



Welcome to [E-XFL.COM](http://E-XFL.COM)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	5MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	22
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f0123sj005sc">https://www.e-xfl.com/product-detail/zilog/z8f0123sj005sc</a>

Flash Information Area .....	130
Operation .....	131
Flash Operation Timing Using the Flash Frequency Registers .....	133
Flash Code Protection Against External Access .....	133
Flash Code Protection Against Accidental Program and Erasure .....	133
Byte Programming .....	135
Page Erase .....	135
Mass Erase .....	135
Flash Controller Bypass .....	136
Flash Controller Behavior in DEBUG Mode .....	136
Flash Control Register Definitions .....	137
Flash Control Register .....	137
Flash Status Register .....	137
Flash Page Select Register .....	138
Flash Sector Protect Register .....	139
Flash Frequency High and Low Byte Registers .....	139
<b>Flash Option Bits .....</b>	<b>141</b>
Operation .....	141
Option Bit Configuration By Reset .....	141
Option Bit Types .....	141
Reading the Flash Information Page .....	143
Flash Option Bit Control Register Definitions .....	143
Trim Bit Address Register .....	143
Trim Bit Data Register .....	144
Flash Option Bit Address Space .....	144
Flash Program Memory Address 0000H .....	144
Flash Program Memory Address 0001H .....	145
Trim Bit Address Space .....	146
Trim Bit Address 0000H—Reserved .....	146
Trim Bit Address 0001H—Reserved .....	146
Trim Bit Address 0002H .....	147
Trim Bit Address 0003H—Reserved .....	147
Trim Bit Address 0004H—Reserved .....	147
Zilog Calibration Data .....	147
ADC Calibration Data .....	147
Serialization Data .....	148
Randomized Lot Identifier .....	149
<b>On-Chip Debugger .....</b>	<b>151</b>
Architecture .....	151

**Ordering Information** ..... **217**  
    Part Number Suffix Designations ..... 226  
**Index** ..... **227**  
**Customer Support** ..... **237**

► **Note:** *This register is only reset during a Power-On Reset sequence. Other System Reset events do not affect it.*

**Table 13. Power Control Register 0 (PWRCTL0)**

<b>BITS</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>FIELD</b>	Reserved	Reserved		VBO	Reserved	ADC	COMP	Reserved
<b>RESET</b>	1	0	0	0	0	0	0	0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>ADDR</b>	F80H							

Reserved—Must be 1

Reserved—Must be 0

VBO—Voltage Brownout Detector Disable

This bit and the VBO\_AO Flash option bit must both enable the VBO for the VBO to be active.

