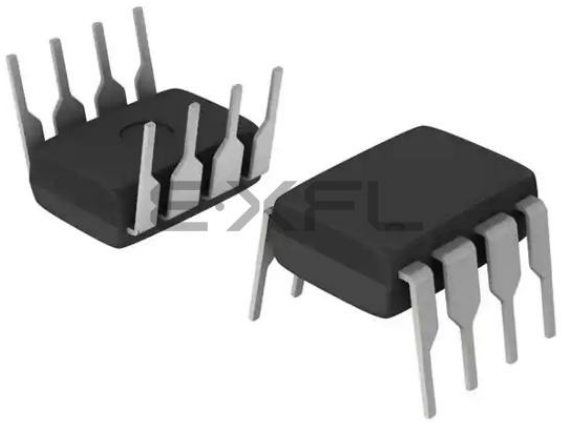


Welcome to E-XFL.COM

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	Active
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	6
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0813pb005eg

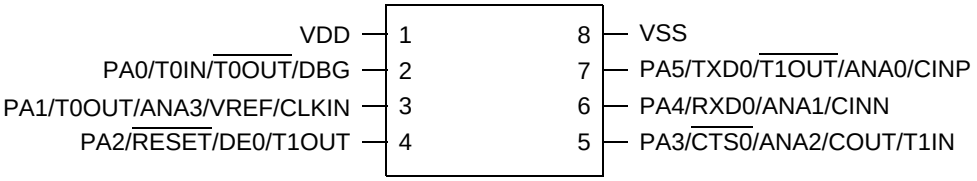


Figure 2. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 8-Pin SOIC, QFN/MLF-S, or PDIP Package*

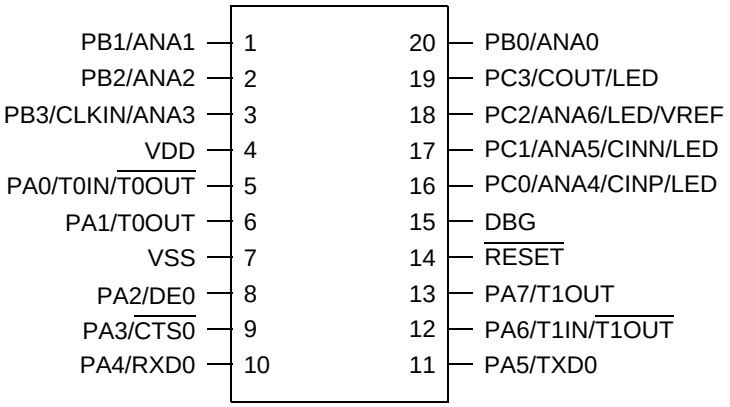


Figure 3. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 20-Pin SOIC, SSOP or PDIP Package*

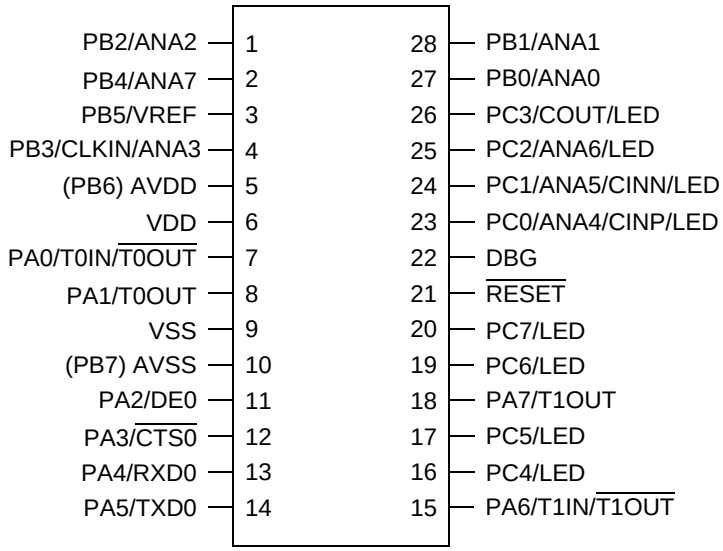


Figure 4. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 28-Pin SOIC, SSOP or PDIP Package*

Note: *Analog input alternate functions (ANA) are not available on Z8F0x13 devices.

Signal Descriptions

Table 3 lists the Z8 Encore! XP F0823 Series signals. To determine the signals available for the specific package styles, see **the** Pin Configurations **section on page 7**.

