary oscillator failure. The failure detection circuitry does not function if the Watchdog Timer is used as the primary oscillator or if the Watchdog Timer Oscillator has been disabled. For either of these cases, it is necessary to disable the detection circuitry by deasserting the WDFEN bit of the OSCCTL Register.

The Watchdog Timer Oscillator failure detection circuit counts system clocks while looking for a Watchdog Timer clock. The logic counts 8004 system clock cycles before determining that a failure has occurred. The system clock rate determines the speed at which the Watchdog Timer failure is detected. A very slow system clock results in very slow detection times.

⚠ **Caution:** It is possible to disable the clock failure detection circuitry as well as all functioning clock sources. In this case, the Z8 Encore! F083A Series device ceases functioning and is recovered only by power-on-reset.

Oscillator Control Register Definitions

The following section provides the bit definitions for the Oscillator Control Register.

Oscillator Control Register

The Oscillator Control Register (OSCCTL), shown in Table 100, enables/disables the various oscillator circuits, enables/disables the failure detection/recovery circuitry and selects the primary oscillator, which becomes the system clock.

The Oscillator Control Register must be unlocked before writing. Writing the two step sequence `E7H` followed by `18H` to the Oscillator Control Register unlocks it. The register is locked at successful completion of a register write to the OSCCTL. Figure 21 on page 156 displays the oscillator control clock switching flow. To determine the waiting times of various oscillator circuits, see Table 118 on page 188.

Table 100. Oscillator Control Register (OSCCTL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	INTEN	XTLEN	WDTEN	POFEN	WDFEN	SCKSEL		
RESET	1	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F86H							

Bit	Description
[7] INTEN	**Internal Precision Oscillator Enable** 1 = Internal precision oscillator is enabled. 0 = Internal precision oscillator is disabled.
[6] XTLEN	**Crystal Oscillator Enable** This setting overrides the GPIO register control for PA0 and PA1. 1 = Crystal oscillator is enabled. 0 = Crystal oscillator is disabled.

Bit	Description (Continued)
[5] WDTEN	**Watchdog Timer Oscillator Enable** 1 = Watchdog Timer Oscillator is enabled. 0 = Watchdog Timer Oscillator is disabled.
[4] POFEN	**Primary Oscillator Failure Detection Enable** 1 = Failure detection and recovery of primary oscillator is enabled. 0 = Failure detection and recovery of primary oscillator is disabled.
[3] WDFEN	**Watchdog Timer Oscillator Failure Detection Enable** 1 = Failure detection of Watchdog Timer Oscillator is enabled. 0 = Failure detection of Watchdog Timer Oscillator is disabled.
[2:0] SCKSEL	**System Clock Oscillator Select** 000 = Internal precision oscillator functions as system clock at 20MHz. 001 = Internal precision oscillator functions as system clock at 119kHz. 010 = Crystal oscillator or external RC oscillator functions as system clock. 011 = Watchdog Timer Oscillator functions as system clock. 100 = External clock signal on PB3 functions as system clock. 101 = Reserved. 110 = Reserved. 111 = Reserved.

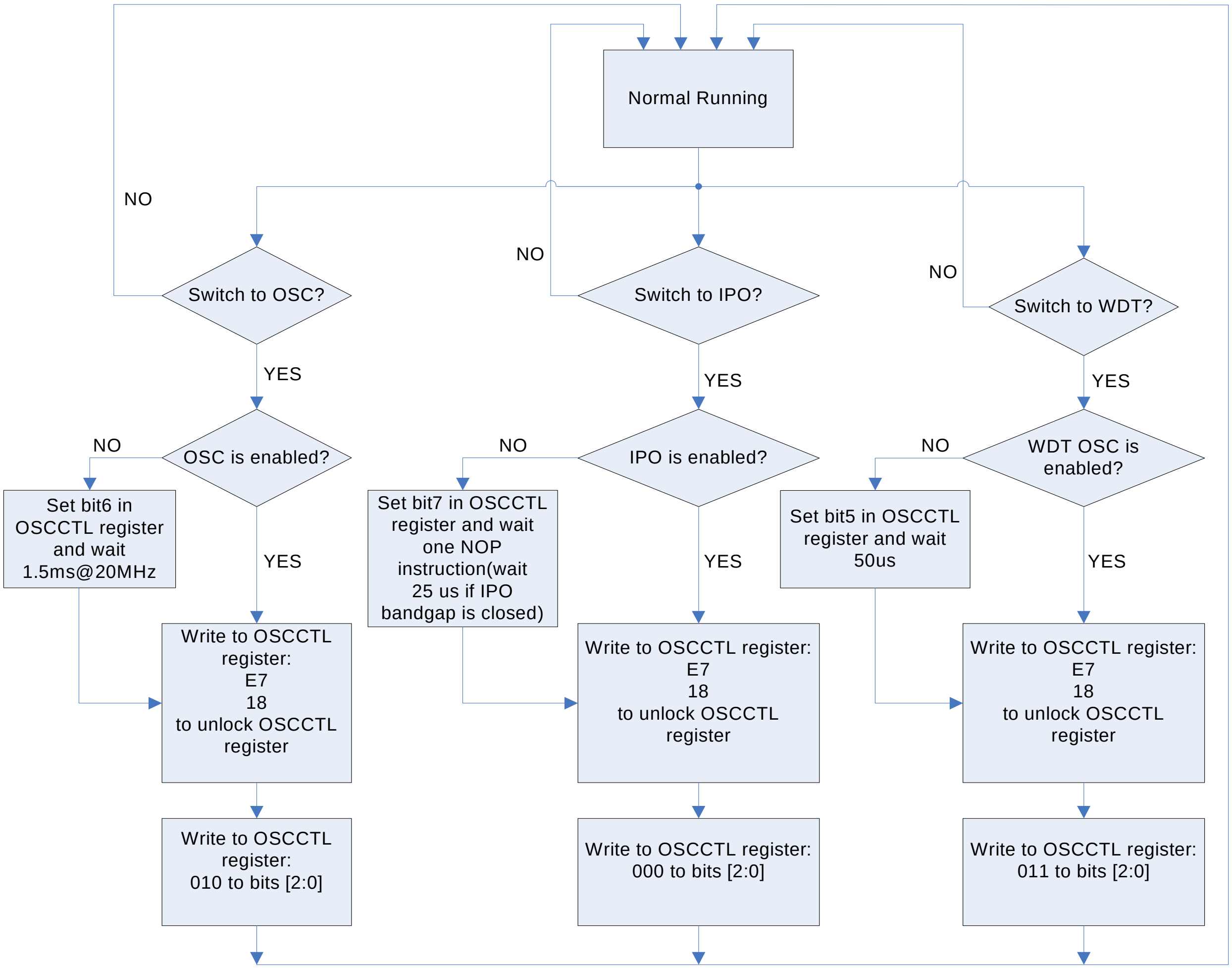

Normal Running
NO
Switch to OSC?
NO
Switch to IPO?
NO
Switch to WDT?
YES
YES
YES
NO
OSC is enabled?
NO
IPO is enabled?
NO
WDT OSC is enabled?
Set bit6 in OSCCTL register and wait 1.5ms@20MHz
YES
Set bit7 in OSCCTL register and wait one NOP instruction(wait 25 us if IPO bandgap is closed)
YES
Set bit5 in OSCCTL register and wait 50us
YES
Write to OSCCTL register: E7 18 to unlock OSCCTL register
Write to OSCCTL register: E7 18 to unlock OSCCTL register
Write to OSCCTL register: E7 18 to unlock OSCCTL register
Write to OSCCTL register: 010 to bits [2:0]
Write to OSCCTL register: 000 to bits [2:0]
Write to OSCCTL register: 011 to bits [2:0]

Figure 21. Oscillator Control Clock Switching Flow Chart

Crystal Oscillator

The products in the Z8 Encore! F083A Series contain an on-chip crystal oscillator for use with external crystals with 32kHz to 20MHz frequencies. In addition, the oscillator supports external RC networks with oscillation frequencies up to 4MHz or ceramic resonators with frequencies up to 8MHz. The on-chip crystal oscillator is used to generate the primary system clock for the internal eZ8 CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin also accepts a CMOS-level clock input signal (32kHz–20MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected. The on-chip crystal oscillator also contains a clock filter function. For details settings of this clock filter, see Table 91 on page 133. But by default, this clock filter is disabled and no divide to the input clock, namely the frequency of the signal on the X_{IN} input pin determines the frequency of the system clock in default settings.

Note: Although the XIN pin is used as an input for an external clock generator, the CLKIN pin is better suited for such use. For more details, see the System Clock Selection section on page 151.

Operating Modes

The Z8 Encore! F083A Series products support four OSCILLATOR modes.

- Minimum power for use with very low frequency crystals (32kHz to 1MHz)
- Medium power for use with medium frequency crystals or ceramic resonators (0.5MHz to 8MHz).
- Maximum power for use with high frequency crystals (8MHz to 20MHz)
- On-chip oscillator configured for use with external RC networks (<4MHz)

The OSCILLATOR mode is selected using user-programmable Flash option bits. For more details, see *the* Flash Option Bits chapter on page 124.

Crystal Oscillator Operation

The Flash option bit XTLDIS controls whether the crystal oscillator is enabled during reset. The crystal may later be disabled after reset if a new oscillator has been selected as the system clock. If the crystal is manually enabled after reset through the OSCCTL Register, the user code must wait at least 5000 IPO cycles for the crystal to stabilize. After this, the crystal oscillator may be selected as the system clock.

Figure 22 displays a recommended configuration for connection with an external fundamental-mode, parallel-resonant crystal operating at 20MHz. Recommended 20MHz crystal specifications are provided in Table 101. Resistor R_1 is optional and limits total power dissipation by the crystal. Printed circuit board layout must add no more than 4pF of stray capacitance to either the X_{IN} or X_{OUT} pins. If oscillation does not occur, reduce the values of capacitors C_1 and C_2 to decrease loading.

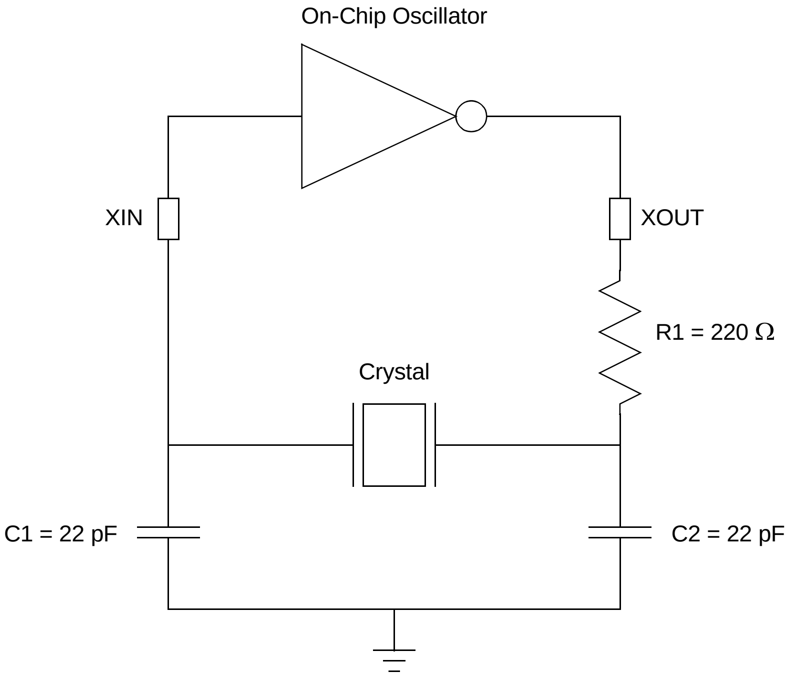


Figure 22. Recommended 20MHz Crystal Oscillator Configuration

Table 101. Recommended Crystal Oscillator Specifications

Parameter	Value	Units	Comments
Frequency	20	MHz	
Resonance	Parallel		
Mode	Fundamental		
Series Resistance (R_S)	60	Ω	Maximum
Load Capacitance (C_L)	30	pF	Maximum
Shunt Capacitance (C_0)	7	pF	Maximum
Drive Level	1	mW	Maximum

Oscillator Operation with an External RC Network

Figure 23 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

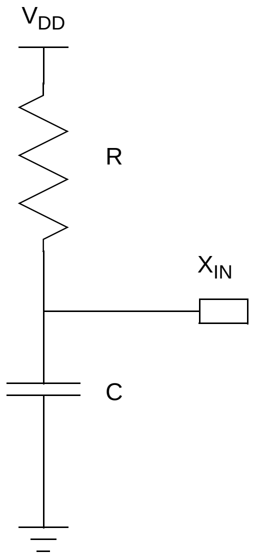


Figure 23. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45 KΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40 KΩ. The typical oscillator frequency is estimated from the values of the resistor elements (R, measured in KΩ) and capacitor elements (C, measured in pF) using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1 \times 10^6}{(0.4 \times R \times C) + (4 \times C)}$$

Figure 24 displays the typical (3.3 V and 25°C) oscillator frequency as a function of the capacitor (C in pF) employed in the RC network assuming a 45 KΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator X_{IN} pin and the printed circuit board must be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20 pF are recommended.

