



Welcome to [E-XFL.COM](https://www.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	16
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f0223ph005eg">https://www.e-xfl.com/product-detail/zilog/z8f0223ph005eg</a>

MULTIPROCESSOR (9-Bit) Mode .....	103
External Driver Enable .....	105
UART Interrupts .....	105
UART Baud Rate Generator .....	108
UART Control Register Definitions .....	108
UART Transmit Data Register .....	109
UART Receive Data Register .....	109
UART Status 0 Register .....	110
UART Status 1 Register .....	111
UART Control 0 and Control 1 Registers .....	112
UART Address Compare Register .....	115
UART Baud Rate High and Low Byte Registers .....	115
Infrared Encoder/Decoder .....	117
Architecture .....	117
Operation .....	117
Transmitting IrDA Data .....	118
Receiving IrDA Data .....	119
Infrared Encoder/Decoder Control Register Definitions .....	120
Analog-to-Digital Converter .....	121
Architecture .....	121
Operation .....	122
Automatic Powerdown .....	123
Single-Shot Conversion .....	123
Continuous Conversion .....	124
Interrupts .....	125
Calibration and Compensation .....	125
ADC Control Register Definitions .....	126
ADC Control Register 0 .....	126
ADC Control/Status Register 1 .....	129
ADC Data High Byte Register .....	130
ADC Data Low Bits Register .....	131
Comparator .....	132
Operation .....	132
Comparator Control Register Definition .....	133
Flash Memory .....	134
Flash Information Area .....	135
Operation .....	135
Flash Operation Timing Using the Flash Frequency Registers .....	137
Flash Code Protection Against External Access .....	137

## ***Reset and Stop Mode Recovery***

The Reset Controller within the Z8 Encore! XP F0823 Series controls Reset and Stop Mode Recovery operation and provides indication of low supply voltage conditions. In typical operation, the following events cause a Reset:

- Power-On Reset (POR)
- Voltage Brown-Out (VBO)
- 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:

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

The VBO circuitry on the device performs the following function:

- Generates the VBO reset when the supply voltage drops below a minimum safe level

### **Reset Types**

F0823 Series MCUs provide several different types of Reset operations. Stop Mode Recovery is considered a form of Reset. Table 9 lists the types of Reset and their operating characteristics. The duration of a System Reset is longer if the external crystal oscillator is enabled by the Flash option bits; this configuration allows additional time for oscillator startup.

## 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.
	Watchdog Timer time-out when configured for Reset.	None.
	RESET pin assertion.	All reset pulses less than three system clocks in width are ignored.
	OCD initiated Reset (OCDCTL[0] set to 1).	System Reset, except the OCD is unaffected by the reset.
STOP Mode	Power-On Reset/Voltage Brown-Out.	Reset delay begins after supply voltage exceeds POR level.
	RESET pin assertion.	All reset pulses less than the specified analog delay are ignored. See the <a href="#">Electrical Characteristics chapter on page 196</a> .
	DBG pin driven Low.	None.

## Power-On Reset

Each device in the Z8 Encore! XP F0823 Series contains an internal POR circuit. The POR circuit monitors the supply voltage and holds the device in the Reset state until the 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, this time-out is longer.

After the Z8 Encore! XP F0823 Series device exits the POR state, the eZ8 CPU fetches the Reset vector. Following the POR, the POR status bit in Watchdog Timer Control (WDTCTL) Register is set to 1.

Figure 5 displays POR operation. For the POR threshold voltage ( $V_{POR}$ ), see the [Electrical Characteristics](#) chapter on page 196.