0 = VBO enabled

1 = VBO disabled

ADC—Analog-to-Digital Converter Disable

0 = Analog-to-Digital Converter enabled

1 = Analog-to-Digital Converter disabled

COMP—Comparator Disable

0 = Comparator is enabled

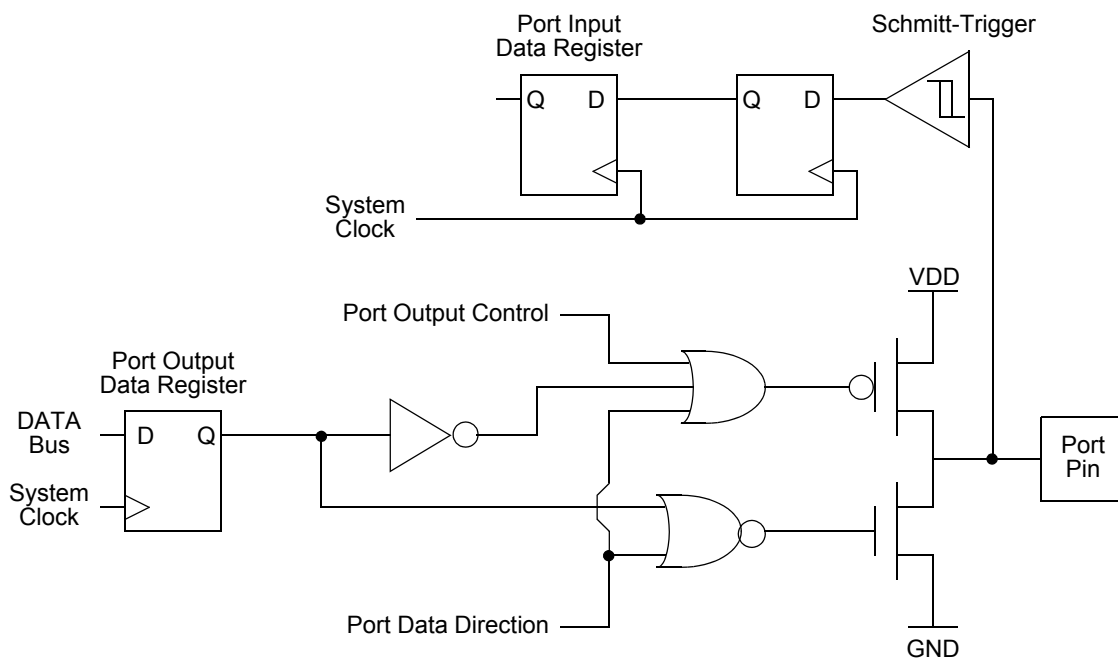
1 = Comparator is disabled

Reserved—Must be 0



## Architecture

Figure 7 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 7. GPIO Port Pin Block Diagram**

## GPIO Alternate Functions

Many of the GPIO port pins are used for general-purpose I/O and access to on-chip peripheral functions such as the timers and serial communication devices. The port A–D Alternate Function sub-registers 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 15 on page 39 lists the alternate functions possible with each port pin. The alternate function associated at a pin is defined through Alternate Function Sets sub-registers 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, those pins function as input and output for the crystal oscillator.

**Table 15. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)**

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port C	PC0	Reserved		AFS1[0]: 0
		ANA4/CINP/LED Drive	ADC or Comparator Input, or LED drive	AFS1[0]: 1
	PC1	Reserved		AFS1[1]: 0
		ANA5/CINN/ LED Drive	ADC or Comparator Input, or LED drive	AFS1[1]: 1
	PC2	Reserved		AFS1[2]: 0
		ANA6/LED/ VREF*	ADC Analog Input or LED Drive or ADC Voltage Reference	AFS1[2]: 1
	PC3	COUT	Comparator Output	AFS1[3]: 0
		LED	LED drive	AFS1[3]: 1
	PC4	Reserved		AFS1[4]: 0
		LED	LED Drive	AFS1[4]: 1
	PC5	Reserved		AFS1[5]: 0
		LED	LED Drive	AFS1[5]: 1
	PC6	Reserved		AFS1[6]: 0
		LED	LED Drive	AFS1[6]: 1
	PC7	Reserved		AFS1[7]: 0
		LED	LED Drive	AFS1[7]: 1

**Note:** Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, Alternate Function selection as described in Port A–C Alternate Function Sub-Registers must also be enabled.

\*VREF is available on PC2 in 20-pin parts only.

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).

### Port A–C High Drive Enable Sub-Registers

The Port A–C High Drive Enable sub-register (Table 23) is accessed through the Port A–C Control register by writing 04H to the Port A–C Address register. Setting the bits in the Port A–C High Drive Enable sub-registers to 1 configures the specified port pins for high current output drive operation. The Port A–C High Drive Enable sub-register affects the pins directly and, as a result, alternate functions are also affected.

**Table 23. Port A–C High Drive Enable Sub-Registers (PxHDE)**

BITS	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
ADDR	If 04H in Port A–C Address Register, accessible through the Port A–C Control Register							

PHDE[7:0]—Port High Drive Enabled.

0 = The Port pin is configured for standard output current drive.

1 = The Port pin is configured for high output current drive.

### Port A–C Stop Mode Recovery Source Enable Sub-Registers

The Port A–C Stop Mode Recovery Source Enable sub-register (Table 24) is accessed through the Port A–C Control register by writing 05H to the Port A–C Address register. Setting the bits in the Port A–C Stop Mode Recovery Source Enable sub-registers to 1 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 24. Port A–C Stop Mode Recovery Source Enable Sub-Registers (PxSMRE)**

BITS	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
ADDR	If 05H in Port A–C Address Register, accessible through the Port A–C Control Register							



**Table 30. LED Drive Enable (LEDEN)**

BITS	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
ADDR	F82H							

LEDEN[7:0]—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 registers contain two control bits for each Port C pin (Table 31).

