



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	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K 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/z8f0423ph005eg">https://www.e-xfl.com/product-detail/zilog/z8f0423ph005eg</a>

Table 3. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
COUT	O	Comparator Output. This is the output of the comparator.
<b>Analog</b>		
ANA[7:0]	I	Analog port. These signals are used as inputs to the ADC. The ANA0, ANA1, and ANA2 pins can also access the inputs and output of the integrated transimpedance amplifier.
VREF	I/O	Analog-to-Digital Converter reference voltage input.
<b>Clock Input</b>		
CLKIN	I	Clock Input Signal. This pin can be used to input a TTL-level signal to be used as the system clock.
<b>LED Drivers</b>		
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.
<b>On-Chip Debugger</b>		
DBG	I/O	Debug. This signal is the control and data input and output to and from the OCD. <b>Caution:</b> The DBG pin is open-drain and requires an external pull-up resistor to ensure proper operation.
<b>Reset</b>		
RESET	I/O	RESET. Generates a reset when asserted (driven Low). Also serves as a reset indicator; the Z8 Encore! XP forces this pin Low when in reset. This pin is open-drain and features an enabled internal pull-up resistor.
<b>Power Supply</b>		
V <sub>DD</sub>	I	Digital Power Supply.
AV <sub>DD</sub> <sup>2</sup>	I	Analog Power Supply.
V <sub>SS</sub>	I	Digital Ground.
AV <sub>SS</sub>	I	Analog Ground.

Notes:

1. PB6 and PB7 are only available in 28-pin packages without ADC. In 28-pin packages with ADC, they are replaced by AV<sub>DD</sub> and AV<sub>SS</sub>.
2. The AV<sub>DD</sub> and AV<sub>SS</sub> signals are available only in 28-pin packages with ADC. They are replaced by PB6 and PB7 on 28-pin packages without ADC.

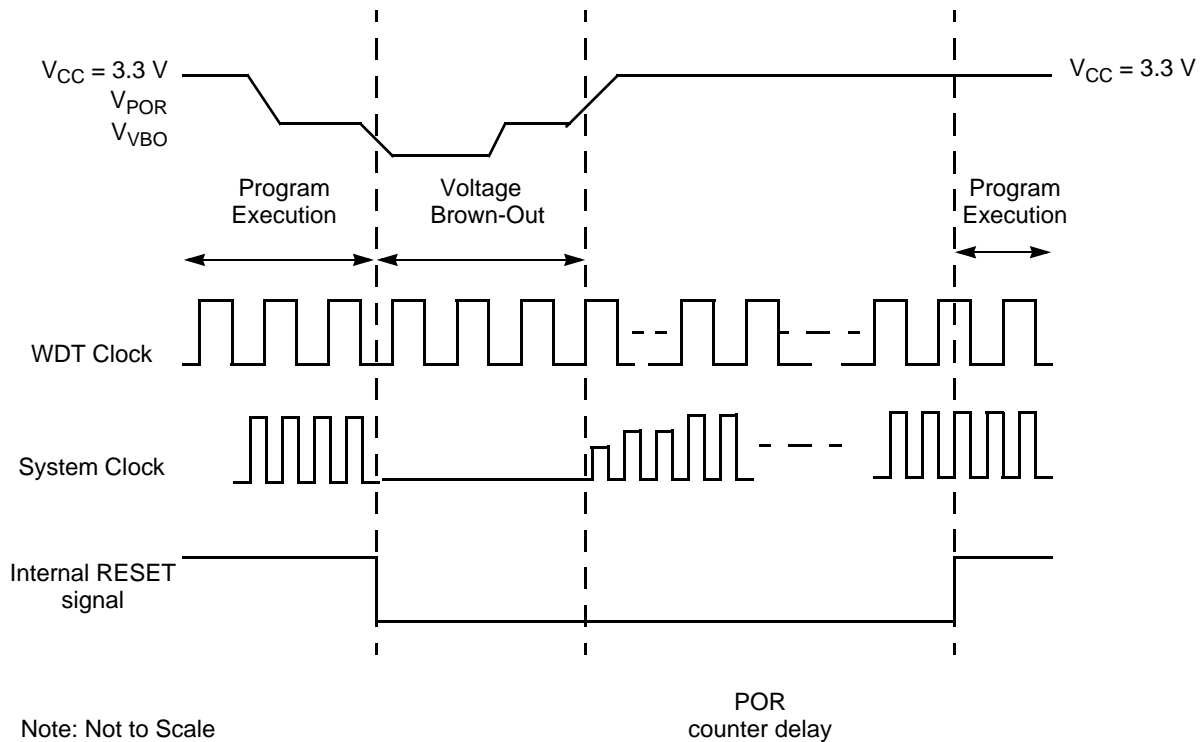


Figure 6. Voltage Brown-Out Reset Operation

The POR level is greater than the VBO level by the specified hysteresis value. This ensures that the device undergoes a POR after recovering from a VBO condition.