## 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.

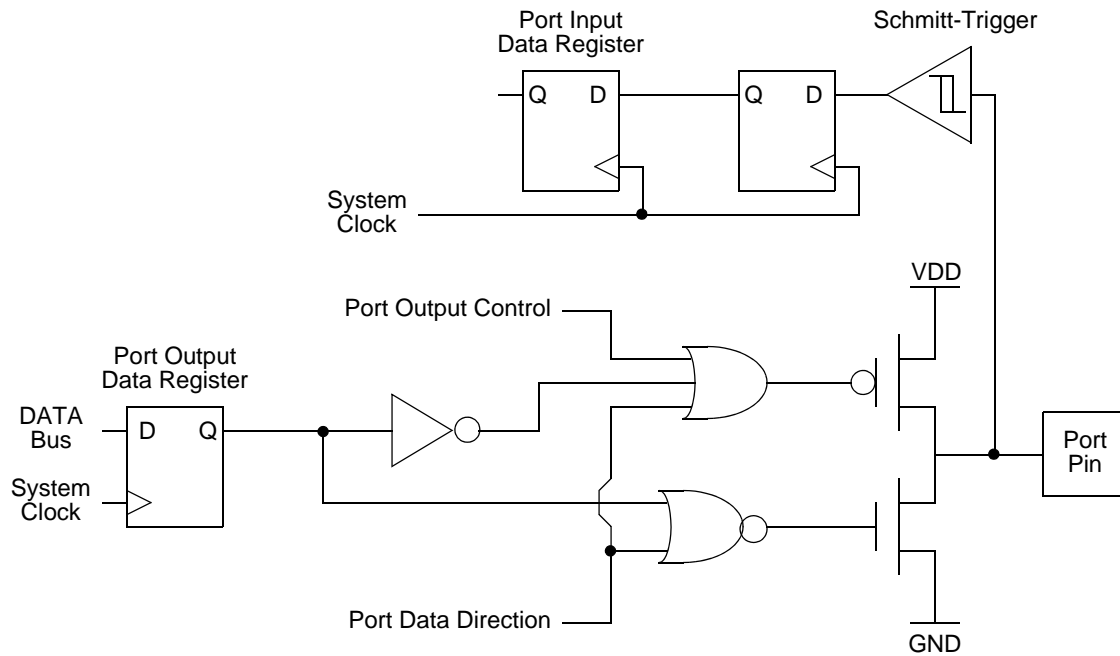


Figure 7. GPIO Port Pin Block Diagram

## 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 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. Tables 16 and 17 list the alternate functions possible with each port pin for 8-pin and non-8-pin parts, respectively. The alternate function associated at a pin is defined through Alternate Function Sets 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, those pins function as input and output for the crystal oscillator.

## LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin (Table 33). These two bits select between four programmable drive levels. Each pin is individually programmable.

**Table 33. 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]	<b>LED Level High Bit</b>
LEDLVLH	{LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA. 01 = 7mA. 10 = 13mA. 11 = 20mA.

**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 a 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
```

## Watchdog Timer Interrupt Assertion

The Watchdog Timer interrupt behavior is different from interrupts generated by other sources. The Watchdog Timer continues to assert an interrupt as long as the timeout condition continues. As it operates on a different (and usually slower) clock domain than the rest of the device, the Watchdog Timer continues to assert this interrupt for many system clocks until the counter rolls over.

Also, 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. If it is appropriate to have the Timer Output 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 initiating 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 computed via 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. Also, 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:

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 can be 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 if a 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. When 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). Also, 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, 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 deassertion and reload events. If appropriate, configure the timer interrupt to be generated only at the input deassertion event or the reload event by setting TICONFIG field 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 the capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to 0001H, and counting resumes. The INPCAP bit in TxCTL1 Register is set to indicate 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 TxCTL1 Register is cleared to indicate the timer interrupt is not because of an input capture event.

Observe the following steps to configure a timer for CAPTURE/COMPARE Mode and initiating 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, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt are 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.
5. Configure the associated GPIO port pin for the Timer Input alternate function.
6. Write to the Timer Control Register to enable the timer.

$$\left(\frac{1}{\text{Baud Rate (Hz)}}\right) \leq \text{DE to Start Bit Setup Time (s)} \leq \left(\frac{2}{\text{Baud Rate (Hz)}}\right)$$

### Transmitter Interrupts

The transmitter generates a single interrupt when the Transmit Data Register Empty bit (TDRE) is set to 1. This indicates that the transmitter is ready to accept new data for transmission. The TDRE interrupt occurs after the Transmit shift register has shifted the first bit of data out. The Transmit Data Register can now be written with the next character to send. This action provides 7 bit periods of latency to load the Transmit Data Register before the Transmit shift register completes shifting the current character. Writing to the UART Transmit Data Register clears the TDRE bit to 0.

### Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte is received and is available in the UART Receive Data Register. This interrupt can be disabled independently of the other receiver interrupt sources. The received data interrupt occurs after the receive character has been received and placed in the Receive Data Register. To avoid an overrun error, software must respond to this received data available condition before the next character is completely received.

---

► **Note:** In MULTIPROCESSOR Mode (MPEN = 1), the receive data interrupts are dependent on the multiprocessor configuration and the most recent address byte.

---

- A break is received
- An overrun is detected
- A data framing error is detected

### UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data Register. The Break Detect and Overrun status bits are not displayed until after the valid data has been read.

After the valid data has been read, the UART Status 0 Register is updated to indicate the overrun condition (and Break Detect, if applicable). The RDA bit is set to 1 to indicate that the Receive Data Register contains a data byte. However, because the overrun error

# ***Analog-to-Digital Converter***

The Analog-to-Digital Converter (ADC) converts an analog input signal to its digital representation. The features of this sigma-delta ADC include:

- 10-bit resolution
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports
- Interrupt upon conversion complete
- Bandgap generated internal voltage reference generator with two selectable levels
- Factory offset and gain calibration

## **Architecture**

Figure 19 displays the major functional blocks of the ADC. An analog multiplexer network selects the ADC input from the available analog pins, ANA0 through ANA7.



These serial numbers are stored in the Flash information page (for more details, see the [Reading the Flash Information Page](#) section on page 148 and the [Serialization Data](#) section on page 154) and are unaffected by mass erasure of the device's Flash memory.

### Randomized Lot Identification Bits

As an optional feature, Zilog is able to provide a factory-programmed random lot identifier. With this feature, all devices in a given production lot are programmed with the same random number. This random number is uniquely regenerated for each successive production lot and is not likely to be repeated.

The randomized lot identifier is a 32-byte binary value, stored in the flash information page (for more details, see the [Reading the Flash Information Page](#) section on page 148 and the [Randomized Lot Identifier](#) section on page 154) and is unaffected by mass erasure of the device's flash memory.

## Reading the Flash Information Page

The following code example shows how to read data from the Flash Information Area.