These two bits select between four programmable drive levels. Each pin is individually programmable.

**Table 31. LED Drive Level High Register (LEDLVLH)**

BITS	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
ADDR	F83H							

LEDLVLH[7:0]—LED Level High Bit

{LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin.

00 = 3 mA

01= 7 mA

10= 13 mA

11= 20 mA

### LED Drive Level Low Register

The LED Drive Level registers contain two control bits for each Port C pin (Table 32).

These two bits select between four programmable drive levels. Each pin is individually programmable.

- Writing a 1 to the IRQE bit in the Interrupt Control register

Interrupts are globally disabled by any of the following actions:

- Execution of a Disable Interrupt (DI) 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 Timer 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 interrupts are enabled with identical interrupt priority (for example, all as Level 2 interrupts), the interrupt priority is assigned from highest to lowest as specified in Table 33 on page 54. Level 3 interrupts are always assigned higher priority than Level 2 interrupts which, in turn, always 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 33. Reset, Watchdog Timer interrupt (if enabled), Primary Oscillator Fail Trap, Watchdog Timer Oscillator Fail Trap, and Illegal Instruction Trap 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 until the next interrupt occurs. Writing a 0 to the corresponding bit in the Interrupt Request register likewise clears the interrupt request.

**! Caution:** *The following coding style that clears bits in the Interrupt Request registers is not recommended. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost.*

### Poor coding style that can result in lost interrupt requests:

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

**Table 46. Interrupt Edge Select Register (IRQES)**

BITS	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
ADDR	FCDH							

IES<sub>x</sub>—Interrupt Edge Select *x*

0 = An interrupt request is generated on the falling edge of the PA<sub>x</sub> input or PD<sub>x</sub>

1 = An interrupt request is generated on the rising edge of the PA<sub>x</sub> input PD<sub>x</sub>

where *x* indicates the specific GPIO port pin number (0 through 7)

## Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) register (Table 47) determines the source of the PAD<sub>x</sub>S 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 47. Shared Interrupt Select Register (IRQSS)**

BITS	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
ADDR	FCEH							

PA6CS—PA6/Comparator Selection

0 = PA6 is used for the interrupt for PA6CS interrupt request

1 = The Comparator is used for the interrupt for PA6CS interrupt request

Reserved—Must be 0

## Interrupt Control Register

The Interrupt Control (IRQCTL) register (Table 48) contains the master enable bit for all interrupts.

# Timers

Z8 Encore! XP<sup>®</sup> F0823 Series products contain up to two 16-bit reloadable timers that are used for timing, event counting, or generation of PWM signals. The timers' features include:

- 16-bit reload counter.
- Programmable prescaler with prescale values 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.

In addition to the timers described in this chapter, the baud rate generator of the UART (if unused) also provides basic timing functionality. For information on using the baud rate generator as an additional timer, see Universal Asynchronous Receiver/Transmitter on page 93.

## Architecture

Figure 9 displays the architecture of the timers.