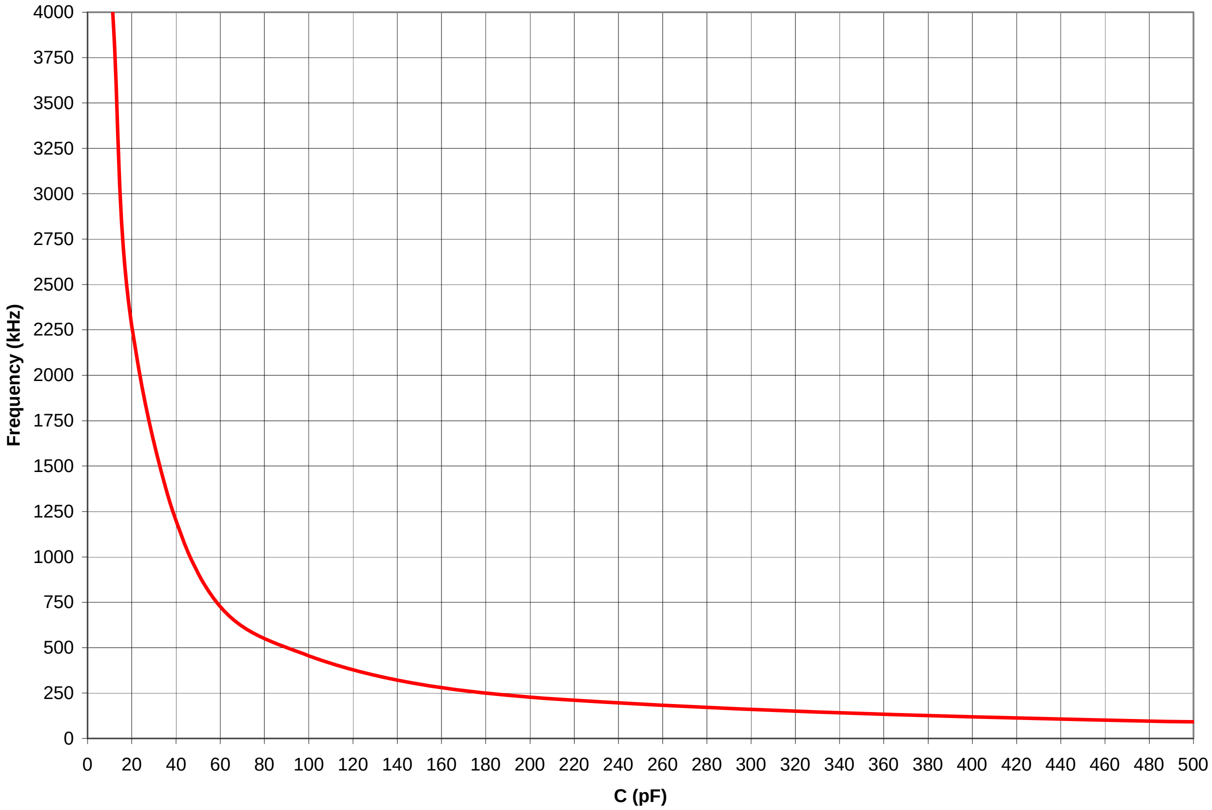


Figure 24. Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45KΩ Resistor

Caution: When operating in external RC OSCILLATOR Mode, the oscillator stops oscillating if the power supply drops below 2.7V, but before the power supply drops to the VBO threshold. The oscillator resumes oscillation when the supply voltage exceeds 2.7V.

Internal Precision Oscillator

The internal precision oscillator (IPO) is designed for use without external components. You can either manually trim the oscillator for a nonstandard frequency or use the automatic factory trimmed version to achieve a 20MHz frequency with ±4% accuracy and 45%–55% duty cycle over the operating temperature and supply voltage of the device. Maximum startup time of IPO is 25µs. IPO features include:

- On-chip RC oscillator that does not require external components
- Output frequency of either 20MHz or 119kHz (contains both a FAST and a SLOW mode)
- Trimming possible through Flash option bits with user override
- Elimination of crystals or ceramic resonators in applications where high timing accuracy is not required

Operation

The internal oscillator is an RC relaxation oscillator with minimized sensitivity to power supply variation. By using ratio tracking thresholds, the effect of power supply voltage is cancelled out. The dominant source of oscillator error is the absolute variance of chip level fabricated components, such as capacitors. An 8-bit trimming register, incorporated into the design, compensates for absolute variation of oscillator frequency. Once trimmed the oscillator frequency is stable and does not require subsequent calibration. Trimming was performed during manufacturing and is not necessary for the user to repeat unless a frequency other than 20MHz (FAST Mode) or 119kHz (SLOW Mode) is required.

Power-down this block for minimum system power.

By default, the oscillator is configured through the Flash option bits. However, the user code overrides these trim values as described in *the* Trim Bit Address Space *section on page 129*.

Select one of two frequencies for the oscillator: 20MHz and 119kHz, using the OSCSEL bits in *the* Oscillator Control *chapter on page 151*.

eZ8 CPU Instruction Set

This chapter describes the following features of the eZ8 CPU instruction set:

Assembly Language Programming Introduction

The eZ8 CPU assembly language provides a means for writing an application program without concern for actual memory addresses or machine instruction formats. A program written in assembly language is called a source program. Assembly language allows the use of symbolic addresses to identify memory locations. It also allows mnemonic codes (op codes and operands) to represent the instructions themselves. The op codes identify the instruction while the operands represent memory locations, registers or immediate data values.

Each assembly language program consists of a series of symbolic commands called statements. Each statement contain labels, operations, operands and comments.

Labels are assigned to a particular instruction step in a source program. The label identifies that step in the program as an entry point for use by other instructions.

The assembly language also includes assembler directives that supplement the machine instruction. The assembler directives or pseudo-ops, are not translated into a machine instruction. Rather, the pseudo-ops are interpreted as directives that control or assist the assembly process.

The source program is processed (assembled) by the assembler to obtain a machine language program called the object code. The object code is executed by the eZ8 CPU. An example segment of an assembly language program is given in the following example.

Assembly Language Source Program Example

```
JP START          ; Everything after the semicolon is a comment.
START:             A label called “START”. The first instruction (JP START) in this
                  ; example causes program execution to jump to the point within the
                  ; program where the START label occurs.
LD R4, R7         ; A Load (LD) instruction with two operands. The first operand,
                  ; Working register R4, is the destination. The second operand,
                  ; Working register R7, is the source. The contents of R7 is
                  ; written into R4.
LD 234H, #%01     ; Another Load (LD) instruction with two operands.
                  ; The first operand, extended mode register Address 234H,
                  ; identifies the destination. The second operand, immediate data
                  ; value 01H, is the source. The 01H value is written into the
                  ; register at address 234H.
```

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as ‘destination, source’. After assembly, the object code usually has the operands in the order ‘source, destination’, but ordering is op code-dependent. The following instruction examples display the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed by users that prefer manual program coding or intend to implement their own assembler.

Example 1

If the contents of registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 102. Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H,	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2

In general, when an instruction format requires an 8-bit register address, the address specify any register location in the range 0–255 or, using escaped mode addressing, a working register R0–R15. If the contents of register 43H and working register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 103. Assembly Language Syntax Example 2

Assembly Language Code	`ADD`	`43H,`	`R8`	`(ADD dst, src)`
Object Code	`04`	`E8`	`43`	`(OPC src, dst)`

The Register File size varies, depending on the device type. See the device-specific Z8 Encore! product specification to determine the exact Register File range available.

eZ8 CPU Instruction Notation

In the eZ8 CPU instruction summary and description sections, the operands, condition codes, status flags and address modes are represented by a notational shorthand that is described in Table 104.

Table 104. Notational Shorthand

Notation	Description	Operand	Range
b	Bit	b	b represents a value from 0 to 7 (000B to 111B).
cc	Condition Code	—	See the condition codes overview in the eZ8 CPU Core User Manual (UM0128).
DA	Direct Address	Addrs	*Addrs* represents a number in the range of 0000H to FFFFH.
ER	Extended addressing Register	Reg	*Reg* represents a number in the range of 000H to FFFH.
IM	Immediate Data	#Data	Data is a number between 00H to FFH/
Ir	Indirect Working Register	@Rn	n = 0–15.
IR	Indirect Register	@Reg	*Reg* represents a number in the range of 00H to FFH/
Irr	Indirect Working Register Pair	@RRp	p = 0, 2, 4, 6, 8, 10, 12 or 14.
IRR	Indirect Register Pair	@Reg	*Reg* represents an even number in the range 00H to FEH.
p	Polarity	p	Polarity is a single bit binary value of either 0B or 1B.
r	Working Register	Rn	n = 0–15.
R	Register	Reg	Reg. represents a number in the range of 00H to FFH.

Table 104. Notational Shorthand (Continued)

Notation	Description	Operand	Range
RA	Relative Address	X	X represents an index in the range of +127 to –128 which is an offset relative to the address of the next instruction
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12 or 14
RR	Register Pair	Reg	Reg. represents an even number in the range of 00H to FEH
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to –128 range.

Table 105 contains additional symbols that are used throughout the instruction summary and instruction set description sections.

Table 105. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow, as shown in the following example.

dst ← dst + src

This example indicates that the source data is added to the destination data; the result is stored in the destination location.

eZ8 CPU Instruction Classes

eZ8 CPU instructions are divided functionally into the following groups:

- Arithmetic
- Bit manipulation
- Block transfer
- CPU control
- Load
- Logical
- Program control
- Rotate and shift

Tables 106 through 113 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instruction are considered as a subset of more than one category. Within these tables, the source operand is identified as ‘src’, the destination operand is ‘dst’ and a condition code is ‘cc’.

Table 106. Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst, src	Add with Carry
ADCX	dst, src	Add with Carry using extended addressing
ADD	dst, src	Add
ADDX	dst, src	Add using extended addressing
CP	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using extended addressing
CPX	dst, src	Compare using extended addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word

Table 106. Arithmetic Instructions (Continued)

Mnemonic	Operands	Instruction
MULT	dst	Multiply
SBC	dst, src	Subtract with Carry
SBCX	dst, src	Subtract with Carry using extended addressing
SUB	dst, src	Subtract
SUBX	dst, src	Subtract using extended addressing

Table 107. Bit Manipulation Instructions

Mnemonic	Operands	Instruction
BCLR	bit, dst	Bit Clear
BIT	p, bit, dst	Bit Set or Clear
BSET	bit, dst	Bit Set
BSWAP	dst	Bit Swap
CCF	—	Complement Carry Flag
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
TCM	dst, src	Test Complement Under Mask
TCMX	dst, src	Test Complement Under Mask using extended addressing
TM	dst, src	Test Under Mask
TMX	dst, src	Test Under Mask using extended addressing

Table 108. Block Transfer Instructions

Mnemonic	Operands	Instruction
LDCI	dst, src	Load constant to/from Program Memory and autoincrement addresses.
LDEI	dst, src	Load external data to/from Data Memory and autoincrement addresses.