## Watchdog Timer Reset

If the device is in NORMAL or STOP Mode, the Watchdog Timer can initiate 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 Watchdog Timer to cause an interrupt, not a System Reset, at time-out.

The WDT status bit in the WDT Control Register is set to signify that the reset was initiated by the Watchdog Timer.

## 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

**Table 35. Trap and Interrupt Vectors in Order of Priority**

<b>Priority</b>	<b>Program Memory Vector Address</b>	<b>Interrupt or Trap Source</b>
Highest	0002H	Reset (not an interrupt)
	0004H	Watchdog Timer (see the <a href="#">Watchdog Timer</a> section on page 91)
	003AH	Primary Oscillator Fail Trap (not an interrupt)
	003CH	Watchdog Timer Oscillator Fail Trap (not an interrupt)
	0006H	Illegal Instruction Trap (not an interrupt)
	0008H	Reserved
	000AH	Timer 1
	000CH	Timer 0
	000EH	UART 0 receiver
	0010H	UART 0 transmitter
	0012H	Reserved
	0014H	Reserved
	0016H	ADC
	0018H	Port A Pin 7, selectable rising or falling input edge
	001AH	Port A Pin 6, selectable rising or falling input edge or Comparator Output
	001CH	Port A Pin 5, selectable rising or falling input edge
	001EH	Port A Pin 4, selectable rising or falling input edge
	0020H	Port A Pin 3 or Port D Pin 3, selectable rising or falling input edge
	0022H	Port A Pin 2 or Port D Pin 2, selectable rising or falling input edge
	0024H	Port A Pin 1, selectable rising or falling input edge
	0026H	Port A Pin 0, selectable rising or falling input edge
	0028H	Reserved
	002AH	Reserved
	002CH	Reserved
	002EH	Reserved
	0030H	Port C Pin 3, both input edges
	0032H	Port C Pin 2, both input edges
	0034H	Port C Pin 1, both input edges
	0036H	Port C Pin 0, both input edges
Lowest	0038H	Reserved



**! Caution:** To avoid retriggerings of the Watchdog Timer interrupt after exiting the associated interrupt service routine, Zilog recommends that the service routine continues to read from the RSTSTAT register until the WDT bit is cleared as shown in the following example.

```
CLEARWDT:
    LDX r0, RSTSTAT ; read reset status register to clear wdt bit
    BTJNZ 5, r0, CLEARWDT ; loop until bit is cleared
```

## Interrupt Control Register Definitions

For all interrupts other than the Watchdog Timer interrupt, the Primary Oscillator Fail Trap, and the Watchdog Timer Oscillator Fail Trap, the interrupt control registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

### Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) register (Table 36) stores the interrupt requests for both vectored and polled interrupts. When a request is presented 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 36. Interrupt Request 0 Register (IRQ0)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1I	T0I	U0RXI	U0TXI	Reserved		ADCI
RESET	0	0	0	0	0	0		0
R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Address	FC0H							

Bit	Description
[7]	<b>Reserved</b> This bit is reserved and must be programmed to 0.
[6] T1I	<b>Timer 1 Interrupt Request</b> 0 = No interrupt request is pending for Timer 1. 1 = An interrupt request from Timer 1 is awaiting service.
[5] T0I	<b>Timer 0 Interrupt Request</b> 0 = No interrupt request is pending for Timer 0. 1 = An interrupt request from Timer 0 is awaiting service.

# Timers

Z8 Encore! XP 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 about using the baud rate generator as an additional timer, see the Universal Asynchronous Receiver/Transmitter chapter on page 97.

## **PWM SINGLE OUTPUT Mode**

In PWM SINGLE OUTPUT Mode, the timer outputs a 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 initiating the PWM operation:

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 write 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:

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

If TPOL 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 the capture occurs on a rising edge or a falling edge 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.