```
; get value at info address 60 (FE60h)
ldx FPS, #80 ; enable access to flash info page
ld R0, #FE
ld R1, #60
ldc R2, @RR0 ; R2 now contains the calibration value
```

## 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 (TRMADR) Register contains the target address for an access to the trim option bits.

**Table 87. 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							

Bit	Description (Continued)
[3] VBO_AO	<b>Voltage Brown-Out Protection Always ON</b> 0 = Voltage Brown-Out Protection can be disabled in STOP Mode to reduce total power consumption. For the block to be disabled, the power control register bit must also be written (see the <a href="#">Power Control Register 0</a> section on page 31). 1 = Voltage Brown-Out Protection is always enabled including during STOP Mode. This setting is the default for unprogrammed (erased) Flash.
[2] FRP	<b>Flash Read Protect</b> 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 setting is the default for unprogrammed (erased) Flash.
[1]	<b>Reserved</b> This bit is reserved and must be programmed to 1.
[0] FWP	<b>Flash Write Protect</b> This Option Bit provides Flash Program Memory protection: 0 = Programming and erasure disabled for all of 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 of Flash program memory.

**Table 90. Flash Options Bits at Program Memory Address 0001H**

Bit	7	6	5	4	3	2	1	0
Field	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.								

---

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

---

### OSC Control Register Unlocking/Locking

To write to 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 can be 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 is 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

Z8 Encore! XP F0823 Series devices can generate non-maskable 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](#) section on page 91.

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below 1 kHz  $\pm 50\%$ . If an external signal is selected as the system oscillator, it is possible that a very slow but non-failing clock can generate a failure condition. Under these

**Table 110. Arithmetic Instructions (Continued)**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
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

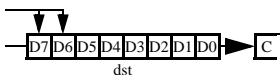

**Table 111. Bit Manipulation Instructions**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
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 112. Block Transfer Instructions**

<b>Mnemonic</b>	<b>Operands</b>	<b>Instruction</b>
LDCI	dst, src	Load Constant to/from Program Memory and Auto-Increment Addresses