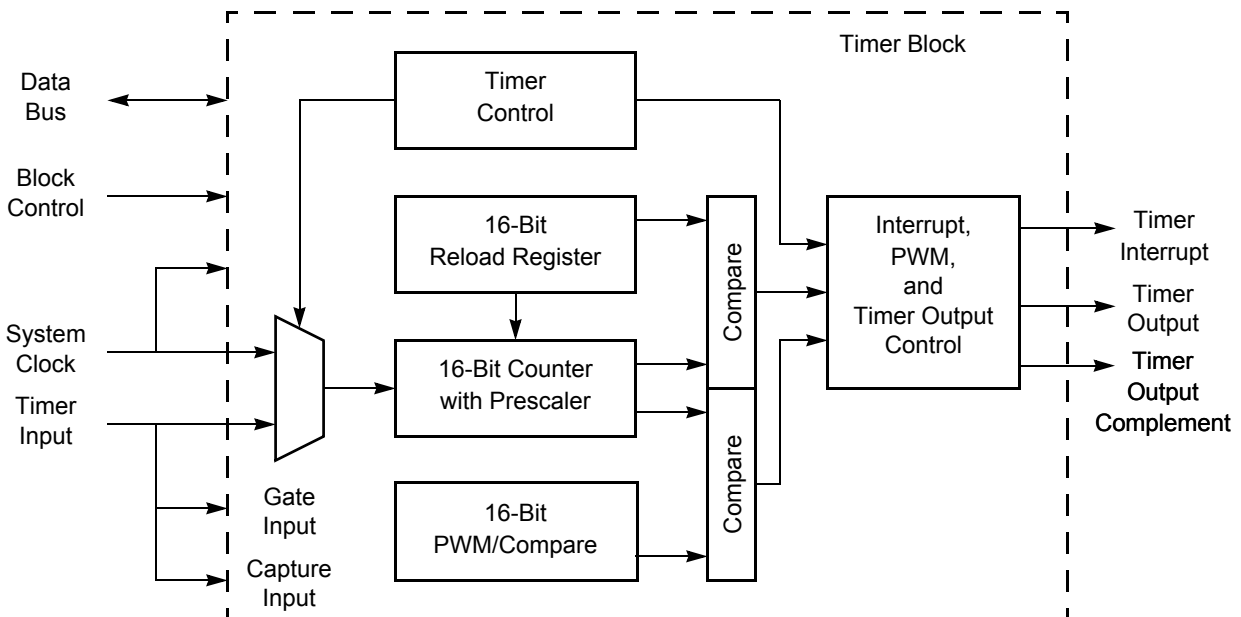


Figure 9. Timer Block Diagram

of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the timer continues counting. The INPCAP bit in TxCTL1 register is set to indicate the timer interrupt is 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 TxCTL1 register clears indicating the timer interrupt is not because of an input capture event.

Follow the steps below for configuring 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 the software to determine if interrupts were generated by either a Capture or a Reload event. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt was 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 from timer start to Capture event can be 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 the 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 if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is

### Software Compensation Procedure

The value read from the ADC high and low byte registers are uncompensated. The user mode software must apply gain and offset correction to this uncompensated value for maximum accuracy. The following formula yields the compensated value:

$$\text{ADC}_{\text{comp}} = (\text{ADC}_{\text{uncomp}} - \text{OFFCAL}) + ((\text{ADC}_{\text{uncomp}} - \text{OFFCAL}) * \text{GAINCAL}) / 2^{16}$$

where GAINCAL is the gain calibration byte, OFFCAL is the offset calibration byte and  $\text{ADC}_{\text{uncomp}}$  is the uncompensated value read from the ADC. The OFFCAL value is in two's complement format, as are the compensated and uncompensated ADC values.

► **Note:** *The offset compensation is performed first, followed by the gain compensation. One bit of resolution is lost because of rounding on both the offset and gain computations. As a result the ADC registers read back 13 bits: 1 sign bit, two calibration bits lost to rounding and 10 data bits. Also note that in the second term, the multiplication must be performed before the division by  $2^{16}$ . Otherwise, the second term evaluates to zero incorrectly.*

! **Caution:** *Although the ADC can be used without the gain and offset compensation, it does exhibit non-unity gain. Designing the ADC with sub-unity gain reduces noise across the ADC range but requires the ADC results to be scaled by a factor of 8/7.*

## ADC Control Register Definitions

The following sections define the ADC control registers.

### ADC Control Register 0

The ADC Control register selects the analog input channel and initiates the analog-to-digital conversion.

**Table 72. ADC Control Register 0 (ADCCTL0)**

BITS	7	6	5	4	3	2	1	0
FIELD	CEN	REFSELL	REFEXT	CONT	ANAIN[3:0]			
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
ADDR	F70H							

CEN—Conversion Enable

0 = Conversion is complete. Writing a 0 produces no effect. The ADC automatically clears this bit to 0 when a conversion is complete.

1 = Begin conversion. Writing a 1 to this bit starts a conversion. If a conversion is already in progress, the conversion restarts. This bit remains 1 until the conversion is complete.