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 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

7. Return to Step 4 to receive additional data.

## **Receiving Data Using the Interrupt-Driven Method**

The UART Receiver interrupt indicates the availability of new data (as well as error conditions). Observe the following steps to configure the UART receiver for interrupt-driven operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the acceptable baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a `DI` instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Receiver interrupt and set the acceptable priority.
5. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
6. Write to the UART Control 1 Register to enable Multiprocessor (9-bit) mode functions, if appropriate.
  - Set the Multiprocessor Mode Select (`MPEN`) to Enable `MULTIPROCESSOR Mode`
  - Set the Multiprocessor Mode Bits, `MPMD[1:0]`, to select the acceptable address matching scheme
  - Configure the UART to interrupt on received data and errors or errors only (interrupt on errors only is unlikely to be useful for Z8 Encore! XP devices without a DMA block)
7. Write the device address to the Address Compare Register (automatic `MULTIPROCESSOR` modes only).
8. Write to the UART Control 0 Register to:
  - Set the receive enable bit (`REN`) to enable the UART for data reception
  - Enable parity, if appropriate and if multiprocessor mode is not enabled, and select either even or odd parity
9. Execute an `EI` instruction to enable interrupts.

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated interrupt service routine (ISR) performs the following:

Bit	Description (Continued)
[2] BRGCTL	<p><b>Baud Rate Control</b></p> <p>This bit causes an alternate UART behavior depending on the value of the REN bit in the UART Control 0 Register.</p> <p>When the UART receiver is <b>not</b> enabled (REN=0), this bit determines whether the Baud Rate Generator issues interrupts.</p> <p>0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value.</p> <p>1 = The Baud Rate Generator generates a receive interrupt when it counts down to 0.</p> <p>Reads from the Baud Rate High and Low Byte registers return the current BRG count value. When the UART receiver is enabled (REN=1), this bit allows reads from the Baud Rate Registers to return the BRG count value instead of the Reload Value.</p> <p>0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value.</p> <p>1 = Reads from the Baud Rate High and Low Byte registers return the current BRG count value. Unlike the Timers, there is no mechanism to latch the Low Byte when the High Byte is read.</p>
[1] RDAIRQ	<p><b>Receive Data Interrupt Enable</b></p> <p>0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller.</p> <p>1 = Received data does not generate an interrupt request to the Interrupt Controller. Only receiver errors generate an interrupt request.</p>
[0] IREN	<p><b>Infrared Encoder/Decoder Enable</b></p> <p>0 = Infrared encoder/decoder is disabled. UART operates normally.</p> <p>1 = Infrared encoder/decoder is enabled. The UART transmits and receives data through the infrared encoder/decoder.</p>

## 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:

$$ADC_{comp} = (ADC_{uncomp} - OFFCAL) + ((ADC_{uncomp} - OFFCAL) * GAINCAL) / 2$$

where GAINCAL is the gain calibration byte, OFFCAL is the offset calibration byte and  $ADC_{uncomp}$  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.

## ADC Data High Byte Register

The ADC Data High Byte Register contains the upper eight bits of the ADC output. The output is an 11-bit two's complement value. During a single-shot conversion, this value is invalid. 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 76. ADC Data High Byte Register (ADCD\_H)**

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

Bit	Description
[7:0]	<b>ADC Data High Byte</b>
ADCDH	This byte contains the upper eight bits of the ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the most recent conversion output is held in this register. These bits are undefined after a Reset.



## Operation

The following section describes the operation of the OCD.

### OCD Interface

The OCD 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, in that transmit and receive 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 from the F0823 Series products to 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 Figure 23 and Figure 24. The recommended method is the buffered implementation depicted in Figure 24. The DBG pin has an internal pull-up resistor which is sufficient for some applications (for more details about the pull-up current, see the [Electrical Characteristics](#) chapter on page 196). For OCD operation at higher data rates or in noisy systems, Zilog recommends an external pull-up resistor.

---

**! Caution:** For operation of the OCD, 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 may require an external pull-up resistor to ensure proper operation.