Table 3. Signal Descriptions

Signal Mnemonic	I/O	Description
General-Purpose I/O Ports A–D		
PA[7:0]	I/O	Port A. These pins are used for general-purpose I/O.
PB[7:0][1]	I/O	Port B. These pins are used for general-purpose I/O. PB6 and PB7 are available only in those devices without an ADC.
PC[7:0]	I/O	Port C. These pins are used for general-purpose I/O.
UART Controllers		
TXD0	O	Transmit Data. This signal is the transmit output from the UART and IrDA.
RXD0	I	Receive Data. This signal is the receive input for the UART and IrDA.
$\overline{\text{CTS0}}$	I	Clear To Send. This signal is the flow control input for the UART.
DE	O	Driver Enable. This signal allows automatic control of external RS-485 drivers. This signal is approximately the inverse of the TXE (Transmit Empty) bit in the UART Status 0 Register. The DE signal can be used to ensure the external RS-485 driver is enabled when data is transmitted by the UART.
Timers		
T0OUT/T1OUT	O	Timer Output 0–1. These signals are output from the timers.
$\overline{\text{T0OUT}}$/$\overline{\text{T1OUT}}$	O	Timer Complement Output 0–1. These signals are output from the timers in PWM DUAL OUTPUT Mode.
T0IN/T1IN	I	Timer Input 0–1. These signals are used as the capture, gating and counter inputs. The T0IN signal is multiplexed $\overline{\text{T0OUT}}$ signals.
Comparator		
CINP/CINN	I	Comparator Inputs. These signals are the positive and negative inputs to the comparator.

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_{DD} and AV_{SS}.
2. The AV_{DD} and AV_{SS} signals are available only in 28-pin packages with ADC. They are replaced by PB6 and PB7 on 28-pin packages without ADC.

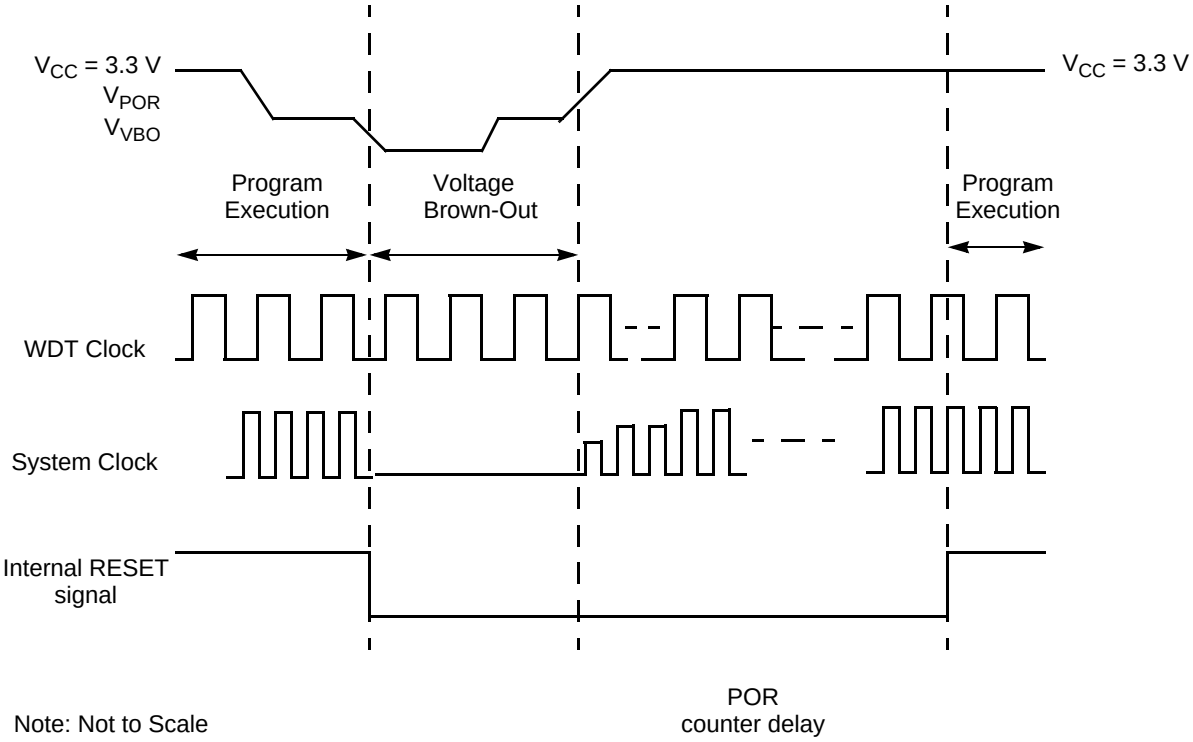


Figure 6. Voltage Brown-Out Reset Operation

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

The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the STOP bit in the Watchdog Timer Control Register is set to 1. Table 11 lists the Stop Mode Recovery sources and resulting actions. The section following the table provides more detailed information about each of the Stop Mode Recovery sources.