Table 109. CPU Control Instructions

Mnemonic	Operands	Instruction
ATM	—	Atomic Execution
CCF	—	Complement Carry Flag
DI	—	Disable Interrupts
EI	—	Enable Interrupts
HALT	—	HALT Mode
NOP	—	No Operation
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
SRP	src	Set Register Pointer
STOP	—	STOP Mode
WDT	—	Watchdog Timer Refresh

Table 110. Load Instructions

Mnemonic	Operands	Instruction
CLR	dst	Clear
LD	dst, src	Load
LDC	dst, src	Load constant to/from Program Memory
LDCI	dst, src	Load constant to/from Program Memory and autoincrement addresses
LDE	dst, src	Load external data to/from Data Memory
LDEI	dst, src	Load external data to/from Data Memory and autoincrement addresses
LDWX	dst, src	Load Word using extended addressing
LDX	dst, src	Load using extended addressing
LEA	dst, X(src)	Load effective address
POP	dst	Pop
POPX	dst	Pop using extended addressing
PUSH	src	Push
PUSHX	src	Push using extended addressing

Table 111. Logical Instructions

Mnemonic	Operands	Instruction
AND	dst, src	Logical AND
ANDX	dst, src	Logical AND using extended addressing
COM	dst	Complement
OR	dst, src	Logical OR
ORX	dst, src	Logical OR using extended addressing
XOR	dst, src	Logical Exclusive OR
XORX	dst, src	Logical Exclusive OR using extended addressing

Table 112. Program Control Instructions

Mnemonic	Operands	Instruction
BRK	—	On-Chip Debugger Break
BTJ	p, bit, src, DA	Bit Test and Jump
BTJNZ	bit, src, DA	Bit Test and Jump if Non-Zero
BTJZ	bit, src, DA	Bit Test and Jump if Zero
CALL	dst	Call Procedure
DJNZ	dst, src, RA	Decrement and Jump Non-Zero
IRET	—	Interrupt Return
JP	dst	Jump
JP cc	dst	Jump Conditional
JR	DA	Jump Relative
JR cc	DA	Jump Relative Conditional
RET	—	Return
TRAP	vector	Software Trap

Table 113. Rotate and Shift Instructions

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right through Carry
SRA	dst	Shift Right Arithmetic
SRL	dst	Shift Right Logical
SWAP	dst	Swap Nibbles

eZ8 CPU Instruction Summary

Table 114 summarizes the eZ8 CPU instructions. The table identifies the addressing modes employed by the instruction, the effect upon the Flags register, the number of CPU clock cycles required for the instruction fetch and the number of CPU clock cycles required for the instruction execution.

Table 114. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	2	3
		r	Ir	13							2	4
		R	R	14							3	3
		R	IR	15							3	4
		R	IM	16							3	3
		IR	IM	17							3	4
ADCX dst, src	dst ← dst + src + C	ER	ER	18	*	*	*	*	0	*	4	3
		ER	IM	19							4	3
ADD dst, src	dst ← dst + src	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	03							2	4
		R	R	04							3	3
		R	IR	05							3	4
		R	IM	06							3	3
		IR	IM	07							3	4
ADDX dst, src	dst ← dst + src	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							4	3

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

		Address Mode			Flags							
Assembly Mnemonic	Symbolic Operation	dst	src	Op Code(s) (Hex)	C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	dst ← dst AND src	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	dst[bit] ← 0	r		E2	–	*	*	0	–	–	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	–	*	*	0	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	–	*	*	0	–	–	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

		Address Mode		Op Code(s)	Flags							
Assembly Mnemonic	**Symbolic Operation**	**dst**	**src**	**(Hex)**	**C**	**Z**	**S**	**V**	**D**	**H**	**Fetch Cycles**	**Instr. Cycles**
CALL dst	SP ← SP -2 @SP ← PC PC ← dst	IRR		D4	–	–	–	–	–	–	2	6
		DA		D6							3	3
CCF	C ← ~C			EF	*	–	–	–	–	–-	1	2
CLR dst	dst ← 00H	R		B0	–	–	–	–	–	–	2	2
		IR		B1							2	3
COM dst	dst ← ~dst	R		60	–	*	*	0	–	–	2	2
		IR		61							2	3
CP dst, src	dst - src	r	r	A2	*	*	*	*	–	–	2	3
		r	Ir	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst - src - C	r	r	1F A2	*	*	*	*	–	–	3	3
		r	Ir	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst - src - C	ER	ER	1F A8	*	*	*	*	–	–	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst - src	ER	ER	A8	*	*	*	*	–	–	4	3
		ER	IM	A9							4	3

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
DA dst	dst ← DA(dst)	R		40	*	*	*	X	–	–	2	2
		IR		41							2	3
DEC dst	dst ← dst - 1	R		30	–	*	*	*	–	–	2	2
		IR		31							2	3
DECW dst	dst ← dst - 1	RR		80	–	*	*	*	–	–	2	5
		IRR		81							2	6
DI	IRQCTL[7] ← 0			8F	–	–	–	–	–	–	1	2
DJNZ dst, RA	dst ← dst – 1 if dst ≠ 0 PC ← PC + X	r		0A-FA	–	–	–	–	–	–	2	3
EI	IRQCTL[7] ← 1			9F	–	–	–	–	–	–	1	2
HALT	HALT Mode			7F	–	–	–	–	–	–	1	2
INC dst	dst ← dst + 1	R		20	–	*	*	–	–	–	2	2
		IR		21							2	3
		r		0E-FE							1	2
INCW dst	dst ← dst + 1	RR		A0	–	*	*	*	–	–	2	5
		IRR		A1							2	6
IRET	FLAGS ← @SP SP ← SP + 1 PC ← @SP SP ← SP + 2 IRQCTL[7] ← 1			BF	*	*	*	*	*	*	1	5
JP dst	PC ← dst	DA		8D	–	–	–	–	–	–	3	2
		IRR		C4							2	3
JP cc, dst	if cc is true PC ← dst	DA		0D-FD	–	–	–	–	–	–	3	2

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
JR dst	PC ← PC + X	DA		8B	–	–	–	–	–	–	2	2
JR cc, dst	if cc is true PC ← PC + X	DA		0B-FB	–	–	–	–	–	–	2	2
LD dst, rc	dst ← src	r	IM	0C-FC	–	–	–	–	–	–	2	2
		r	X(r)	C7							3	3
		X(r)	r	D7							3	4
		r	Ir	E3							2	3
		R	R	E4							3	2
		R	IR	E5							3	4
		R	IM	E6							3	2
		IR	IM	E7							3	3
		Ir	r	F3							2	3
		IR	R	F5							3	3
LDC dst, src	dst ← src	r	Irr	C2	–	–	–	–	–	–	2	5
		Ir	Irr	C5							2	9
		Irr	r	D2							2	5
LDCI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	C3	–	–	–	–	–	–	2	9
		Irr	Ir	D3							2	9
LDE dst, src	dst ← src	r	Irr	82	–	–	–	–	–	–	2	5
		Irr	r	92							2	5
LDEI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	83	–	–	–	–	–	–	2	9
		Irr	Ir	93							2	9
LDWX dst, src	dst ← src	ER	ER	1FE8	–	–	–	–	–	–	5	4

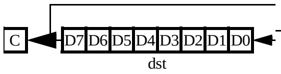
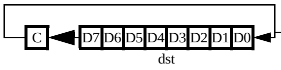
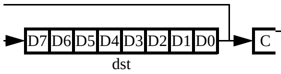
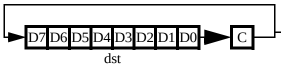
Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Flags Z	Flags S	Flags V	Flags D	Flags H	Fetch Cycles	Instr. Cycles
LDX dst, src	dst ← src	r	ER	84	–	–	–	–	–	–	3	2
		Ir	ER	85							3	3
		R	IRR	86							3	4
		IR	IRR	87							3	5
		r	X(rr)	88							3	4
		X(rr)	r	89							3	4
		ER	r	94							3	2
		ER	Ir	95							3	3
		IRR	R	96							3	4
		IRR	IR	97							3	5
		ER	ER	E8							4	2
		ER	IM	E9							4	2
LEA dst, X(src)	dst ← src + X	r	X(r)	98	–	–	–	–	–	–	3	3
		rr	X(rr)	99							3	5
MULT dst	dst[15:0] ← dst[15:8] * dst[7:0]	RR		F4	–	–	–	–	–	–	2	8
NOP	No operation			0F	–	–	–	–	–	–	1	2
OR dst, src	dst ← dst OR src	r	r	42	–	*	*	0	–	–	2	3
		r	Ir	43							2	4
		R	R	44							3	3
		R	IR	45							3	4
		R	IM	46							3	3
		IR	IM	47							3	4

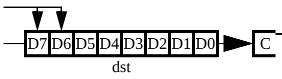
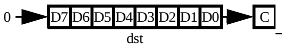
Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
ORX dst, src	dst ← dst OR src	ER	ER	48	–	*	*	0	–	–	4	3
		ER	IM	49							4	3
POP dst	dst ← @SP SP ← SP + 1	R		50	–	–	–	–	–	–	2	2
		IR		51							2	3
POPX dst	dst ← @SP SP ← SP + 1	ER		D8	–	–	–	–	–	–	3	2
PUSH src	SP ← SP – 1 @SP ← src	R		70	–	–	–	–	–	–	2	2
		IR		71							2	3
		IM		IF70							3	2
PUSHX src	SP ← SP – 1 @SP ← src	ER		C8	–	–	–	–	–	–	3	2
RCF	C ← 0			CF	0	–	–	–	–	–	1	2
RET	PC ← @SP SP ← SP + 2			AF	–	–	–	–	–	–	1	4
RL dst		R		90	*	*	*	*	–	–	2	2
		IR		91							2	3
RLC dst		R		10	*	*	*	*	–	–	2	2
		IR		11							2	3
RR dst		R		E0	*	*	*	*	–	–	2	2
		IR		E1							2	3
RRC dst		R		C0	*	*	*	*	–	–	2	2
		IR		C1							2	3

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
SBC dst, src	dst ← dst – src - C	r	r	32	*	*	*	*	1	*	2	3
		r	Ir	33							2	4
		R	R	34							3	3
		R	IR	35							3	4
		R	IM	36							3	3
		IR	IM	37							3	4
SBCX dst, src	dst ← dst – src - C	ER	ER	38	*	*	*	*	1	*	4	3
		ER	IM	39							4	3
SCF	C ← 1			DF	1	–	–	–	–	–	1	2
SRA dst		R		D0	*	*	*	0	–	–	2	2
		IR		D1							2	3
SRL dst		R		1F C0	*	*	0	*	–	–	3	2
		IR		1F C1							3	3
SRP src	RP ← src		IM	01	–	–	–	–	–	–	2	2
STOP	STOP Mode			6F	–	–	–	–	–	–	1	2
SUB dst, src	dst ← dst – src	r	r	22	*	*	*	*	1	*	2	3
		r	Ir	23							2	4
		R	R	24							3	3
		R	IR	25							3	4
		R	IM	26							3	3
		IR	IM	27							3	4
SUBX dst, src	dst ← dst – src	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29							4	3

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
SWAP dst	dst[7:4] ↔ dst[3:0]	R		F0	X	*	*	X	–	–	2	2
		IR		F1							2	3
TCM dst, src	(NOT dst) AND src	r	r	62	–	*	*	0	–	–	2	3
		r	Ir	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	–	*	*	0	–	–	4	3
		ER	IM	69							4	3
TM dst, src	dst AND src	r	r	72	–	*	*	0	–	–	2	3
		r	Ir	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	–	*	*	0	–	–	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	–	–	–	–	–	–	2	6
WDT				5F	–	–	–	–	–	–	1	2

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Flags Z	Flags S	Flags V	Flags D	Flags H	Fetch Cycles	Instr. Cycles
XOR dst, src	dst ← dst XOR src	r	r	B2	–	*	*	0	–	–	2	3
		r	Ir	B3							2	4
		R	R	B4							3	3
		R	IR	B5							3	4
		R	IM	B6							3	3
		IR	IM	B7							3	4
XORX dst, src	dst ← dst XOR src	ER	ER	B8	–	*	*	0	–	–	4	3
		ER	IM	B9							4	3

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Op Code Maps

A description of the opcode map data and the abbreviations are provided in Figure 25. Table 115 lists op code map abbreviations.