---

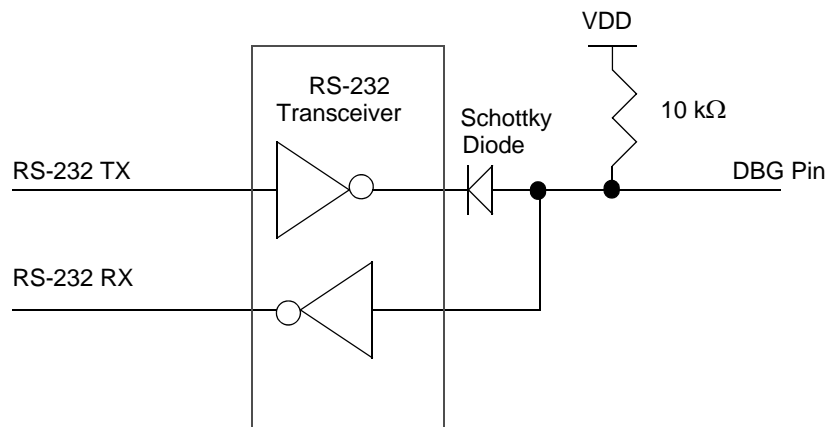


Figure 23. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 1 of 2

**Read Program Counter (07H).** The Read Program Counter command reads the value in 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, 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 can be written 1–256 bytes at a time (256 bytes can be 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 can be read 1–256 bytes at a time (256 bytes can be 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 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 can be written 1–65536 bytes at a time (65536 bytes can be 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 can be read 1–65536 bytes at a time (65536 bytes can be 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.

# Oscillator Control

Z8 Encore! XP F0823 Series devices uses three possible clocking schemes, each user-selectable. These three schemes are:

- On-chip precision trimmed RC oscillator
- External clock drive
- On-chip low power Watchdog Timer oscillator

In addition, F0823 Series devices contain clock failure detection and recovery circuitry, which allow continued operation despite a 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.

## System Clock Selection

The oscillator control block selects from the available clocks. Table 104 details each clock source and its usage.

**Table 104. Oscillator Configuration and Selection**

Clock Source	Characteristics	Required Setup
Internal Precision RC Oscillator	<ul style="list-style-type: none"><li>• 32.8kHz or 5.53MHz</li><li>• <math>\pm 4\%</math> accuracy when trimmed</li><li>• No external components required</li></ul>	<ul style="list-style-type: none"><li>• Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator at either 5.53MHz or 32.8kHz</li></ul>
External Clock Drive	<ul style="list-style-type: none"><li>• 0 to 20MHz</li><li>• Accuracy dependent on external clock source</li></ul>	<ul style="list-style-type: none"><li>• Write GPIO registers to configure PB3 pin for external clock function</li><li>• Unlock and write OSCCTL to select external system clock</li><li>• Apply external clock signal to GPIO</li></ul>
Internal Watchdog Timer Oscillator	<ul style="list-style-type: none"><li>• 10kHz nominal</li><li>• <math>\pm 40\%</math> accuracy; no external components required</li><li>• Very Low power consumption</li></ul>	<ul style="list-style-type: none"><li>• Enable WDT if not enabled and wait until WDT Oscillator is operating.</li><li>• Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator</li></ul>

**Table 108. 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 109 lists additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

**Table 109. 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.

$\text{dst} \leftarrow \text{dst} + \text{src}$

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

**Table 119. Opcode Map Abbreviations**

<b>Abbreviation</b>	<b>Description</b>	<b>Abbreviation</b>	<b>Description</b>
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