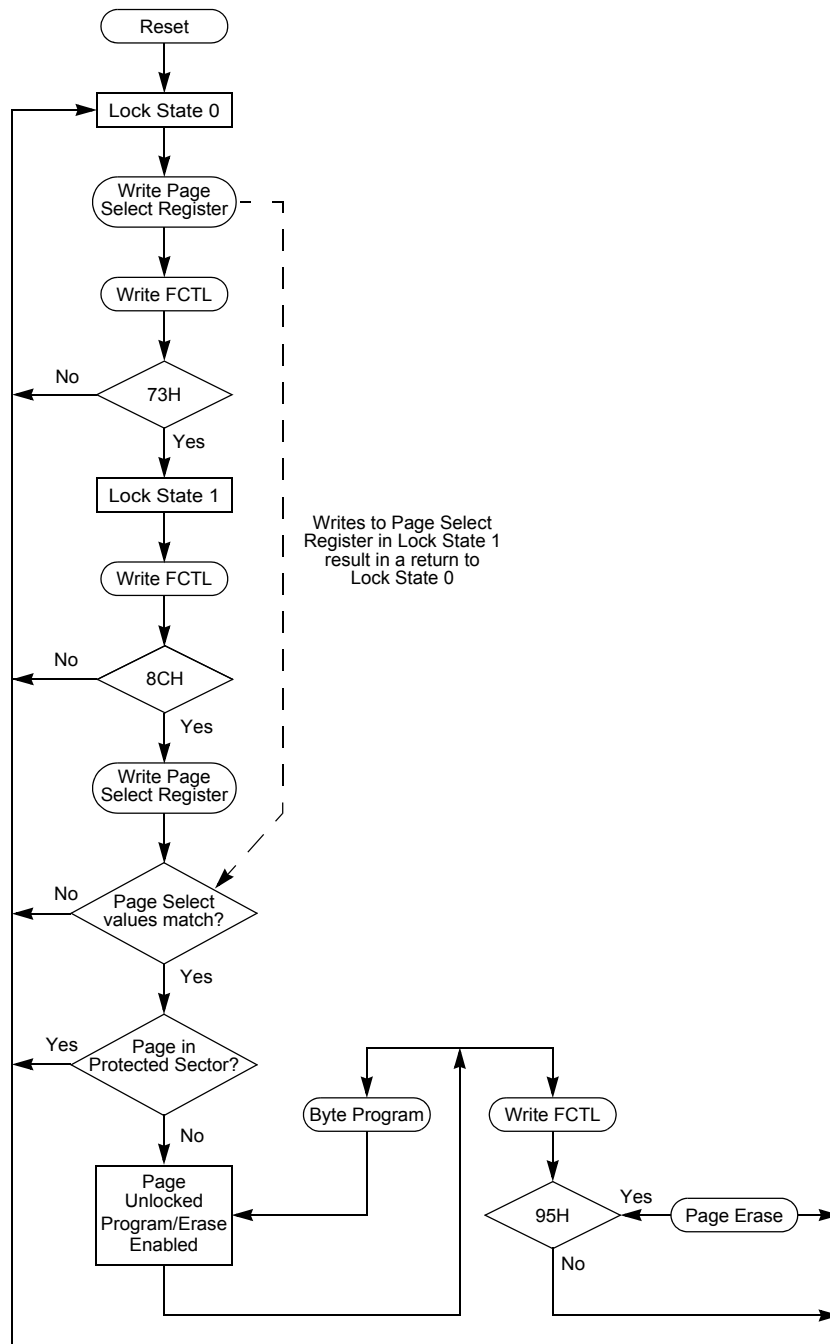


Figure 21. Flash Controller Operation Flowchart

## Flash Control Register Definitions

### Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FCTL) register before programming or erasing the Flash memory. Writing the sequence 73H 8CH, sequentially, to the Flash Control register unlocks the Flash Controller. When the Flash Controller is unlocked, the Flash memory can be 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 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 shares its Register File address with the read-only Flash Status Register.

