



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	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f0223hh005eg">https://www.e-xfl.com/product-detail/zilog/z8f0223hh005eg</a>

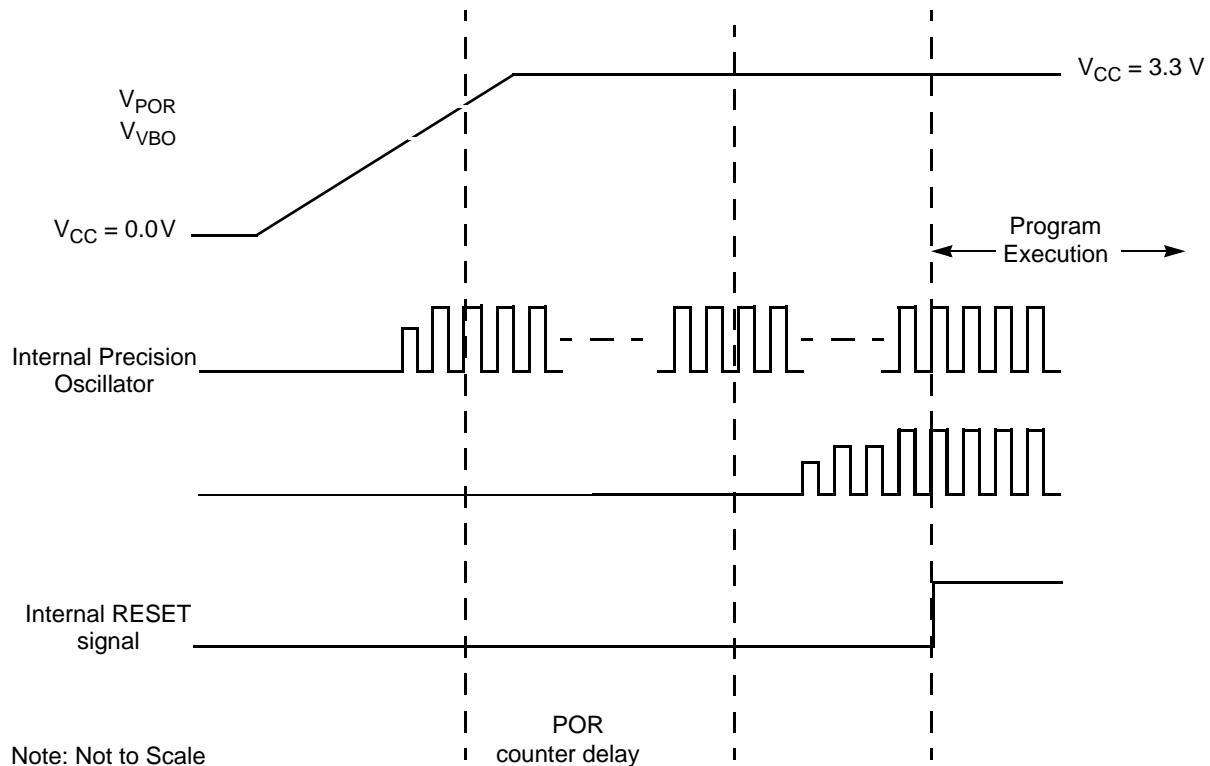


Figure 5. Power-On Reset Operation

## Voltage Brown-Out Reset

The devices in the Z8 Encore! XP F0823 Series provide low VBO protection. The VBO circuit senses when the supply voltage drops to an unsafe level (below the VBO threshold voltage) and forces the device into the Reset state. While the supply voltage remains below the POR voltage threshold ( $V_{POR}$ ), the VBO block holds the device in the Reset.

After the supply voltage again exceeds the Power-On Reset voltage threshold, the device progresses through a full System Reset sequence, as described in the [Power-On Reset](#) section on page 23. Following POR, the POR status bit in the Reset Status (RSTSTAT) Register is set to 1. Figure 6 displays Voltage Brown-Out operation. For the VBO and POR threshold voltages ( $V_{VBO}$  and  $V_{POR}$ ), see the [Electrical Characteristics](#) chapter on page 196.