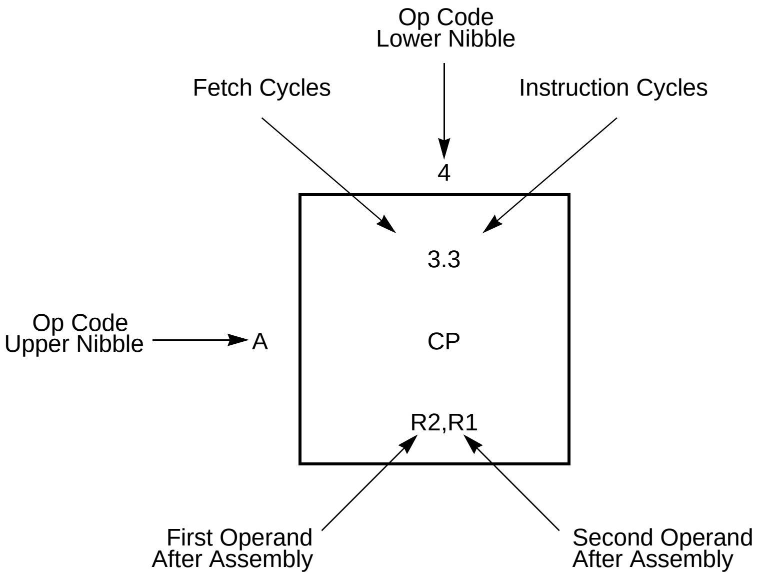


Figure 25. Op Code Map Cell Description

Table 115. Op Code Map Abbreviations

Abbreviation	Description	Abbreviation	Description
b	Bit position	IRR	Indirect Register Pair
cc	Condition code	p	Polarity (0 or 1)
X	8-bit signed index or displacement	r	4-bit Working Register
DA	Destination address	R	8-bit register
ER	Extended addressing register	r1, R1, Ir1, Irr1, IR1, rr1, RR1, IRR1, ER1	Destination address
IM	Immediate data value	r2, R2, Ir2, Irr2, IR2, rr2, RR2, IRR2, ER2	Source address
Ir	Indirect Working Register	RA	Relative
IR	Indirect register	rr	Working Register Pair
Irr	Indirect Working Register Pair	RR	Register Pair

Figures 26 and 27 provide operation code mapping information about each of the eZ8 CPU instructions.

Lower Nibble (Hex)

Upper Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1.1 **BRK**	2.2 **SRP** IM	2.3 **ADD** r1,r2	2.4 **ADD** r1,Ir2	3.3 **ADD** R2,R1	3.4 **ADD** IR2,R1	3.3 **ADD** R1,IM	3.4 **ADD** IR1,IM	4.3 **ADDX** ER2,ER1	4.3 **ADDX** IM,ER1	2.3 **DJNZ** r1,X	2.2 **JR** cc,X	2.2 **LD** r1,IM	3.2 **JP** cc,DA	1.2 **INC** r1	1.2 **NOP**
1	2.2 **RLC** R1	2.3 **RLC** IR1	2.3 **ADC** r1,r2	2.4 **ADC** r1,Ir2	3.3 **ADC** R2,R1	3.4 **ADC** IR2,R1	3.3 **ADC** R1,IM	3.4 **ADC** IR1,IM	4.3 **ADCX** ER2,ER1	4.3 **ADCX** IM,ER1	↓	↓	↓	↓	↓	See 2nd Op Code Map
2	2.2 **INC** R1	2.3 **INC** IR1	2.3 **SUB** r1,r2	2.4 **SUB** r1,Ir2	3.3 **SUB** R2,R1	3.4 **SUB** IR2,R1	3.3 **SUB** R1,IM	3.4 **SUB** IR1,IM	4.3 **SUBX** ER2,ER1	4.3 **SUBX** IM,ER1						
3	2.2 **DEC** R1	2.3 **DEC** IR1	2.3 **SBC** r1,r2	2.4 **SBC** r1,Ir2	3.3 **SBC** R2,R1	3.4 **SBC** IR2,R1	3.3 **SBC** R1,IM	3.4 **SBC** IR1,IM	4.3 **SBCX** ER2,ER1	4.3 **SBCX** IM,ER1						
4	2.2 **DA** R1	2.3 **DA** IR1	2.3 **OR** r1,r2	2.4 **OR** r1,Ir2	3.3 **OR** R2,R1	3.4 **OR** IR2,R1	3.3 **OR** R1,IM	3.4 **OR** IR1,IM	4.3 **ORX** ER2,ER1	4.3 **ORX** IM,ER1						
5	2.2 **POP** R1	2.3 **POP** IR1	2.3 **AND** r1,r2	2.4 **AND** r1,Ir2	3.3 **AND** R2,R1	3.4 **AND** IR2,R1	3.3 **AND** R1,IM	3.4 **AND** IR1,IM	4.3 **ANDX** ER2,ER1	4.3 **ANDX** IM,ER1						1.2 **WDT**
6	2.2 **COM** R1	2.3 **COM** IR1	2.3 **TCM** r1,r2	2.4 **TCM** r1,Ir2	3.3 **TCM** R2,R1	3.4 **TCM** IR2,R1	3.3 **TCM** R1,IM	3.4 **TCM** IR1,IM	4.3 **TCMX** ER2,ER1	4.3 **TCMX** IM,ER1						1.2 **STOP**
7	2.2 **PUSH** R2	2.3 **PUSH** IR2	2.3 **TM** r1,r2	2.4 **TM** r1,Ir2	3.3 **TM** R2,R1	3.4 **TM** IR2,R1	3.3 **TM** R1,IM	3.4 **TM** IR1,IM	4.3 **TMX** ER2,ER1	4.3 **TMX** IM,ER1						1.2 **HALT**
8	2.5 **DECW** RR1	2.6 **DECW** IRR1	2.5 **LDE** r1,Irr2	2.9 **LDEI** Ir1,Irr2	3.2 **LDX** r1,ER2	3.3 **LDX** Ir1,ER2	3.4 **LDX** IRR2,R1	3.5 **LDX** IRR2,IR1	3.4 **LDX** r1,rr2,X	3.4 **LDX** rr1,r2,X						1.2 **DI**
9	2.2 **RL** R1	2.3 **RL** IR1	2.5 **LDE** r2,Irr1	2.9 **LDEI** Ir2,Irr1	3.2 **LDX** r2,ER1	3.3 **LDX** Ir2,ER1	3.4 **LDX** R2,IRR1	3.5 **LDX** IR2,IRR1	3.3 **LEA** r1,r2,X	3.5 **LEA** rr1,rr2,X						1.2 **EI**
A	2.5 **INCW** RR1	2.6 **INCW** IRR1	2.3 **CP** r1,r2	2.4 **CP** r1,Ir2	3.3 **CP** R2,R1	3.4 **CP** IR2,R1	3.3 **CP** R1,IM	3.4 **CP** IR1,IM	4.3 **CPX** ER2,ER1	4.3 **CPX** IM,ER1						1.4 **RET**
B	2.2 **CLR** R1	2.3 **CLR** IR1	2.3 **XOR** r1,r2	2.4 **XOR** r1,Ir2	3.3 **XOR** R2,R1	3.4 **XOR** IR2,R1	3.3 **XOR** R1,IM	3.4 **XOR** IR1,IM	4.3 **XORX** ER2,ER1	4.3 **XORX** IM,ER1						1.5 **IRET**
C	2.2 **RRC** R1	2.3 **RRC** IR1	2.5 **LDC** r1,Irr2	2.9 **LDCI** Ir1,Irr2	2.3 **JP** IRR1	2.9 **LDC** Ir1,Irr2		3.4 **LD** r1,r2,X	3.2 **PUSHX** ER2							1.2 **RCF**
D	2.2 **SRA** R1	2.3 **SRA** IR1	2.5 **LDC** r2,Irr1	2.9 **LDCI** Ir2,Irr1	2.6 **CALL** IRR1	2.2 **BSWAP** R1	3.3 **CALL** DA	3.4 **LD** r2,r1,X	3.2 **POPX** ER1							1.2 **SCF**
E	2.2 **RR** R1	2.3 **RR** IR1	2.2 **BIT** p,b,r1	2.3 **LD** r1,Ir2	3.2 **LD** R2,R1	3.3 **LD** IR2,R1	3.2 **LD** R1,IM	3.3 **LD** IR1,IM	4.2 **LDX** ER2,ER1	4.2 **LDX** IM,ER1						1.2 **CCF**
F	2.2 **SWAP** R1	2.3 **SWAP** IR1	2.6 **TRAP** Vector	2.3 **LD** Ir1,r2	2.8 **MULT** RR1	3.3 **LD** R2,IR1	3.3 **BTJ** p,b,r1,X	3.4 **BTJ** p,b,Ir1,X								

Figure 26. First Op Code Map

	Lower Nibble (Hex)															
Upper Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7	3.2 PUS															
8																
9																
A			3.3 **CPC** r1,r2	3.4 **CPC** r1,Ir2	4.3 **CPC** R2,R1	4.4 **CPC** IR2,R1	4.3 **CPC** R1,IM	4.4 **CPC** IR1,IM	5.3 **CPCX** ER2,ER1	5.3 **CPCX** IM,ER1						
B																
C	3.2 **SRL** R1	3.3 **SRL** IR1														
D																
E									5, 4 **LDWX** ER2,ER1							
F																

Figure 27. Second Op Code Map after 1FH

Electrical Characteristics

The data in this chapter represents all known data prior to qualification and characterization of the F083A Series of products, and is therefore subject to change. Additional electrical characteristics may be found in the individual chapters of this document.

Absolute Maximum Ratings

Stresses greater than those listed in *Table 116* may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, tie unused inputs to one of the supply voltages (V_{DD} or V_{SS}).

Table 116. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units	Notes
Ambient temperature under bias	0	+105	°C	
Storage temperature	-65	+150	°C	
Voltage on any pin with respect to V_{SS}	-0.3	+5.5	V	
Voltage on V_{DD} pin with respect to V_{SS}	-0.3	+3.6	V	
Maximum current on input and/or inactive output pin	-5	+5	µA	
Maximum output current from active output pin	-25	+25	mA	
20-pin Packages Maximum Ratings at 0°C to 70°C				
Total power dissipation		430	mW	
Maximum current into V_{DD} or out of V_{SS}		120	mA	
28-pin Packages Maximum Ratings at 0°C to 70°C				
Total power dissipation		450	mW	
Maximum current into V_{DD} or out of V_{SS}		125	mA	

DC Characteristics

Table 117 lists the DC characteristics of the Z8 Encore! F083A Series products. All voltages are referenced to V_{SS}, the primary system ground.

Table 117. DC Characteristics

		T_A = 0°C to +70°C			T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
V_{DD}	Supply Voltage				2.7	–	3.6	V	power supply noise not to exceed 100 mV Peak to Peak
V_{IL1}	Low Level Input Voltage				-0.3	–	$0.3*V_{DD}$	V	For all input pins except $\overline{\text{RESET}}$.
V_{IL2}	Low Level Input Voltage				-0.3	–	0.8	V	For $\overline{\text{RESET}}$.
V_{IH1}	High Level Input Voltage				2.0	–	5.5	V	For all input pins without analog or oscillator function.
V_{IH2}	High Level Input Voltage				2.0	–	V_{DD}+ 0.3	V	For those pins with analog or oscillator function.
V_{OL1}	Low Level Output Voltage				–	–	0.4	V	I_{OL} = 2 mA; V_{DD} = 3.0 V High Output Drive disabled.
V_{OH1}	High Level Output Voltage				2.4	–	–	V	I_{OH} = -2 mA; V_{DD} = 3.0 V High Output Drive disabled.
V_{OL2}	Low Level Output Voltage				–	–	0.6	V	I_{OL} = 20 mA; V_{DD} = 3.3 V High Output Drive enabled.
V_{OH2}	High Level Output Voltage				2.4	–	–	V	I_{OH} = -20 mA; V_{DD} = 3.3 V High Output Drive enabled.
I_{IL}	Input Leakage Current				-5	–	+5	µA	V_{DD} = 3.6 V; V_{IN} = V_{DD} or V_{SS}[1]

Notes:

1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.
3. See Figure 28 on page 187 on page 187 for HALT Mode current and Figure 29 on page 187 for ACTIVE (Normal) Mode current. The typical values are taken from the chart at 20 MHz.
4. Inputs are at V_{DD}, AV_{DD}, V_{ss} or AV_{SS} power rails and outputs are floating. Pull-up enabled inputs are driven to V_{DD} or floating.
5. Typicals are at 3.3 V and 27°C.