Table 11. Stop Mode Recovery Sources and Resulting Action

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

Stop Mode Recovery Using Watchdog Timer Time-Out

If the Watchdog Timer times out during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Watchdog Timer Control Register, the WDT and STOP bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and Z8 Encore! XP F0823 Series device is configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery Using a GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery.

Note: The SMR pulses shorter than specified does not trigger a recovery. When this happens, the STOP bit in the Reset Status (RSTSTAT) Register is set to 1.

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

Table 17. Port Alternate Function Mapping (Non 8-Pin Parts)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A[1]	PA0	T0IN/$\overline{\text{T0OUT}}$	Timer 0 Input/Timer 0 Output Complement	N/A
		Reserved		
	PA1	T0OUT	Timer 0 Output	
		Reserved		
	PA2	$\overline{\text{DE0}}$	$\overline{\text{UART 0 Driver Enable}}$	
		Reserved		
	PA3	$\overline{\text{CTS0}}$	$\overline{\text{UART 0 Clear to Send}}$	
		Reserved		
	PA4	RXD0/IRRX0	UART 0 / IrDA 0 Receive Data	
		Reserved		
	PA5	TXD0/IRTX0	UART 0 / IrDA 0 Transmit Data	
		Reserved		
	PA6	T1IN/$\overline{\text{T1OUT}}$[2]	Timer 1 Input/Timer 1 Output Complement	
		Reserved		
	PA7	T1OUT	Timer 1 Output	
		Reserved		

Notes:
1. Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in the Port A–C Alternate Function Subregisters section on page 43 automatically enables the associated alternate function.
2. Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in the Timer Pin Signal Operation section on page 83.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, alternate function selection as described in the Port A–C Alternate Function Subregisters section on page 43 must also be enabled.
4. V_{REF} is available on PB5 in 28-pin products only.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, Alternate Function selection as described in the Port A–C Alternate Function Subregisters section on page 43 must also be enabled.
6. V_{REF} is available on PC2 in 20-pin parts only.

Table 23. Port A–C Alternate Function Subregisters (PxAF)

Bit	7	6	5	4	3	2	1	0
Field	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
RESET	00H (Ports A–C); 04H (Port A of 8-pin device)							
R/W	R/W							
Address	If 02H in Port A–C Address Register, accessible through the Port A–C Control Register							

Bit	Description
[7:0] AFx	**Port Alternate Function enabled** 0 = The port pin is in NORMAL Mode and the DDx bit in the Port A–C Data Direction Subregister determines the direction of the pin. 1 = The alternate function selected through Alternate Function Set subregisters is enabled. Port pin operation is controlled by the alternate function.

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

Port A–C Output Control Subregisters

The Port A–C Output Control Subregister (Table 24) is accessed through the Port A–C Control Register by writing 03H to the Port A–C Address Register. Setting the bits in the Port A–C Output Control subregisters to 1 configures the specified port pins for open-drain operation. These subregisters affect the pins directly and, as a result, alternate functions are also affected.

Table 24. Port A–C Output Control Subregisters (PxOC)

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

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

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

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

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]	**Reserved** This bit is reserved and must be programmed to 0 when read.
[6] T1ENL	**Timer 1 Interrupt Request Enable Low Bit**

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.

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:

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

In CAPTURE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

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

COMPARE Mode

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

If the Timer reaches FFFFH, the timer rolls over to 0000H and continue counting. Observe the following steps to configure a timer for COMPARE Mode and to initiate the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COMPARE Mode
   - Set the prescale value
   - Set the initial logic level (High or Low) for the Timer Output alternate function, if appropriate
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the compare value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers.

7. Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge.

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

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

Reading the Timer Count Values

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

Timer Pin Signal Operation

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

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

Timer Control Register Definitions

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

Timer 0–1 High and Low Byte Registers: see page 83

Timer Reload High and Low Byte Registers: see page 84

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

Timer 0–1 Control Registers: see page 86

Timer 0–1 High and Low Byte Registers

The Timer 0–1 High and Low Byte (TxH and TxL) registers (Table 51 and Table 52) contain the current 16-bit timer count value. When the timer is enabled, a read from TxH

Watchdog Timer Refresh

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

When Z8 Encore! XP F0823 Series devices are operating in DEBUG Mode (using the OCD), the Watchdog Timer is continuously refreshed to prevent any Watchdog Timer time-outs.