**Table 79. Flash Control Register (FCTL)**

BITS	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
ADDR	FF8H							

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 indicates the current state of the Flash Controller. This register can be read at any time. The read-only Flash Status Register shares its Register File address with the write-only Flash Control Register.

**Table 80. Flash Status Register (FSTAT)**

BITS	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
ADDR	FF8H							



# On-Chip Debugger

Z8 Encore! XP® F0823 Series devices contain an integrated On-Chip Debugger (OCD) that provides advanced debugging features that include:

- Single pin interface
- Reading and writing of the register file
- Reading and writing of program and data memory
- Setting of breakpoints and watchpoints
- Executing eZ8 CPU instructions
- Debug pin sharing with general-purpose input-output function to maximize the pins available

## Architecture

The on-chip debugger consists of four primary functional blocks: transmitter, receiver, auto-baud detector/generator, and debug controller. Figure 22 displays the architecture of the OCD.

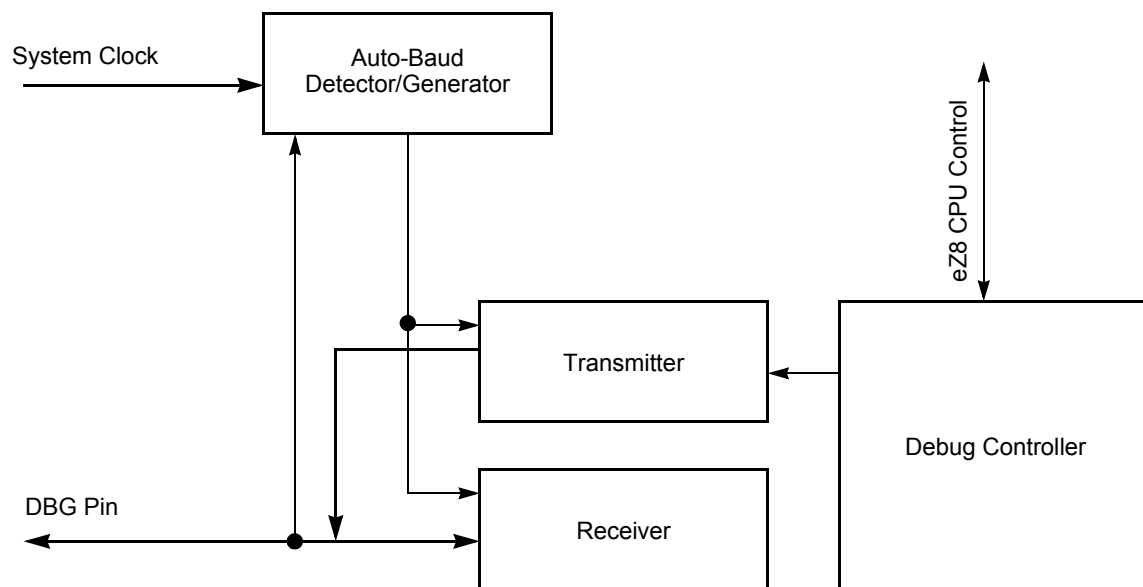


Figure 22. On-Chip Debugger Block Diagram

**Table 115. eZ8 CPU Instruction Summary (Continued)**

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	lr	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	–	–	–	–	–	–	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
		lr		F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X	r		F6	–	–	–	–	–	–	3	3
		lr		F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X	r		F6	–	–	–	–	–	–	3	3
		lr		F7							3	4
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
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