Table 117. DC Characteristics (Continued)

		T_A = 0°C to +70°C			T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
I_{TL}	Tristate Leakage Current				-5	–	+5	µA	V_{DD} = 3.6V
I_{LED}	Controlled Current Drive				1.5	3	4.5	mA	See GPIO section on LED description
					2.8	7	10.5	mA	
					7.8	13	19.5	mA	
					12	20	30	mA	
C_{PAD}	GPIO Port Pad Capacitance				–	8.0[2]	–	pF	TBD
C_{XIN}	XIN Pad Capacitance				–	8.0[2]	–	pF	TBD
C_{XOUT}	XOUT Pad Capacitance				–	9.5[2]	–	pF	TBD
I_{PU}	Weak Pull-up Current				50	120	220	µA	V_{DD} = 2.7 - 3.6V
ICC	Supply Current in ACTIVE Mode					8		mA	V_{DD} = 2.7 - 3.6V[3,4]
ICCH	Supply Current in HALT Mode					2		mA	V_{DD} = 2.7 - 3.6V[3,4]
ICCS	Supply Current in STOP Mode			2			8	µA	Without Watchdog Timer running[3,4]

Notes:

1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.
3. See Figure 28 on page 187 on page 187 for HALT Mode current and Figure 29 on page 187 for ACTIVE (Normal) Mode current. The typical values are taken from the chart at 20 MHz.
4. Inputs are at V_{DD}, AV_{DD}, V_{ss} or AV_{ss} power rails and outputs are floating. Pull-up enabled inputs are driven to V_{DD} or floating.
5. Typicals are at 3.3 V and 27°C.

Figure 28 displays the typical current consumption while operating at 25ºC and 3.3V versus the system clock frequency in HALT Mode.

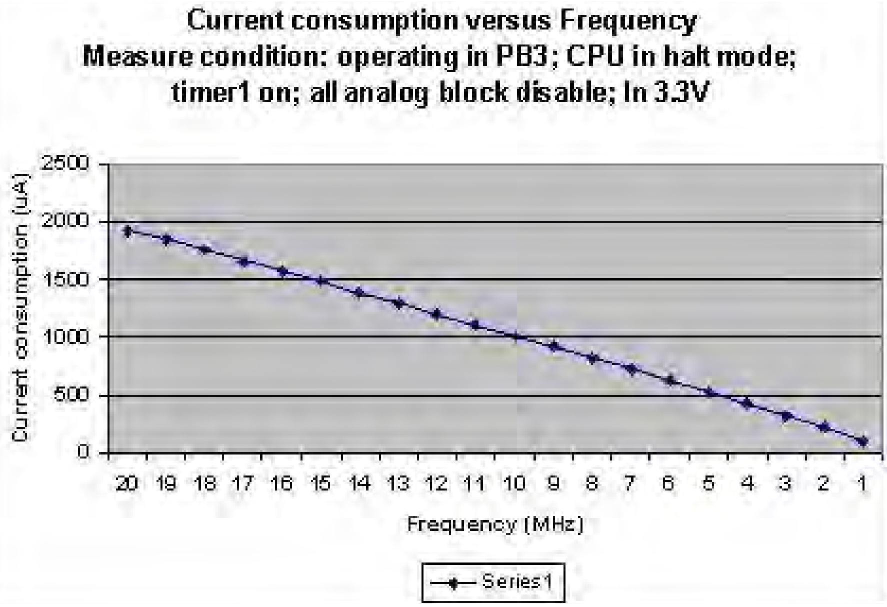


Figure 28. I_{CC} versus System Clock Frequency (HALT Mode)

Figure 29 displays the typical current consumption versus the system clock frequency in NORMAL Mode.

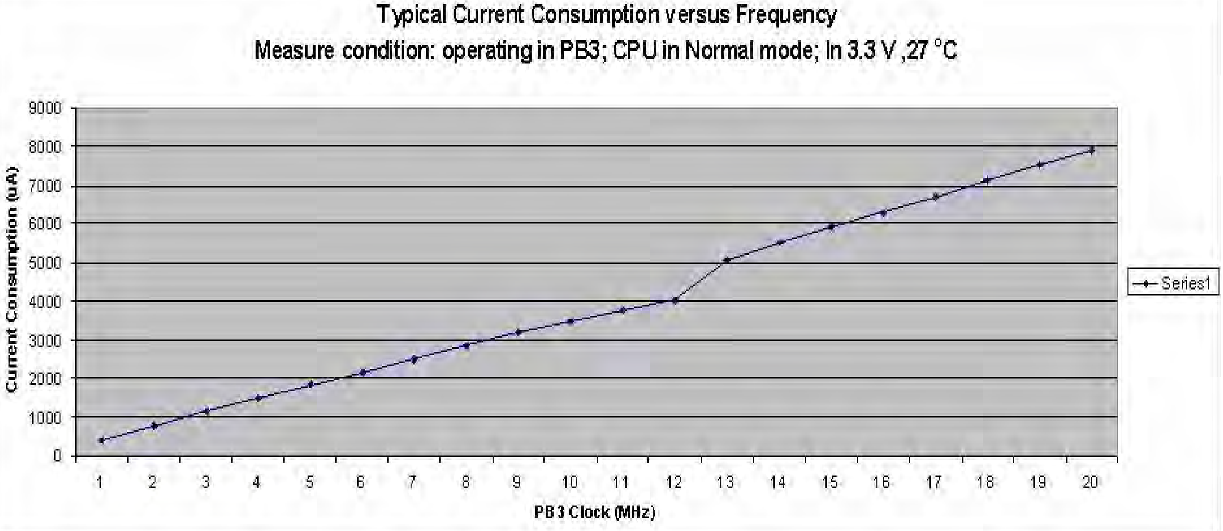


Figure 29. I_{CC} versus System Clock Frequency (NORMAL Mode)

AC Characteristics

The section provides information about the AC characteristics and timing. All AC timing information assumes a standard load of 50pF on all outputs.

Table 118. AC Characteristics

		V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C		V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C			
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Units**	**Conditions**
F_{SYSCLK}	System Clock Frequency			–	20.0	MHz	Read-only from Flash memory.
				0.032768	20.0	MHz	Program or erasure of Flash memory.
F_{XTAL}	Crystal Oscillator Frequency			1.0	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver.
F_{IPO}	Internal Precision Oscillator Frequency			0.119	20	MHz	Oscillator is not adjustable over the entire range. The user can select Min or Max value only.
F_{IPO}	Internal Precision Oscillator Frequency			19.2	20.8	MHz	High speed with trimming.
F_{IPO}	Internal Precision Oscillator Frequency			15.0	25.0	MHz	High speed without trimming.
F_{IPO}	Internal Precision Oscillator Frequency			114.2	123.8	kHz	Low speed with trimming.
F_{IPO}	Internal Precision Oscillator Frequency			89	149	kHz	Low speed without trimming.
T_{XIN}	System Clock Period			50	–	ns	$T_{CLK} = 1/F_{SYSCLK.}$
T_{XINH}	System Clock High Time			20	30	ns	T_{CLK} = 50ns.
T_{XINL}	System Clock Low Time			20	30	ns	T_{CLK} = 50ns.

Table 118. AC Characteristics (Continued)

		V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C		V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C			
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Units**	**Conditions**
T_{XINR}	System Clock Rise Time			–	3	ns	T_{CLK} = 50ns.
T_{XINF}	System Clock Fall Time			–	3	ns	T_{CLK} = 50ns.
$T_{XTALSET}$	Crystal Oscillator Setup Time			–	30,000	cycle	Crystal oscillator cycles.
T_{IPOSET}	Internal Precision Oscillator Startup Time			–	25	µs	Start-up time after enable.
T_{WDTSET}	WDT Startup Time			–	50	µs	Start-up time after reset.

On-Chip Peripheral AC and DC Electrical Characteristics

Table 119 tabulates the electrical characteristics of the POR and VBO blocks.

Table 119. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

		T_A = 0°C to +70°C			T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**[1]	**Max**	**Units**	**Conditions**
V_{POR}	POR Voltage Threshold				2.20	2.45	2.70	V	V_{DD} = V_{POR} (default VBO trim)
V_{VBO}	Voltage Brown-Out Reset Voltage Threshold				2.15	2.40	2.65	V	V_{DD} = V_{VBO} (default VBO trim)
	V_{POR} to V_{VBO} hysteresis					50	75	mV	
	Starting V_{DD} voltage to ensure valid POR.				–	V_{SS}	–	V	
T_{ANA}	POR Analog Delay				–	50	–	µs	V_{DD} > V_{POR}; T_{POR} Digital Reset delay follows T_{ANA}

Note:
1. Data in the typical column is from characterization at 3.3 V and 0°C. These values are provided for design guidance only and are not tested in production.

Table 119. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

		T_A = 0°C to +70°C			T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
T_{POR}	POR Digital Delay				TBD	13	TBD	µs	66 Internal Precision Oscillator cycles
T_{POR}	POR Digital Delay				TBD	8	TBD	ms	5000 Internal Precision Oscillator cycles
T_{SMR}	Stop Mode Recovery with crystal oscillator disabled				TBD	13	TBD	µs	66 Internal Precision Oscillator cycles
T_{SMR}	Stop Mode Recovery with crystal oscillator enabled				TBD	8	TBD	ms	5000 Internal Precision Oscillator cycles
T_{VBO}	Voltage Brown-Out Pulse Rejection Period				–	10	–	µs	$V_{DD} < V_{VBO}$ to generate a Reset.
T_{RAMP}	Time for V_{DD} to transition from V_{SS} to V_{POR} to ensure valid Reset				0.10	–	100	ms	

Note:

1. Data in the typical column is from characterization at 3.3 V and 0°C. These values are provided for design guidance only and are not tested in production.

Table 120. Flash Memory Electrical Characteristics and Timing

	V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C				
Parameter	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Notes**
Flash Byte Read Time				50	–	–	ns	
Flash Byte Program Time				20	–	–	µs	
Flash Page Erase Time				50	–	–	ms	
Flash Mass Erase Time				50	–	–	ms	
Writes to Single Address Before Next Erase				–	–	2		
Flash Row Program Time				–	–	8	ms	Cumulative program time for single row cannot exceed limit before next erase. This parameter is only an issue when bypassing the Flash Controller.
Data Retention				10	–	–	years	25°C
Endurance				10,000	–	–	cycles	Program/erase cycles

Table 121. Watchdog Timer Electrical Characteristics and Timing

		V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
	Active power consumption					2	3	µA	
F_{WDT}	WDT oscillator frequency				2.5	5	7.5	kHz	

Table 122. Nonvolatile Data Storage

	V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C				
Parameter	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Notes**
NVDS Byte Read Time				71	–	258	µs	With system clock at 20MHz
NVDS Byte Program Time				126	–	136	µs	With system clock at 20MHz
Data Retention				10	–	–	years	25°C
Endurance				100,000	–	–	cycles	Cumulative write cycles for entire memory

Note: For every 200 writes, a maintenance operation is necessary. In this rare occurrence, the write takes up to 58ms to complete.

Table 123. Analog-to-Digital Converter Electrical Characteristics and Timing

		V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
N	Resolution				–	10	–	bits	
DNL	Differential Nonlinearity[1]				–1	–	+4	LSB	
INL	Integral Nonlinearity[1]				–5	–	+5	LSB	
	Gain Error					15		LSB	
	Offset Error				–15	–	15	LSB	PDIP package
					–9	–	9	LSB	Other packages
V_{REF}	On chip reference				1.9	2.0	2.1	V	
I_{DD}ADC	ADC Active Current					4		mA	

Note:

1. When the input voltage is lower than 20mV, the conversion error is out of specification tolerance.

Table 123. Analog-to-Digital Converter Electrical Characteristics and Timing (Continued)

		V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
I_{DD2}ADC	ADC Quiescent Current						1	µA	
Z_{IN}	Input Impedance				10			MΩ	
V_{IN}	Input Voltage Range				0		2.0	V	Internal reference.
					0		0.9*V_{DD}		External reference.
T_{CONV}	Conversion Time				2.8			µs	10MHz (ADC Clock)
GBW_{IN}	Input Bandwidth					350		kHz	
T_{WAKE}	Wake Up Time					0.02		ms	Internal reference.
						10		ms	External reference.
	Input Clock Duty				45	50	55	%	
f_{ADC_CLK}	Maximum Frequency of ADC_CLK						10	MHz	

Note:
1. When the input voltage is lower than 20mV, the conversion error is out of specification tolerance.