## 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.

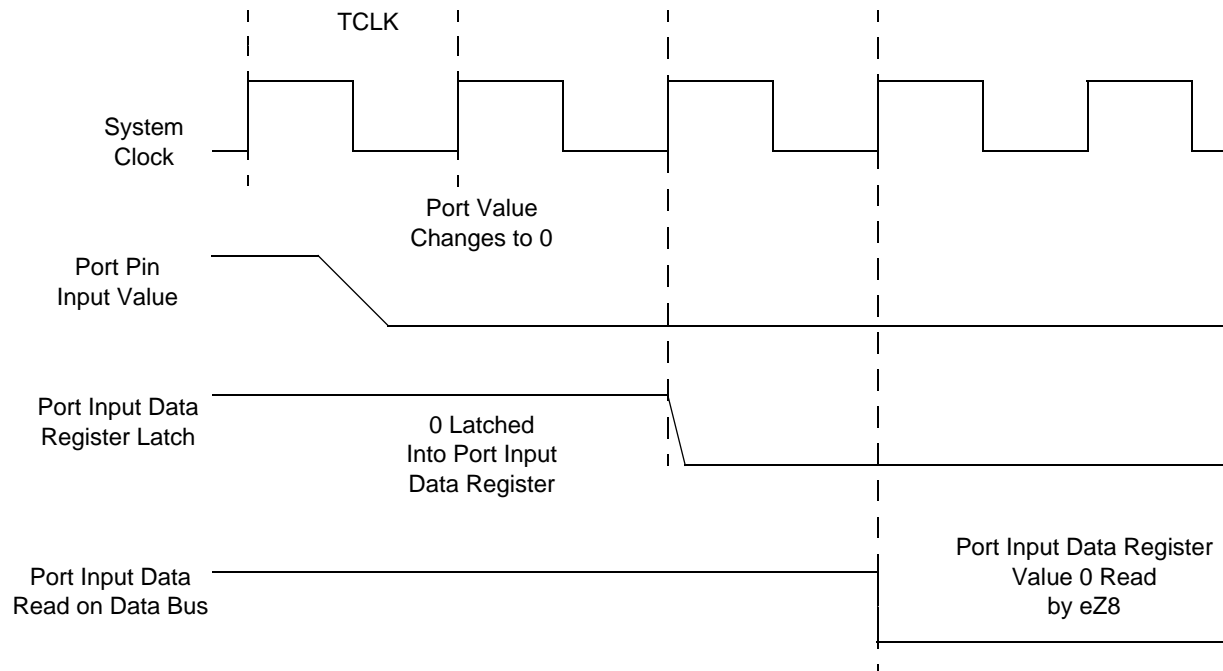
**Table 121. DC Characteristics**

$T_A = -40^{\circ}\text{C to } +105^{\circ}\text{C}$ (unless otherwise specified)						
Symbol	Parameter	Minimum	Typical	Maximum	Units	Conditions
$V_{DD}$	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
$T_{S\_PORT}$	Port Input Transition to $X_{IN}$ 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)	1 $\mu$ s	

**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 8 KB Flash</b>								
<b>Standard Temperature: 0°C to 70°C</b>								
Z8F0813PB005SG	8 KB	1 KB	6	12	2	0	1	PDIP 8-pin package
Z8F0813QB005SG	8 KB	1 KB	6	12	2	0	1	QFN 8-pin package
Z8F0813SB005SG	8 KB	1 KB	6	12	2	0	1	SOIC 8-pin package
Z8F0813SH005SG	8 KB	1 KB	16	18	2	0	1	SOIC 20-pin package
Z8F0813HH005SG	8 KB	1 KB	16	18	2	0	1	SSOP 20-pin package
Z8F0813PH005SG	8 KB	1 KB	16	18	2	0	1	PDIP 20-pin package
Z8F0813SJ005SG	8 KB	1 KB	24	18	2	0	1	SOIC 28-pin package
Z8F0813HJ005SG	8 KB	1 KB	24	18	2	0	1	SSOP 28-pin package
Z8F0813PJ005SG	8 KB	1 KB	24	18	2	0	1	PDIP 28-pin package
<b>Extended Temperature: -40°C to 105°C</b>								
Z8F0813PB005EG	8 KB	1 KB	6	12	2	0	1	PDIP 8-pin package
Z8F0813QB005EG	8 KB	1 KB	6	12	2	0	1	QFN 8-pin package
Z8F0813SB005EG	8 KB	1 KB	6	12	2	0	1	SOIC 8-pin package
Z8F0813SH005EG	8 KB	1 KB	16	18	2	0	1	SOIC 20-pin package
Z8F0813HH005EG	8 KB	1 KB	16	18	2	0	1	SSOP 20-pin package
Z8F0813PH005EG	8 KB	1 KB	16	18	2	0	1	PDIP 20-pin package
Z8F0813SJ005EG	8 KB	1 KB	24	18	2	0	1	SOIC 28-pin package
Z8F0813HJ005EG	8 KB	1 KB	24	18	2	0	1	SSOP 28-pin package
Z8F0813PJ005EG	8 KB	1 KB	24	18	2	0	1	PDIP 28-pin package



**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 Development Kit</b>								
Z8F08A28100KITG								Z8 Encore! XP F082A Series Development Kit (20- and 28-Pin)
Z8F04A28100KITG								Z8 Encore! XP F042A Series Development Kit (20- and 28-Pin)
Z8F04A08100KITG								Z8 Encore! XP F042A Series Development Kit (8-Pin)
ZUSBSC00100ZACG								USB Smart Cable Accessory Kit
ZUSBOPTSC01ZACG								Opto-Isolated USB Smart Cable Accessory Kit
ZENETSC0100ZACG								Ethernet Smart Cable Accessory Kit