Watchdog Timer Time-Out Response

The Watchdog Timer times out when the counter reaches 000000H. A time-out of the Watchdog Timer generates either an interrupt or a system reset. The WDT_RES Flash Option Bit determines the time-out response of the Watchdog Timer. For information about programming of the WDT_RES Flash Option Bit, see **the** Flash Option Bits chapter on page 146.

WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the Watchdog Timer issues an interrupt request to the interrupt controller and sets the WDT status bit in the Watchdog Timer Control Register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executing code from the vector address. After time-out and interrupt generation, the Watchdog Timer counter rolls over to its maximum value of FFFFFH and continues counting. The Watchdog Timer counter is not automatically returned to its Reload Value.

The Reset Status Register (see the Reset Status Register section on page 28) must be read before clearing the WDT interrupt. This read clears the WDT time-out Flag and prevents further WDT interrupts for immediately occurring.

WDT Interrupt in STOP Mode

If configured to generate an interrupt when a time-out occurs and F0823 Series are in STOP Mode, the Watchdog Timer automatically initiates a Stop Mode Recovery and generates an interrupt request. Both the WDT status bit and the STOP bit in the Watchdog Timer Control Register are set to 1 following a WDT time-out in STOP Mode. For more information about Stop Mode Recovery, see **the** Reset and Stop Mode Recovery chapter on page 21.

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

– Set or clear CTSE to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.

8. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data transmission. Because the UART Transmit Data Register is empty, an interrupt is generated immediately. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Write the UART Control 1 Register to select the multiprocessor bit for the byte to be transmitted:

   Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.

2. Write the data byte to the UART Transmit Data Register. The transmitter automatically transfers the data to the Transmit Shift register and transmits the data.
3. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.
4. Execute the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data Register to again become empty.

Receiving Data Using the Polled Method

Observe the following steps to configure the UART for polled data reception:

1. Write to the UART Baud Rate High and Low Byte registers to set an acceptable baud rate for the incoming data stream.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Write to the UART Control 1 Register to enable MULTIPROCESSOR Mode functions, if appropriate.
4. 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
5. Check the RDA bit in the UART Status 0 Register to determine if the Receive Data Register contains a valid data byte (indicated by a 1). If RDA is set to 1 to indicate available data, continue to Step 6. If the Receive Data Register is empty (indicated by a 0), continue to monitor the RDA bit awaiting reception of the valid data.
6. Read data from the UART Receive Data Register. If operating in MULTIPROCESSOR (9-bit) Mode, further actions may be required depending on the MULTIPROCESSOR Mode bits MPMD[1:0].

Bit	Description (Continued)
[2] BRGCTL	**Baud Rate Control** This bit causes an alternate UART behavior depending on the value of the REN bit in the UART Control 0 Register. When the UART receiver is **not** enabled (REN=0), this bit determines whether the Baud Rate Generator issues interrupts. 0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value. 1 = The Baud Rate Generator generates a receive interrupt when it counts down to 0. 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. 0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value. 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.
[1] RDAIRQ	**Receive Data Interrupt Enable** 0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller. 1 = Received data does not generate an interrupt request to the Interrupt Controller. Only receiver errors generate an interrupt request.
[0] IREN	**Infrared Encoder/Decoder Enable** 0 = Infrared encoder/decoder is disabled. UART operates normally. 1 = Infrared encoder/decoder is enabled. The UART transmits and receives data through the infrared encoder/decoder.

– CEN resets to 0 to indicate the conversion is complete

6. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

Continuous Conversion

When configured for continuous conversion, the ADC continuously performs an analog-to-digital conversion on the selected analog input. Each new data value over-writes the previous value stored in the ADC Data registers. An interrupt is generated after each conversion.

! Caution: In CONTINUOUS Mode, ADC updates are limited by the input signal bandwidth of the ADC and the latency of the ADC and its digital filter. Step changes at the input are not detected at the next output from the ADC. The response of the ADC (in all modes) is limited by the input signal bandwidth and the latency.

Observe the following steps for setting up the ADC and initiating continuous conversion:

1. Enable the acceptable analog input by configuring the general-purpose I/O pins for alternate function. This action disables the digital input and output driver.
2. Write the ADC Control/Status Register 1 to configure the ADC:
   – Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in the ADC Control/Status Register 1.
3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control Register can be written simultaneously:
   – Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
   – Set CONT to 1 to select continuous conversion.
   – If the internal VREF must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
   – Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in ADC Control Register 0.
   – Set CEN to 1 to start the conversions.