Table 124. Comparator Electrical Characteristics

		V_{DD} = 2.7V to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
V_{OS}	Input DC Offset					5		mV	
V_{CREF}	Programmable Internal Reference Voltage Range				0		1.8	V	User-programmable in 200 mV step
V_{CREF}	Programmable internal reference voltage				0.92	1.0	1.08	V	Default (CMP0[REFLVL]=5H)
T_{PROP}	Propagation delay					100		ns	
V_{HYS}	Input hysteresis					8		mV	

General Purpose I/O Port Input Data Sample Timing

Figure 30 and Table 125 display timing data for the GPIO port input sampling operation. The input value on a GPIO port pin is sampled on the rising edge of the system clock. The port value is available to the eZ8 CPU on the second rising clock edge following the change of the port value.

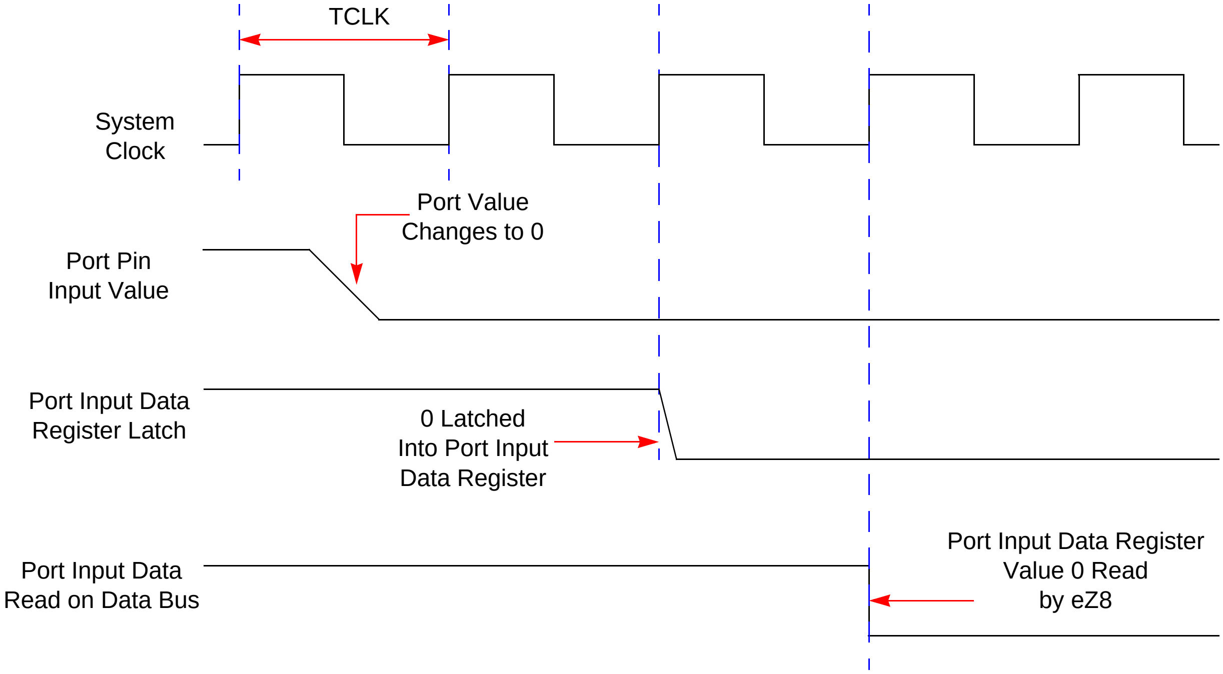

TCLK
System Clock
Port Value Changes to 0
Port Pin Input Value
Port Input Data Register Latch
0 Latched Into Port Input Data Register
Port Input Data Register Value 0 Read by eZ8
Port Input Data Read on Data Bus

Figure 30. Port Input Sample Timing

Table 125. GPIO Port Input Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_{S_PORT}	Port Input Transition to XIN Rise Setup Time (not pictured)	5	–
T_{H_PORT}	X_{IN} Rise to Port Input Transition Hold Time (not pictured)	0	–
T_{SMR}	GPIO Port Pin Pulse Width to ensure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources)	1µs	

GPIO Port Output Timing

Figure 31 and Table 126 provide timing information for the GPIO port pins.

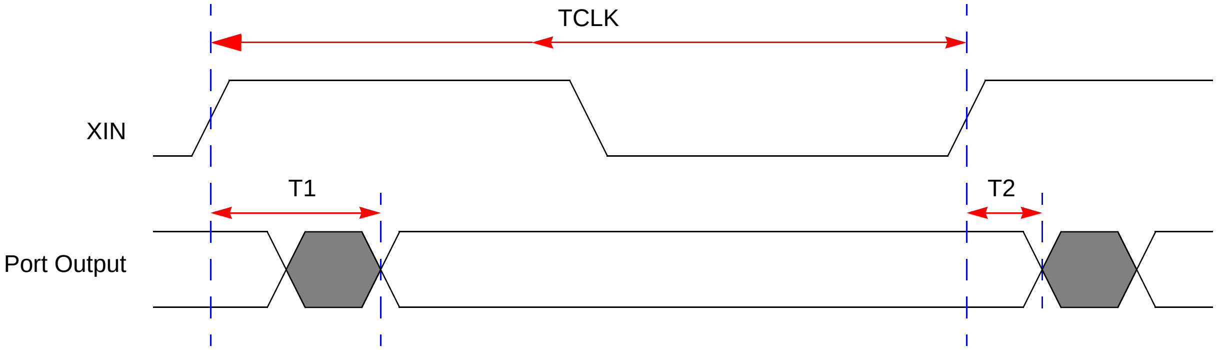


Figure 31. GPIO Port Output Timing

Table 126. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
GPIO Port pins			
T_1	XIN Rise to Port Output Valid Delay	–	15
T_2	XIN Rise to Port Output Hold Time	2	–

On-Chip Debugger Timing

Figure 32 and Table 127 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4ns maximum rise and fall time.

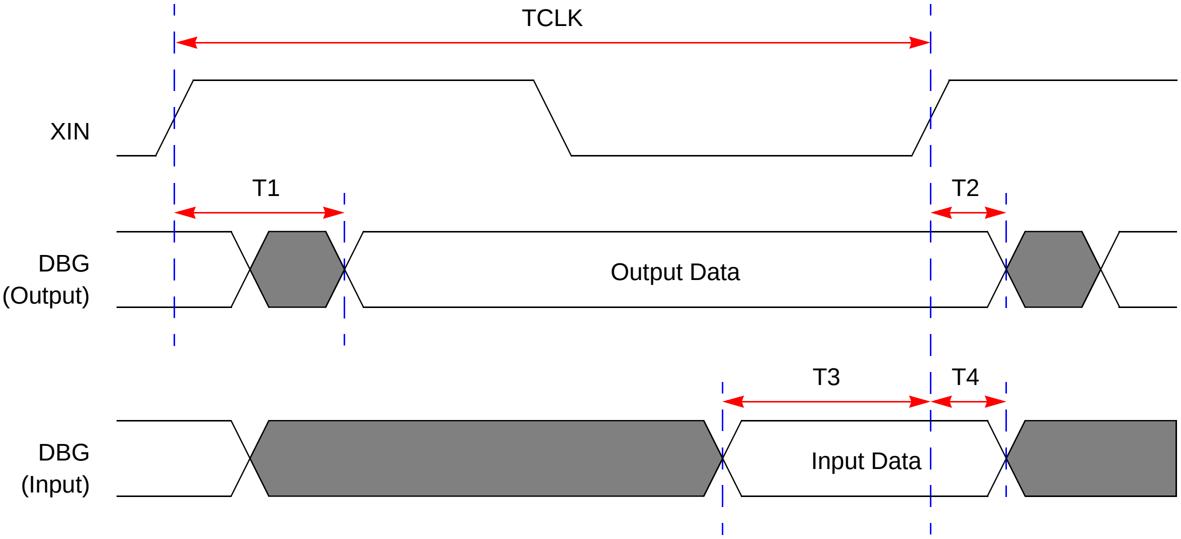


Figure 32. On-Chip Debugger Timing

Table 127. On-Chip Debugger Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
DBG			
T_1	X_{IN} Rise to DBG Valid Delay	–	15
T_2	X_{IN} Rise to DBG Output Hold Time	2	–
T_3	DBG to XIN Rise Input Setup Time	5	–
T_4	DBG to XIN Rise Input Hold Time	5	–

Table 128. Power Consumption Reference Table

Category	Block	Power Consumption Typical	Maximum
Logic	CPU/Peripherals @20MHz	5mA	
Flash	Flash@20MHz		12mA
Analog	ADC@20MHz	4mA	4.5mA
	IPO@20MHz	350µA	400µA
	Comparator@10MHz	330µA	450µA
	POR & VBO	120µA	150µA
	WDT OSC	2µA	3µA
	OSC@20MHz	600µA	900µA
	Clock Filter	120µA	150µA

Note: The power consumption values in this table are subject to change upon characterization.

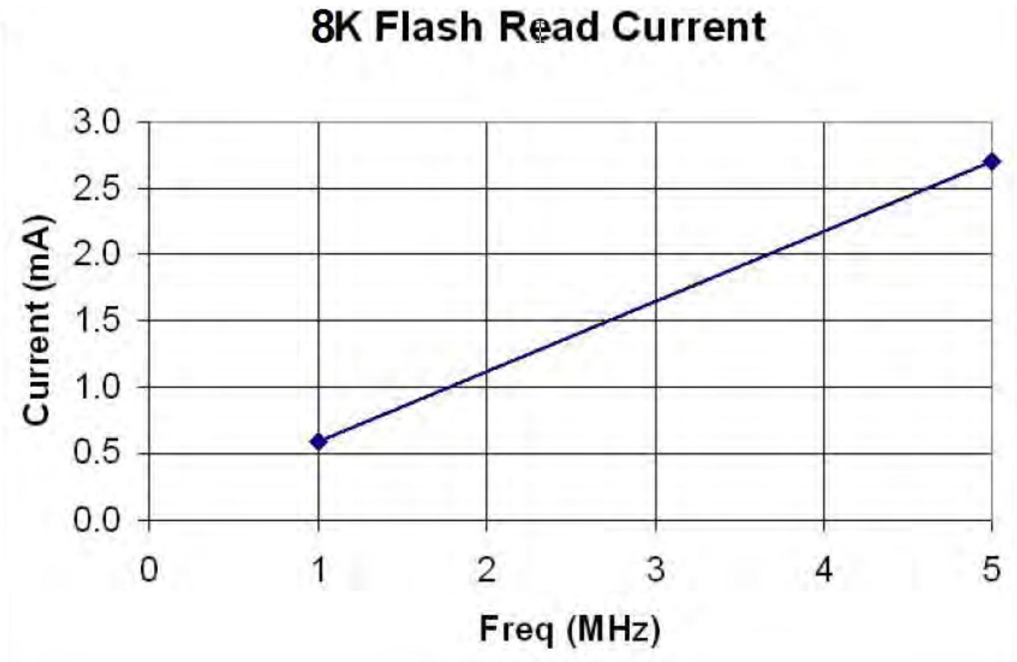


Figure 33. Flash Current Diagram

Packaging

Zilog's F083A Series of MCUs includes the Z8F043A and Z8F083A devices, which are available in the following packages:

- 20-Pin Quad Flat No-Lead Package (QFN)
- 20-pin Small Outline Integrated Circuit Package (SOIC)
- 20-pin Plastic Dual-Inline Package (PDIP)
- 20-pin Small Shrink Outline Package (SSOP)
- 28-Pin Quad Flat No-Lead Package (QFN)
- 28-pin Small Outline Integrated Circuit Package (SOIC)
- 28-pin Plastic Dual-Inline Package (PDIP)
- 28-pin Small Shrink Outline Package (SSOP)

Current diagrams for each of these packages are published in Zilog's Packaging Product Specification (PS0072), which is available free for download from the Zilog website.