The VBO circuit can be either enabled or disabled during STOP Mode. Operation during STOP Mode is set by the VBO\_AO Flash Option bit. For information about configuring VBO\_AO, see the [Flash Option Bits](#) chapter on page 146.

clock and reset signals, the required reset duration can be as short as three clock periods and as long as four. A reset pulse three clock cycles in duration might trigger a reset; a pulse four cycles in duration always triggers a reset.

While the  $\overline{\text{RESET}}$  input pin is asserted Low, the Z8 Encore! XP F0823 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 WDT Control (WDTCTL) register is set to 1.

## External Reset Indicator

During System Reset or when enabled by the GPIO logic (see [the Port A–C Control Registers section on page 42](#)), 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! XP F0823 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.

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 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 OCD block is not reset but the rest 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 into STOP Mode when eZ8 CPU executes a STOP instruction. For more details about STOP Mode, see [the Low-Power Modes section on page 30](#). During Stop Mode Recovery, the CPU is held in reset for 66 IPO cycles if the crystal oscillator is disabled or 5000 cycles if it is enabled. The SMR delay also included the time required to start up the IPO.

Stop Mode Recovery does not affect on-chip registers other than the Watchdog Timer Control Register (WDTCTL) 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.

## Port A–C Data Direction Subregisters

The Port A–C Data Direction Subregister is accessed through the Port A–C Control Register by writing 01H to the Port A–C Address Register; see Table 22.

**Table 22. Port A–C 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–C Address Register, accessible through the Port A–C Control Register.							

Bit	Description
[7:0]	<b>Data Direction</b>
DDx	<p>These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction register setting.</p> <p>0 = Output. Data in the Port A–C Output Data Register is driven onto the port pin.</p> <p>1 = Input. The port pin is sampled and the value written into the Port A–C Input Data Register.</p> <p>The output driver is tristated.</p>

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

## Port A–C Alternate Function Subregisters

The Port A–C Alternate Function Subregister (Table 23) is accessed through the Port A–C Control Register by writing 02H to the Port A–C Address Register. The Port A–C Alternate Function subregisters enable the alternate function selection on pins. If disabled, pins function as GPIO. If enabled, select one of four alternate functions using alternate function set subregisters 1 and 2 as described in the [Port A–C Alternate Function Set 1 Subregisters](#) section on page 48 and the [Port A–C Alternate Function Set 2 Subregisters](#) section on page 49. See the [GPIO Alternate Functions](#) section on page 34 to determine the alternate function associated with each port pin.

---

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

---



## Port A–C Output Data Register

The Port A–C Output Data Register (Table 31) controls the output data to the pins.

**Table 31. Port A–C Output Data Register (PxOUT)**

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							

Bit	Description
[7:0] PxOUT	<b>Port Output Data</b> 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 32, 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 32. 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	<b>LED Drive Enable</b> 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.

# Interrupt Controller

The interrupt controller on the Z8 Encore! XP F0823 Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 unique interrupt vectors
  - 12 GPIO port pin interrupt sources (two are shared)
  - 8 on-chip peripheral interrupt sources (two are shared)
- Flexible GPIO interrupts
  - Eight selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Watchdog Timer can be 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 about interrupt servicing by the eZ8 CPU, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#) available for download at [www.zilog.com](http://www.zilog.com).

## Interrupt Vector Listing

Table 35 lists all of 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 following odd Program Memory address.

---

► **Note:** Some port interrupts are not available on the 8- and 20-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

---

- Execution of an Return from Interrupt (IRET) instruction
- 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 35](#) on page 55. 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 35. 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:** 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.

---

Table 40. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	Reserved		ADCENH
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	FC1H							

Bit	Description
[7]	<b>Reserved</b> This bit is reserved and must be programmed to 0.
[6] T1ENH	<b>Timer 1 Interrupt Request Enable High Bit</b>
[5] T0ENH	<b>Timer 0 Interrupt Request Enable High Bit</b>
[4] U0RENH	<b>UART 0 Receive Interrupt Request Enable High Bit</b>
[3] U0TENH	<b>UART 0 Transmit Interrupt Request Enable High Bit</b>
[2:1]	<b>Reserved</b> These bits are reserved and must be programmed to 00.
[0] ADCENH	<b>ADC Interrupt Request Enable High Bit</b>