LDEI	dst, src	Load External Data to/from Data Memory and Auto-Increment Addresses

Table 118. 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			
SBC dst, src	$dst \leftarrow dst - src - C$	r	r	32	*	*	*	*	1	*	2	3	
		r	lr	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	$dst \leftarrow dst - src - C$	ER	ER	38	*	*	*	*	1	*	4	3	
		ER	IM	39							4	3	
SCF	$C \leftarrow 1$			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	$RP \leftarrow src$		IM	01	–	–	–	–	–	–	2	2	
STOP	STOP Mode			6F	–	–	–	–	–	–	1	2	
SUB dst, src	$dst \leftarrow dst - src$	r	r	22	*	*	*	*	1	*	2	3	
		r	lr	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	$dst \leftarrow dst - src$	ER	ER	28	*	*	*	*	1	*	4	3	
		ER	IM	29							4	3	
SWAP dst	$dst[7:4] \leftrightarrow dst[3:0]$	R		F0	X	*	*	X	–	–	2	2	
		IR		F1							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.

## DC Characteristics

Table 121 lists the DC characteristics of the Z8 Encore! XP F0823 Series products. All voltages are referenced to  $V_{SS}$ , the primary system ground.

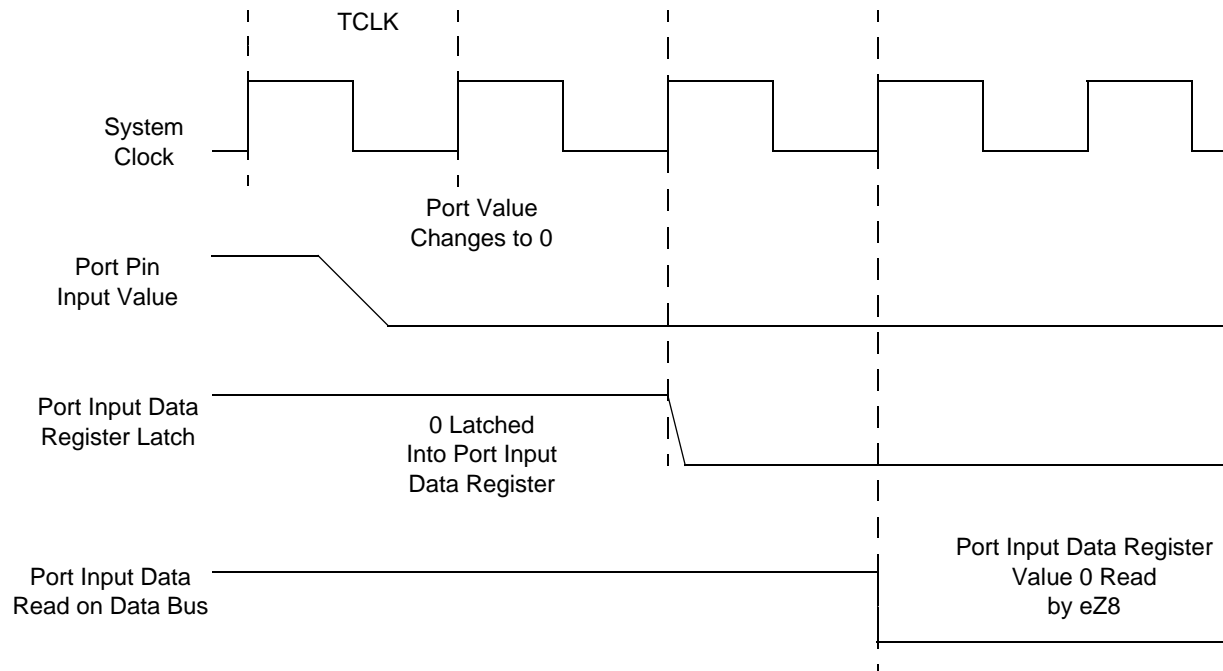
**Table 121. DC Characteristics**

Symbol	Parameter	$T_A = -40^{\circ}\text{C to } +105^{\circ}\text{C}$ (unless otherwise specified)			Units	Conditions
		Minimum	Typical	Maximum		
$V_{DD}$	Supply Voltage	2.7	–	3.6	V	
$V_{IL1}$	Low Level Input Voltage	–0.3	–	$0.3 \cdot V_{DD}$	V	
$V_{IH1}$	High Level Input Voltage	$0.7 \cdot V_{DD}$	–	5.5	V	For all input pins without analog or oscillator function. For all signal pins on the 8-pin devices. Programmable pull-ups must also be disabled.
$V_{IH2}$	High Level Input Voltage	$0.7 \cdot V_{DD}$	–	$V_{DD} + 0.3$	V	For those pins with analog or oscillator function (20-/28-pin devices only), or when programmable pull-ups are enabled.
$V_{OL1}$	Low Level Output Voltage	–	–	0.4	V	$I_{OL} = 2\text{mA}$ ; $V_{DD} = 3.0\text{V}$ High Output Drive disabled.
$V_{OH1}$	High Level Output Voltage	2.4	–	–	V	$I_{OH} = -2\text{mA}$ ; $V_{DD} = 3.0\text{V}$ High Output Drive disabled.
$V_{OL2}$	Low Level Output Voltage	–	–	0.6	V	$I_{OL} = 20\text{mA}$ ; $V_{DD} = 3.3\text{V}$ High Output Drive enabled.
$V_{OH2}$	High Level Output Voltage	2.4	–	–	V	$I_{OH} = -20\text{mA}$ ; $V_{DD} = 3.3\text{V}$ High Output Drive enabled.
$I_{IH}$	Input Leakage Current	–	$\pm 0.002$	$\pm 5$	$\mu\text{A}$	$V_{IN} = V_{DD}$ $V_{DD} = 3.3\text{V}$
$I_{IL}$	Input Leakage Current	–	$\pm 0.007$	$\pm 5$	$\mu\text{A}$	$V_{IN} = V_{SS}$ $V_{DD} = 3.3\text{V}$
$I_{TL}$	Tristate Leakage Current	–	–	$\pm 5$	$\mu\text{A}$	