Ordering Information

Order your F083A Series products from Zilog using the part numbers shown in Table 129. For more information about ordering, please consult your local Zilog sales office. The Sales Location page on the Zilog website lists all regional offices.

Table 129. Z8 Encore! F083A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	ADC Channels	Description
Z8 Encore! F083A with 8KB Flash					
Standard Temperature: 0°C to 70°C					
Z8F083ASH020SG	8KB	256	100 B	7	SOIC 20-pin
Z8F083AHH020SG	8KB	256	100 B	7	SSOP 20-pin
Z8F083APH020SG	8KB	256	100 B	7	PDIP 20-pin
Z8F083AQH020SG	8KB	256	100 B	7	QFN 20-pin
Z8F083ASJ020SG	8KB	256	100 B	8	SOIC 28-pin
Z8F083AHJ020SG	8KB	256	100 B	8	SSOP 28-pin
Z8F083AQJ020SG	8KB	256	100 B	8	QFN 28-pin
Extended Temperature: –40°C to 105°C					
Z8F083ASH020EG	8KB	256	100 B	7	SOIC 20-pin
Z8F083AHH020EG	8KB	256	100 B	7	SSOP 20-pin
Z8F083APH020EG	8KB	256	100 B	7	PDIP 20-pin
Z8F083AQH020EG	8KB	256	100 B	7	QFN 20-pin
Z8F083ASJ020EG	8KB	256	100 B	8	SOIC 28-pin
Z8F083AHJ020EG	8KB	256	100 B	8	SSOP 28-pin
Z8F083AQJ020EG	8KB	256	100 B	8	QFN 28-pin
Z8 Encore! F083A with 4KB Flash					
Standard Temperature: 0°C to 70°C					
Z8F043ASH020SG	4KB	256	100 B	7	SOIC 20-pin
Z8F043AHH020SG	4KB	256	100 B	7	SSOP 20-pin
Z8F043APH020SG	4KB	256	100 B	7	PDIP 20-pin
Z8F043AQH020SG	4KB	256	100 B	7	QFN 20-pin
Z8F043ASJ020SG	4KB	256	100 B	8	SOIC 28-pin
Z8F043AHJ020SG	4KB	256	100 B	8	SSOP 28-pin
Z8F043AQJ020SG	4KB	256	100 B	8	QFN 28-pin

Table 129. Z8 Encore! F083A Series Ordering Matrix (Continued)

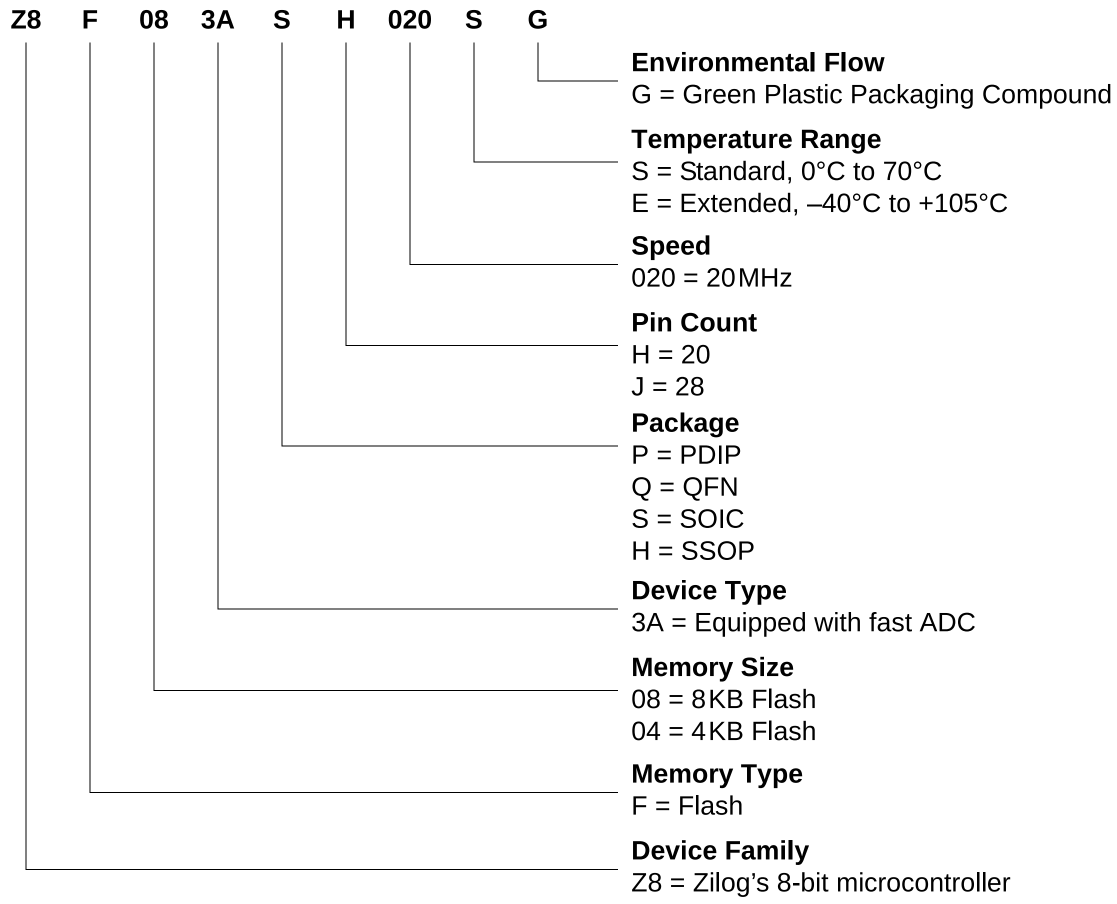
Part Number	Flash	RAM	NVDS	ADC Channels	Description
Extended Temperature: –40°C to 105°C					
Z8F043ASH020EG	4KB	256	100 B	7	SOIC 20-pin
Z8F043AHH020EG	4KB	256	100 B	7	SSOP 20-pin
Z8F043APH020EG	4KB	256	100 B	7	PDIP 20-pin
Z8F043AQH020EG	4KB	256	100 B	7	QFN 20-pin
Z8F043ASJ020EG	4KB	256	100 B	8	SOIC 28-pin
Z8F043AHJ020EG	4KB	256	100 B	8	SSOP 28-pin
Z8F043AQJ020EG	4KB	256	100 B	8	QFN 28-pin
ZUSBOPTSC01ZACG					Opto-Isolated USB Smart Cable Accessory Kit
ZUSBSC00100ZACG					USB Smart Cable Accessory Kit
ZENETSC0100ZACG					Ethernet Smart Cable Accessory Kit
Z8F083A0128ZCOG					F083A Development Kit

Visit the Zilog website at http://www.zilog.com for ordering information about Z8 Encore! F083A Series development tools and accessories.

Part Number Suffix Designations

Zilog part numbers consist of a number of components, as indicated in the following example.

Example. Part number Z8F083ASH020SG is an 8-bit 20 MHz Flash MCU with 8 KB of Program Memory, equipped with a fast ADC in a 20-pin SOIC package, operating within a 0ºC to +70ºC temperature range and built using lead-free solder.



Z8 F 08 3A S H 020 S G

Environmental Flow
G = Green Plastic Packaging Compound

Temperature Range
S = Standard, 0°C to 70°C
E = Extended, –40°C to +105°C

Speed
020 = 20 MHz

Pin Count
H = 20
J = 28

Package
P = PDIP
Q = QFN
S = SOIC
H = SSOP

Device Type
3A = Equipped with fast ADC

Memory Size
08 = 8 KB Flash
04 = 4 KB Flash

Memory Type
F = Flash

Device Family
Z8 = Zilog’s 8-bit microcontroller

In Table 130, a "√" mark indicates F083A Series package availability by pin count.

Table 130. Package and Pin Count Description

	Pin Count	
Package	20	28
PDIP	√	
QFN	√	√
SOIC	√	√
SSOP	√	√

Appendix A. Register Tables

For the reader’s convenience, this appendix lists all F083A Series registers numerically by hexadecimal address.

General Purpose RAM

In the F083A Series, the `000–EFF` hexadecimal address range is partitioned for general-purpose random access memory, as follows.

Hex Addresses: 000–0FF

This address range is reserved for general-purpose register file RAM. For more details, see the Register File section on page 14.

Hex Addresses: 100–EFF

This address range is reserved.

Timer 0

For more information about these Timer Control registers, see the Timer Control Register Definitions section on page 83.

Hex Address: F00

Table 131. Timer 0 High Byte Register (T0H)

Bit	7	6	5	4	3	2	1	0
Field	TH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F00H							

Hex Address: F01

Table 132. Timer 0 Low Byte Register (T0L)

Bit	7	6	5	4	3	2	1	0
Field	TL							
RESET	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F01H							

Hex Address: F02

Table 133. Timer 0 Reload High Byte Register (T0RH)

Bit	7	6	5	4	3	2	1	0
Field	TRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F02H							

Hex Address: F03

Table 134. Timer 0 Reload Low Byte Register (T0RL)

Bit	7	6	5	4	3	2	1	0
Field	TRL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F03H							

Hex Address: F04

Table 135. Timer 0 PWM High Byte Register (T0PWMH)

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F04H							

Hex Address: F05

Table 136. Timer 0 PWM Low Byte Register (T0PWML)

Bit	7	6	5	4	3	2	1	0
Field	PWML							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F05H							

Hex Address: F06

Table 137. Timer 0 Control Register 0 (T0CTL0)

Bit	7	6	5	4	3	2	1	0
Field	TMODEHI	TICONFIG		Reserved	PWMD			INPCAP
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F06H							

Hex Address: F07

Table 138. Timer 0 Control Register 1 (T0CTL1)

Bit	7	6	5	4	3	2	1	0
Field	TEN	TPOL	PRES			TMODE		
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F07H							

Hex Address: F08

Table 139. Timer 1 High Byte Register (T1H)

Bit	7	6	5	4	3	2	1	0
Field	TH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F08H							

Hex Address: F09

Table 140. Timer 1 Low Byte Register (T1L)

Bit	7	6	5	4	3	2	1	0
Field	TL							
RESET	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F09H							

Hex Address: F0A

Table 141. Timer 1 Reload High Byte Register (T1RH)

Bit	7	6	5	4	3	2	1	0
Field	TRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F0AH							

Hex Address: F0B

Table 142. Timer 1 Reload Low Byte Register (T1RL)

Bit	7	6	5	4	3	2	1	0
Field	TRL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F0BH							

Hex Address: F0C

Table 143. Timer 1 PWM High Byte Register (T1PWMH)

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F0CH							

Hex Address: F0D

Table 144. Timer 1 PWM Low Byte Register (T1PWML)

Bit	7	6	5	4	3	2	1	0
Field	PWML							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F0DH							

Hex Address: F0E

Table 145. Timer 1 Control Register 0 (T1CTL0)

Bit	7	6	5	4	3	2	1	0
Field	TMODEHI	TICONFIG		Reserved	PWMD			INPCAP
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F0EH							

Hex Address: F0F

Table 146. Timer 1 Control Register 1 (T1CTL1)

Bit	7	6	5	4	3	2	1	0
Field	TEN	TPOL	PRES			TMODE		
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F0FH							

Hex Addresses: F10–F6F

This address range is reserved.

Analog-to-Digital Converter (ADC)

For more information about these ADC registers, see the ADC Control Register Definitions section on page 102.

Hex Address: F70

Table 147. ADC Control Register 0 (ADCCTL0)

Bit	7	6	5	4	3	2	1	0
Field	START	Reserved	REFEN	ADCEN	Reserved	ANAIN[2:0]		
RESET	0	0	0	0	0	0	0	0
R/W	R/W1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F70h							

Bit Position	Description
[7] START	**ADC Start/Busy** 0 = Writing to 0 has no effect. Reading a 0 indicates that the ADC is available to begin a conversion. 1 = Writing to 1 starts a conversion. Reading a 1 indicates that a conversion is currently in progress.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] REFEN	**Reference Enable** 0 = Internal reference voltage is disabled, allowing an external reference voltage to be used by the ADC. 1 = Internal reference voltage for the ADC is enabled. The internal reference voltage is measured on the VREF pin.
[4] ADCEN	**ADC Enable** 0 = ADC is disabled for low power operation. 1 = ADC is enabled for normal use.
[3]	**Reserved** This bit is reserved and must be programmed to 0.