Table 41. IRQ0 Enable Low Bit Register (IRQ0ENL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENL	T0ENL	U0RENL	U0TENL	Reserved		ADCENL
RESET	0	0	0	0	0	0		0
R/W	R	R/W	R/W	R/W	R/W	R		R/W
Address	FC2H							

Bit	Description
[7]	<b>Reserved</b> This bit is reserved and must be programmed to 0 when read.
[6] T1ENL	<b>Timer 1 Interrupt Request Enable Low Bit</b>

## IRQ2 Enable High and Low Bit Registers

Table 45 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers (Table 46 and Table 47) form a priority encoded enabling for interrupts in the Interrupt Request 2 register. Priority is generated by setting bits in each register.

**Table 45. 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: where x indicates the register bits from 0–7.

**Table 46. 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]	<b>Reserved</b> These bits are reserved and must be programmed to 0000.
[3] C3ENH	<b>Port C3 Interrupt Request Enable High Bit</b>
[2] C2ENH	<b>Port C2 Interrupt Request Enable High Bit</b>
[1] C1ENH	<b>Port C1 Interrupt Request Enable High Bit</b>
[0] C0ENH	<b>Port C0 Interrupt Request Enable High Bit</b>

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 57. 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	<b>Timer Mode High Bit</b> This bit along with the TMODE field in TxCTL1 Register determines the operating mode of the timer. This is the most-significant bit of the Timer mode selection value.
[6:5] TICONFIG	<b>Timer Interrupt Configuration</b> This field configures timer interrupt definition. 0x = Timer Interrupt occurs on all defined reload, compare and input events. 10 = Timer Interrupt only on defined input capture/deassertion events. 11 = Timer Interrupt only on defined reload/compare events.
[4] 	<b>Reserved</b> This bit is reserved and must be programmed to 0.
[3:1] PWMD	<b>PWMD—PWM Delay value</b> 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 cycles delay. 010 = 4 cycles delay. 011 = 8 cycles delay. 100 = 16 cycles delay. 101 = 32 cycles delay. 110 = 64 cycles delay. 111 = 128 cycles delay.
[0] INPCAP	<b>Input Capture Event</b> This bit indicates if the most recent timer interrupt is caused by a Timer Input capture event. 0 = Previous timer interrupt is not a result of Timer Input capture event. 1 = Previous timer interrupt is a result of Timer Input capture event.

### Timer 0–1 Control Register 1

The Timer 0–1 Control (TxCTL1) registers enable/disable the timers, set the prescaler value, and determine the timer operating mode.

## 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 information 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 information, see **the Reset and Stop Mode Recovery** chapter on page 21.

## Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watchdog Timer Control Register (WDTCTL) 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 just listed. 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 can occur 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 94

Watchdog Timer Reload Upper Byte Register (WDTU): see page 95

**Table 61. Watchdog Timer Reload Upper Byte Register (WDTU)**

Bit	7	6	5	4	3	2	1	0
Field	WDTU							
RESET	00H							
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF1H							
Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value.								

Bit	Description
[7:0] WDTU	<b>WDT Reload Upper Byte</b> Most significant byte (MSB), Bits[23:16], of the 24-bit WDT reload value.

**Table 62. Watchdog Timer Reload High Byte Register (WDTH)**

Bit	7	6	5	4	3	2	1	0
Field	WDTH							
RESET	04H							
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF2H							
Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value.								

Bit	Description
[7:0] WDTH	<b>WDT Reload High Byte</b> Middle byte, Bits[15:8], of the 24-bit WDT reload value.

**Table 63. Watchdog Timer Reload Low Byte Register (WDTL)**

Bit	7	6	5	4	3	2	1	0
Field	WDTL							
RESET	00H							
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	FF3H							
Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value.								



## UART Transmit Data Register

Data bytes written to the UART Transmit Data Register (Table 64) are shifted out on the TXD<sub>x</sub> pin. The Write-only UART Transmit Data Register shares a Register File address with the read-only UART Receive Data Register.