**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.

available to the eZ8 CPU on the second rising clock edge following the change of the port value.



**Figure 29. Port Input Sample Timing**

**Table 130. GPIO Port Input Timing**

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
T <sub>S_PORT</sub>	Port Input Transition to X <sub>IN</sub> Rise Setup Time (Not pictured)	5	–
T <sub>H_PORT</sub>	X <sub>IN</sub> Rise to Port Input Transition Hold Time (Not pictured)	0	–
T <sub>SMR</sub>	GPIO Port Pin Pulse Width to ensure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources)	1 μs	

# Packaging

Zilog's F0823 Series of MCUs includes the Z8F0113, Z8F0123, Z8F0213, Z8F0223, Z8F0413, Z8F0423, Z8F0813 and Z8F0823 devices, which are available in the following packages:

- 8-pin Plastic Dual Inline Package (PDIP)
- 8-Pin Quad Flat No-Lead Package (QFN)/MLF-S<sup>1</sup>
- 20-pin Plastic Dual-Inline Package (PDIP)
- 20-pin Small Outline Integrated Circuit Package (SOIC)
- 20-pin Small Shrink Outline Package (SSOP)
- 28-pin Plastic Dual-Inline Package (PDIP)
- 28-pin Small Outline Integrated Circuit Package (SOIC)
- 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.

---

1. The footprint of the QFN)/MLF-S package is identical to that of the 8-pin SOIC package, but with a lower profile.



**Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)**

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Description
<b>Z8 Encore! XP F0823 Series with 1 KB Flash, 10-Bit Analog-to-Digital Converter</b>								
<b>Standard Temperature: 0°C to 70°C</b>								
Z8F0123PB005SG	1 KB	256 B	6	12	2	4	1	PDIP 8-pin package
Z8F0123QB005SG	1 KB	256 B	6	12	2	4	1	QFN 8-pin package
Z8F0123SB005SG	1 KB	256 B	6	12	2	4	1	SOIC 8-pin package
Z8F0123SH005SG	1 KB	256 B	16	18	2	7	1	SOIC 20-pin package
Z8F0123HH005SG	1 KB	256 B	16	18	2	7	1	SSOP 20-pin package
Z8F0123PH005SG	1 KB	256 B	16	18	2	7	1	PDIP 20-pin package
Z8F0123SJ005SG	1 KB	256 B	22	18	2	8	1	SOIC 28-pin package
Z8F0123HJ005SG	1 KB	256 B	22	18	2	8	1	SSOP 28-pin package
Z8F0123PJ005SG	1 KB	256 B	22	18	2	8	1	PDIP 28-pin package
<b>Extended Temperature: -40°C to 105°C</b>								
Z8F0123PB005EG	1 KB	256 B	6	12	2	4	1	PDIP 8-pin package
Z8F0123QB005EG	1 KB	256 B	6	12	2	4	1	QFN 8-pin package
Z8F0123SB005EG	1 KB	256 B	6	12	2	4	1	SOIC 8-pin package
Z8F0123SH005EG	1 KB	256 B	16	18	2	7	1	SOIC 20-pin package
Z8F0123HH005EG	1 KB	256 B	16	18	2	7	1	SSOP 20-pin package
Z8F0123PH005EG	1 KB	256 B	16	18	2	7	1	PDIP 20-pin package
Z8F0123SJ005EG	1 KB	256 B	22	18	2	8	1	SOIC 28-pin package
Z8F0123HJ005EG	1 KB	256 B	22	18	2	8	1	SSOP 28-pin package
Z8F0123PJ005EG	1 KB	256 B	22	18	2	8	1	PDIP 28-pin package