Bit	Description (Continued)
[3:0] ANAIN	**Analog Input Select** These bits select the analog input for conversion. Not all port pins in this list are available in all packages for Z8 Encore! XP F0823 Series. For information about the port pins available with each package style, see **the** Pin Description section on page 7. Do not enable unavailable analog inputs. Usage of these bits changes depending on the buffer mode selected in ADC Control/Status Register 1. For the reserved values, all input switches are disabled to avoid leakage or other undesirable operation. ADC samples taken with reserved bit settings are undefined. **Single-Ended:** 0000 = ANA0. 0001 = ANA1. 0010 = ANA2. 0011 = ANA3. 0100 = ANA4. 0101 = ANA5. 0110 = ANA6. 0111 = ANA7. 1000 = Reserved. 1001 = Reserved. 1010 = Reserved. 1011 = Reserved. 1100 = Reserved. 1101 = Reserved. 1110 = Reserved. 1111 = Reserved.

Flash Memory

The products in Z8 Encore! XP F0823 Series features either 8KB (8192), 4KB (4096), 2KB (2048) or 1KB (1024) of nonvolatile Flash memory with read/write/erase capability. Flash Memory can be programmed and erased in-circuit by either user code or through the On-Chip Debugger.

The Flash Memory array is arranged in pages with 512 bytes per page. The 512-byte page is the minimum Flash block size that can be erased. Each page is divided into 8 rows of 64 bytes.

For program/data protection, the Flash memory is also divided into sectors. In the Z8 Encore! XP F0823 Series, these sectors are either 1024 bytes (in the 8KB devices) or 512 bytes in size (all other memory sizes); each sector maps to a page. Page and sector sizes are not generally equal.

The first two bytes of the Flash program memory are used as Flash Option bits. For more information about their operation, see the Flash Option Bits chapter on page 146.

Table 79 describes the Flash memory configuration for each device in the Z8 Encore! XP F0823 Series. Figure 20 displays the Flash memory arrangement.

Table 79. Z8 Encore! XP F0823 Series Flash Memory Configurations

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash Sector Size (bytes)
Z8F08x3	8 (8192)	16	0000H–1FFFH	1024
Z8F04x3	4 (4096)	8	0000H–0FFFH	512
Z8F02x3	2 (2048)	4	0000H–07FFH	512
Z8F01x3	1 (1024)	2	0000H–03FFH	512

- If the PA2/$\overline{\text{RESET}}$ pin is held Low while a 32-bit key sequence is issued to the PA0/DBG pin, the DBG feature is unlocked. After releasing PA2/$\overline{\text{RESET}}$, it is pulled high. At this point, the PA0/DBG pin can be used to autobaud and cause the device to enter DEBUG Mode. For more details, see the OCD Unlock Sequence (8-Pin Devices Only) section on page 161.

Exiting DEBUG Mode

The device exits DEBUG Mode following any of these operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset
- Watchdog Timer reset
- Asserting the $\overline{\text{RESET}}$ pin Low to initiate a Reset
- Driving the DBG pin Low while the device is in STOP Mode initiates a system reset

OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 Start bit, 8 data bits (least-significant bit first), and 1 Stop bit as displayed in Figure 25.

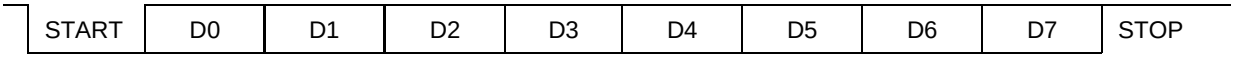


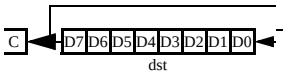
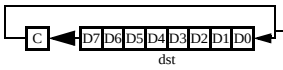
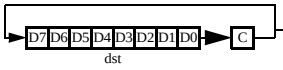
Figure 25. OCD Data Format

Note: When responding to a request for data, the OCD may commence transmitting immediately after receiving the stop bit of an incoming frame. Therefore, when sending the stop bit, the host must not actively drive the DBG pin High for more than 0.5 bit times. Zilog recommends that, if possible, the host drives the DBG pin using an open-drain output.

OCD Autobaud Detector/Generator

To run over a range of baud rates (data bits per second) with various system clock frequencies, the OCD contains an auto-baud detector/generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the host is the character 80H. The character 80H has eight continuous bits Low (one Start bit plus 7 data

Table 118. eZ8 CPU Instruction Summary (Continued)

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

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