**Table 64. UART Transmit Data Register (U0TXD)**

Bit	7	6	5	4	3	2	1	0
Field	TXD							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
Address	F40H							

Bit	Description
[7:0] TXD	<b>Transmit Data</b> UART transmitter data byte to be shifted out through the TXD <sub>x</sub> pin.

## UART Receive Data Register

Data bytes received through the RXD<sub>x</sub> pin are stored in the UART Receive Data Register (Table 65). The read-only UART Receive Data Register shares a Register File address with the Write-only UART Transmit Data Register.

**Table 65. UART Receive Data Register (U0RXD)**

Bit	7	6	5	4	3	2	1	0
Field	RXD							
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	F40H							

Bit	Description
[7:0] RXD	<b>Receive Data</b> UART receiver data byte from the RXD <sub>x</sub> pin.

## 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.

**Table 81. 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]	<b>Flash Command</b>
FCMD	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.

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

**Table 85. 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] **Flash Frequency High Byte**  
FFREQH High byte of the 16-bit Flash Frequency value.

**Table 86. 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] **Flash Frequency Low Byte**  
FFREQL Low byte of the 16-bit Flash Frequency value.

# On-Chip Debugger

Z8 Encore! XP F0823 Series devices contain an integrated On-Chip Debugger (OCD) which 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.