**Table 115. eZ8 CPU Instruction Summary (Continued)**

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
LDC dst, src	$\text{dst} \leftarrow \text{src}$	r	lrr	C2	–	–	–	–	–	–	2	5
		lr	lrr	C5							2	9
		lrr	r	D2							2	5
LDCI dst, src	$\text{dst} \leftarrow \text{src}$ $\text{r} \leftarrow \text{r} + 1$ $\text{rr} \leftarrow \text{rr} + 1$	lr	lrr	C3	–	–	–	–	–	–	2	9
		lrr	lr	D3							2	9
LDE dst, src	$\text{dst} \leftarrow \text{src}$	r	lrr	82	–	–	–	–	–	–	2	5
		lrr	r	92							2	5
LDEI dst, src	$\text{dst} \leftarrow \text{src}$ $\text{r} \leftarrow \text{r} + 1$ $\text{rr} \leftarrow \text{rr} + 1$	lr	lrr	83	–	–	–	–	–	–	2	9
		lrr	lr	93							2	9
LDWX dst, src	$\text{dst} \leftarrow \text{src}$	ER	ER	1FE8	–	–	–	–	–	–	5	4
LDX dst, src	$\text{dst} \leftarrow \text{src}$	r	ER	84	–	–	–	–	–	–	3	2
		lr	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	lr	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)	$\text{dst} \leftarrow \text{src} + \text{X}$	r	X(r)	98	–	–	–	–	–	–	3	3
		rr	X(rr)	99							3	5
MULT dst	$\text{dst}[15:0] \leftarrow \text{dst}[15:8] * \text{dst}[7:0]$	RR		F4	–	–	–	–	–	–	2	8
NOP	No operation			0F	–	–	–	–	–	–	1	2
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

Figure 36 displays the 8-pin Quad Flat No-Lead package (QFN)/MLF-S available for the Z8 Encore! XP F0823 Series devices. This package has a footprint identical to that of the 8-pin SOIC, but with a lower profile.

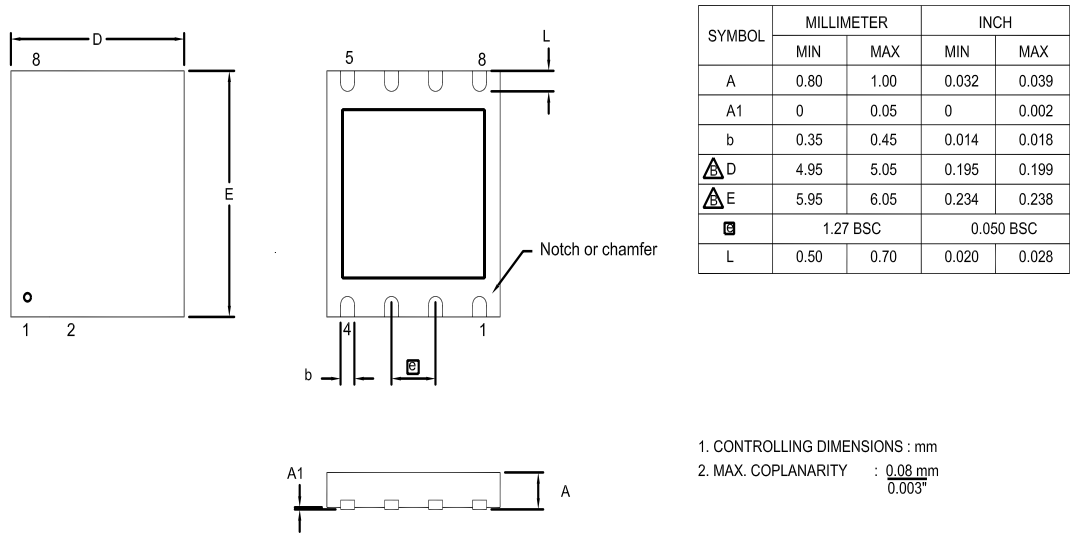


Figure 36. 8-Pin Quad Flat No-Lead Package (QFN)/MLF-S

Figure 37 displays the 20-pin Plastic Dual Inline Package (PDIP) available for Z8 Encore! XP F0823 Series devices.