Bit Position	Description (Continued)
[2:0] ANAIN	**Analog Input Select** 000 = ANA0 input is selected for analog-to-digital conversion.
	001 = ANA1 input is selected for analog-to-digital conversion.
	010 = ANA2 input is selected for analog-to-digital conversion.
	011 = ANA3 input is selected for analog-to-digital conversion.
	100 = ANA4 input is selected for analog-to-digital conversion.
	101 = ANA5 input is selected for analog-to-digital conversion.
	110 = ANA6 input is selected for analog-to-digital conversion.
	111 = ANA7 input is selected for analog-to-digital conversion.

Hex Address: F71

This address range is reserved.

Hex Address: F72

Table 148. ADC Data High Byte Register (ADCD_H)

Bit	7	6	5	4	3	2	1	0
Field	ADCDH							
RESET	X							
R/W	R							
Address	F72H							

Bit Position	Description
[7:0]	**ADC High Byte** 00h–FFh = The last conversion output is held in the data registers until the next ADC conversion is completed.

Hex Address: F73

Table 149. ADC Data Low Bits Register (ADCD_L)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	ADCDL		Reserved					
RESET	X		X					
R/W	R		R					
Address	F73H							

Bit Position	**Description**
[7:6]	**ADC Low Bits** 00–11b = These bits are the two least significant bits of the 10-bit ADC output. These bits are undefined after a reset. The low bits are latched into this register whenever the ADC Data High Byte Register is read.
[5:0]	**Reserved** These bits are reserved and must be programmed to 000000.

Hex Address: F74

Table 150. ADC Sample Settling Time (ADCSST)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved				SST			
RESET	0				1	1	1	1
R/W	R				R/W			
Address	F74H							

Bit Position	**Description**
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:0] SST	**Sample Setting Time** 0h–Fh = Sample settling time in number of system clock periods to meet 0.5µs minimum.

Hex Address: F75

Table 151. ADC Sample Time (ADCST)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved		ST					
RESET	0		1	1	1	1	1	1
R/W	R/W		R/W					
Address	F75H							

Bit Position	**Description**
[7:6]	**Reserved** These bits are reserved and must be programmed to 00.
[5:0] ST	**Sample Time** 0h–Fh = Sample/hold time in number of system clock periods to meet 1µs minimum.

Hex Address: F76

Table 152. ADC Clock Prescale Register (ADCCP)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved					DIV8	DIV4	DIV2
RESET	0					0	0	0
R/W	R/W							
Address	F76H							

Bit Position	**Description**
[0] DIV2	**DIV2** 0 = Clock is not divided 1 = System clock is divided by 2 for ADC clock
[1] DIV4	**DIV4** 0 = Clock is not divided 1 = System clock is divided by 4 for ADC clock
[2] DIV8	**DIV8** 0 = Clock is not divided 1 = System clock is divided by 8 for ADC clock
[7:3]	**Reserved** These bits are reserved and must be programmed to 00000.

Hex Addresses: F77–F7F

This address range is reserved.

Low Power Control

For more information about the Power Control Register, see the Power Control Register Definitions section on page 31.

Hex Address: F80

Table 153. Power Control Register 0 (PWRCTL0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			VBO	Reserved	Reserved	COMP	Reserved
RESET	1	0	0	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F80H							

Hex Address: F81

This address is reserved.

LED Controller

For more information about the LED Drive registers, see the GPIO Control Register Definitions section on page 39.

Hex Address: F82

Table 154. LED Drive Enable (LEDEN)

Bit	7	6	5	4	3	2	1	0
Field	LEDEN[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F82H							

Hex Address: F83

Table 155. LED Drive Level High Register (LEDLVLH)

Bit	7	6	5	4	3	2	1	0
Field	LEDLVLH[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F83H							

Hex Address: F84

Table 156. LED Drive Level Low Register (LEDLVLL)

Bit	7	6	5	4	3	2	1	0
Field	LEDLVLL[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F84H							

Hex Address: F85

This address is reserved.

Oscillator Control

For more information about the Oscillator Control registers, see the Oscillator Control Register Definitions section on page 154.

Hex Address: F86

Table 157. Oscillator Control Register (OSCCTL)

Bit	7	6	5	4	3	2	1	0
Field	INTEN	XTLEN	WDTEN	POFEN	WDFEN	SCKSEL		
RESET	1	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F86H							

Hex Addresses: F87–F8F

This address range is reserved.

Comparator 0

For more information about the Comparator Register, see the Comparator Control Register Definition section on page 109.

Hex Address: F90

Table 158. Comparator Control Register (CMP0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	INNSEL	REFLVL				Reserved	
RESET	0	0	0	1	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F90H							

Hex Addresses: F91–FBF

This address range is reserved.

Interrupt Controller

For more information about the Interrupt Control registers, see the Interrupt Control Register Definitions section on page 58.

Hex Address: FC0

Table 159. Interrupt Request 0 Register (IRQ0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1I	T0I	Reserved	Reserved	Reserved	Reserved	ADCI
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC0H							

Hex Address: FC1

Table 160. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	T1ENH	T0ENH	Reserved	Reserved	Reserved	Reserved	ADCENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC1H							

Hex Address: FC2

Table 161. IRQ0 Enable Low Bit Register (IRQ0ENL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	T1ENL	T0ENL	Reserved	Reserved	Reserved	Reserved	ADCENL
RESET	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R	R	R/W
Address	FC2H							

Hex Address: FC3

Table 162. Interrupt Request 1 Register (IRQ1)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PA7I	PA6CI	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC3H							

Hex Address: FC4

Table 163. IRQ1 Enable High Bit Register (IRQ1ENH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PA7ENH	PA6CENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC4H							

Hex Address: FC5

Table 164. IRQ1 Enable Low Bit Register (IRQ1ENL)

Bit	7	6	5	4	3	2	1	0
Field	PA7ENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC5H							

Hex Address: FC6

Table 165. Interrupt Request 2 Register (IRQ2)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				PC3I	PC2I	PC1I	PC0I
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC6H							

Hex Address: FC7

Table 166. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				C3ENH	C2ENH	C1ENH	C0ENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC7H							

Hex Address: FC8

Table 167. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				C3ENL	C2ENL	C1ENL	C0ENL
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC8H							

Hex Addresses: FC9–FCC

This address range is reserved.

Hex Address: FCD

Table 168. Interrupt Edge Select Register (IRQES)

Bit	7	6	5	4	3	2	1	0
Field	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FCDH							

Hex Address: FCE

Table 169. Shared Interrupt Select Register (IRQSS)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	PA6CS	Reserved					
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FCEH							

Hex Address: FCF

Table 170. Interrupt Control Register (IRQCTL)

Bit	7	6	5	4	3	2	1	0
Field	IRQE	Reserved						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R
Address	FCFH							

GPIO Port A

For more information about the GPIO registers, see the GPIO Control Register Definitions section on page 39.

Hex Address: FD0

Table 171. Port A GPIO Address Register (PAADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD0H							

Hex Address: FD1

Table 172. Port A Control Registers (PACTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD1H							

Hex Address: FD2

Table 173. Port A Input Data Registers (PAIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	FD2H							

Hex Address: FD3

Table 174. Port A Output Data Register (PAOUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD3H							

Hex Address: FD4

Table 175. Port B GPIO Address Register (PBADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD4H							

Hex Address: FD5

Table 176. Port B Control Registers (PBCTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD5H							

Hex Address: FD6

Table 177. Port B Input Data Registers (PBIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	FD6H							

Hex Address: FD7

Table 178. Port B Output Data Register (PBOUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD7H							

Hex Address: FD8

Table 179. Port C GPIO Address Register (PCADDR)

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD8H							

Hex Address: FD9

Table 180. Port C Control Registers (PCCTL)

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FD9H							

Hex Address: FDA

Table 181. Port C Input Data Registers (PCIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	FDAH							

Hex Address: FDB

Table 182. Port C Output Data Register (PCOUT)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FDBH							

Hex Address: FDC

Table 183. Port D GPIO Address Register (PDADDR)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FDCH							

Hex Address: FDD

Table 184. Port D Control Registers (PDCTL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FDDH							

Hex Address: FDE

This address is reserved.

Hex Address: FDF

Table 185. Port D Output Data Register (PDOUT)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FDFH							

Hex Addresses: FE0–FEF

This address range is reserved.

Watchdog Timer (WDT)

For more information about the Watchdog Timer registers, see the Watchdog Timer Control Register Definitions section on page 95.

Hex Address: FF0

This register address is shared with the read-only Reset Status Register.

Table 186. Watchdog Timer Control Register (WDTCTL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	WDTUNLK							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
Address	FF0H							

Table 187. Reset Status Register (RSTSTAT)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	POR	STOP	WDT	EXT	Reserved			
RESET	See Table 12 on page 29			0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	FF0H							

Hex Address: FF1

Table 188. Watchdog Timer Reload Upper Byte Register (WDTU)

Bit	7	6	5	4	3	2	1	0
Field	WDTU							
RESET	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF1H							
Note: *R = Read returns the current WDT count value; W = Write sets the appropriate reload value.								

Hex Address: FF2

Table 189. Watchdog Timer Reload High Byte Register (WDTH)

Bit	7	6	5	4	3	2	1	0
Field	WDTH							
RESET	0	0	0	0	0	1	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF2H							
Note: *R = Read returns the current WDT count value; W = Write sets the appropriate reload value.								

Hex Address: FF3

Table 190. Watchdog Timer Reload Low Byte Register (WDTL)

Bit	7	6	5	4	3	2	1	0
Field	WDTL							
RESET	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF3H							
Note: *R = Read returns the current WDT count value; W = Write sets the appropriate reload value.								

Hex Addresses: FF4–FF5

This address range is reserved.

Trim Bit Control

For more information about the Trim Bit Control registers, see the Flash Option Bit Control Register Definitions section on page 126.

Hex Address: FF6

Table 191. Trim Bit Address Register (TRMADR)

Bit	7	6	5	4	3	2	1	0
Field	TRMADR - Trim Bit Address (00H to 1FH)							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF6H							

Hex Address: FF7

Table 192. Trim Bit Data Register (TRMDR)

Bit	7	6	5	4	3	2	1	0
Field	TRMDR - Trim Bit Data							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF7H							

Flash Memory Controller

For more information about the Flash Control registers, see the Flash Control Register Definitions section on page 119.

Hex Address: FF8

Table 193. Flash Control Register (FCTL)

Bit	7	6	5	4	3	2	1	0
Field	FCMD							
RESET	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	FF8H							

Hex Address: FF8

Table 194. Flash Status Register (FSTAT)

Bit	7	6	5	4	3	2	1	0
Field	Reserved		FSTAT					
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	FF8H							

Hex Address: FF9

The Flash Page Select Register, shown in Table 195, is shared with the Flash Sector Protect Register, which is shown in Table 196.

Table 195. Flash Page Select Register (FPS)

Bit	7	6	5	4	3	2	1	0
Field	INFO_EN	PAGE						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF9H							

Table 196. Flash Sector Protect Register (FPROT)

Bit	7	6	5	4	3	2	1	0
Field	SPROT7	SPROT6	SPROT5	SPROT4	SPROT3	SPROT2	SPROT1	SPROT0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF9H							

Hex Address: FFA

Table 197. Flash Frequency High Byte Register (FFREQH)

Bit	7	6	5	4	3	2	1	0
Field	FFREQH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FFAH							

Hex Address: FFB

Table 198. Flash Frequency Low Byte Register (FFREQL)

Bit	7	6	5	4	3	2	1	0
Field	FFREQL							
RESET	0							
R/W	R/W							
Address	FFBH							

Index

Customer Support

To share comments, get your technical questions answered or report issues you may be experiencing with our products, please visit Zilog’s Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.