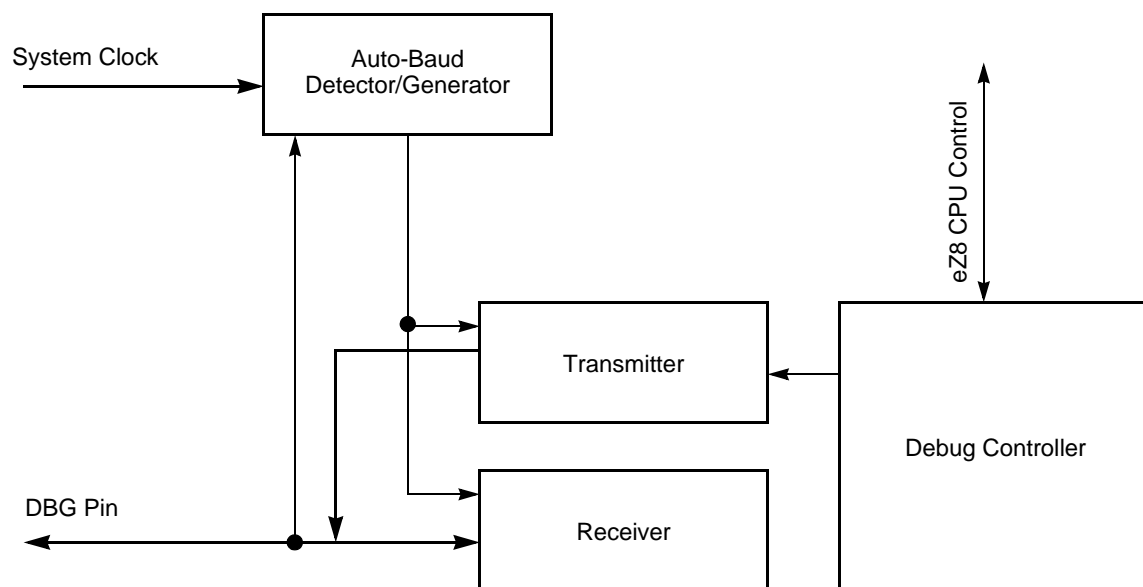


Figure 22. On-Chip Debugger Block Diagram

**Table 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

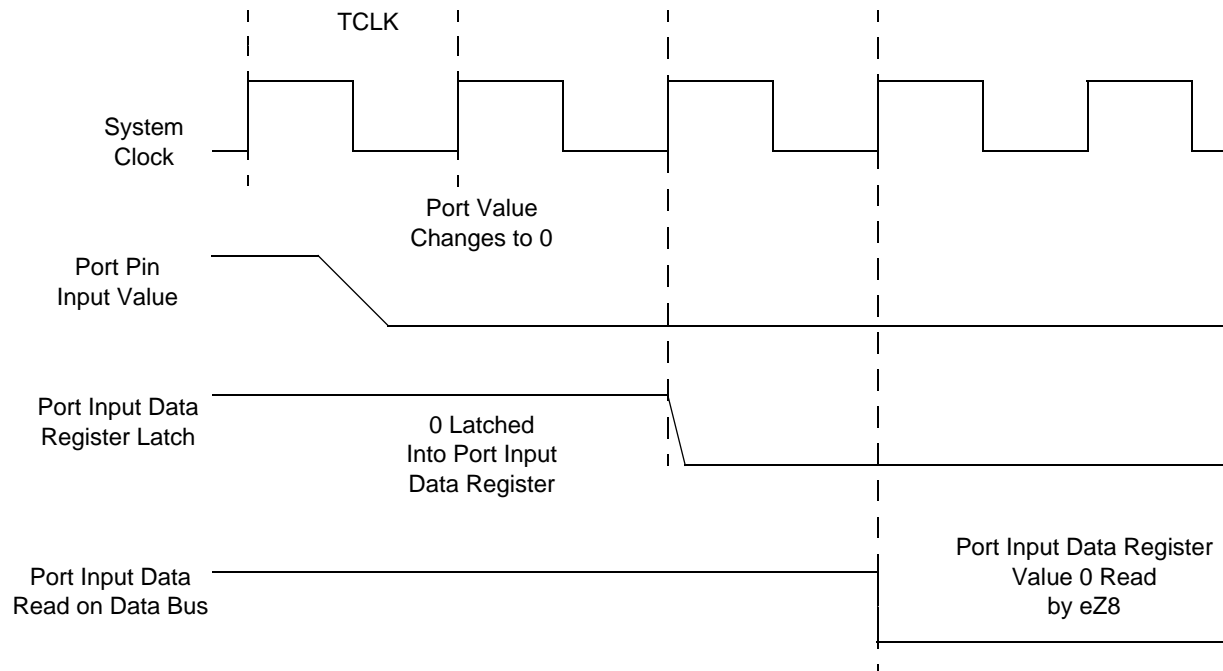
**Table 111. Bit Manipulation Instructions**

<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

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	

compare with carry - extended addressing 178  
 complement 181  
 complement carry flag 179, 180  
 condition code 176  
 continuous conversion (ADC) 124  
 CONTINUOUS mode 88  
 control register definition, UART 108  
 Control Registers 13, 16  
 COUNTER modes 89  
 CP 178  
 CPC 178  
 CPCX 178  
 CPU and peripheral overview 4  
 CPU control instructions 180  
 CPX 178  
 Customer Support 230

## **D**

DA 176, 178  
 data memory 15  
 DC characteristics 197  
 debugger, on-chip 156  
 DEC 178  
 decimal adjust 178  
 decrement 178  
 decrement and jump non-zero 181  
 decrement word 178  
 DECW 178  
 destination operand 177  
 device, port availability 33  
 DI 180  
 direct address 176  
 disable interrupts 180  
 DJNZ 181  
 dst 177

## **E**

EI 180  
 electrical characteristics 196  
     ADC 203  
     flash memory and timing 202  
     GPIO input data sample timing 204

Watchdog Timer 202, 204  
 enable interrupt 180  
 ER 176  
 extended addressing register 176  
 external pin reset 25  
 eZ8 CPU features 4  
 eZ8 CPU instruction classes 178  
 eZ8 CPU instruction notation 176  
 eZ8 CPU instruction set 174  
 eZ8 CPU instruction summary 182

## **F**

FCTL register 141, 148, 149  
 features, Z8 Encore! 1  
 first opcode map 194  
 FLAGS 177  
 flags register 177  
 flash  
     controller 4  
     option bit address space 149  
     option bit configuration - reset 146  
     program memory address 0000H 149  
     program memory address 0001H 150  
 flash memory 134  
     arrangement 135  
     byte programming 139  
     code protection 137  
     configurations 134  
     control register definitions 141, 148  
     controller bypass 140  
     electrical characteristics and timing 202  
     flash control register 141, 148, 149  
     flash option bits 138  
     flash status register 142  
     flow chart 136  
     frequency high and low byte registers 144  
     mass erase 139  
     operation 135  
     operation timing 137  
     page erase 139  
     page select register 142, 144  
 FPS register 142, 144  
 FSTAT register 142



part selection guide 2