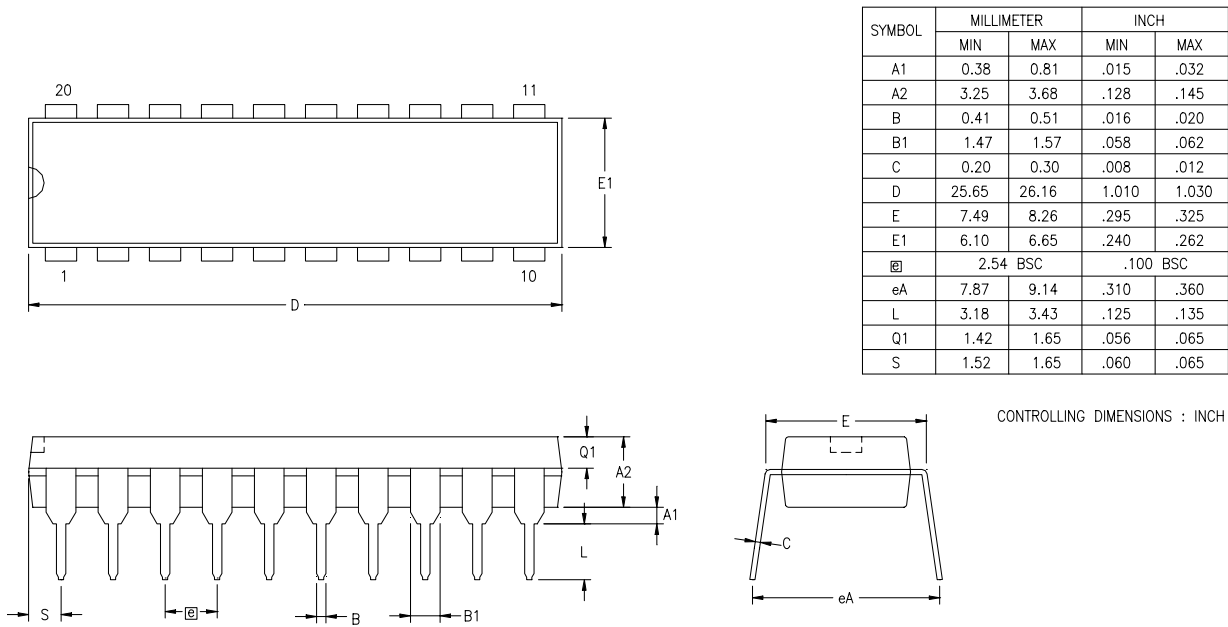


Figure 37. 20-Pin Plastic Dual Inline Package (PDIP)

## **T**

- TCM 176
- TCMX 176
- test complement under mask 176
- test complement under mask - extended addressing 176
- test under mask 176
- test under mask - extended addressing 176
- timer signals 9
- timers 67
  - architecture 67
  - block diagram 67
  - CAPTURE mode 74, 75, 84, 85
  - CAPTURE/COMPARE mode 78, 85
  - COMPARE mode 76, 84
  - CONTINUOUS mode 69, 84
  - COUNTER mode 70, 71
  - COUNTER modes 84
  - GATED mode 77, 84
  - ONE-SHOT mode 68, 84
  - operating mode 68
  - PWM mode 72, 73, 84, 85
  - reading the timer count values 79
  - reload high and low byte registers 80
  - timer control register definitions 80
  - timer output signal operation 79
- timers 0-3
  - control registers 82, 83
  - high and low byte registers 80, 81
- TM 176
- TMX 176
- tools, hardware and software 226
- transmit
  - IrDA data 114
- transmitting UART data-polled method 95
- transmitting UART dat-interrupt-driven method 96
- TRAP 178

## **U**

- UART 4
  - architecture 93
  - baud rate generator 103

- control register definitions 104
- controller signals 9
- data format 94
- interrupts 101
- MULTIPROCESSOR mode 99
- receiving data using interrupt-driven method 98
- receiving data using the polled method 97
- transmitting data using the interrupt-driven method 96
- transmitting data using the polled method 95
- x baud rate high and low registers 110
- x control 0 and control 1 registers 107
- x status 0 and status 1 registers 105, 106
- UxBRH register 110
- UxBRL register 110
- UxCTL0 register 107, 110
- UxCTL1 register 108
- UxRXD register 105
- UxSTAT0 register 105
- UxSTAT1 register 106
- UxTXD register 104

## **V**

- vector 173
- Voltage Brownout reset (VBR) 24

## **W**

- Watchdog Timer
  - approximate time-out delay 87
  - CNTL 24
  - control register 89, 127, 167
  - electrical characteristics and timing 200, 202
  - interrupt in normal operation 88
  - interrupt in STOP mode 88
  - refresh 88, 177
  - reload unlock sequence 89
  - reload upper, high and low registers 90
  - reset 25
  - reset in normal operation 89