



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	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	6
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	16 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f012asb020sc

Register Map

Table 7 provides the address map for the Register File of the Z8 Encore! XP[®] F082A Series devices. Not all devices and package styles in the Z8 Encore! XP F082A Series support the ADC, or all of the GPIO Ports. Consider registers for unimplemented peripherals as Reserved.

Table 7. Register File Address Map

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
General-Purpose RAM				
Z8F082A/Z8F081A Devices				
000–3FF	General-Purpose Register File RAM	—	XX	
400–EFF	Reserved	—	XX	
Z8F042A/Z8F041A Devices				
000–3FF	General-Purpose Register File RAM	—	XX	
400–EFF	Reserved	—	XX	
Z8F022A/Z8F021A Devices				
000–1FF	General-Purpose Register File RAM	—	XX	
200–EFF	Reserved	—	XX	
Z8F012A/Z8F011A Devices				
000–0FF	General-Purpose Register File RAM	—	XX	
100–EFF	Reserved	—	XX	
Timer 0				
F00	Timer 0 High Byte	T0H	00	87
F01	Timer 0 Low Byte	T0L	01	87
F02	Timer 0 Reload High Byte	T0RH	FF	88
F03	Timer 0 Reload Low Byte	T0RL	FF	88
F04	Timer 0 PWM High Byte	T0PWMH	00	88
F05	Timer 0 PWM Low Byte	T0PWML	00	89
F06	Timer 0 Control 0	T0CTL0	00	83
F07	Timer 0 Control 1	T0CTL1	00	84
Timer 1				
F08	Timer 1 High Byte	T1H	00	87
F09	Timer 1 Low Byte	T1L	01	87
F0A	Timer 1 Reload High Byte	T1RH	FF	88
XX=Undefined				

PIN[7:0]—Port Input Data

Sampled data from the corresponding port pin input.

0 = Input data is logical 0 (Low).

1 = Input data is logical 1 (High).

Port A–D Output Data Register

The Port A–D Output Data register ([Table 28](#)) controls the output data to the pins.

Table 28. Port A–D Output Data Register (PxOUT)

BITS	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
ADDR	FD3H, FD7H, FDBH, FDFH							

POUT[7:0]—Port Output Data

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.

LED Drive Enable Register

The LED Drive Enable register ([Table 29](#)) activates the controlled current drive. The Port C pin must first be enabled by setting the Alternate Function register to select the LED function.

Table 29. LED Drive Enable (LEDEN)

BITS	7	6	5	4	3	2	1	0
FIELD	LEDEN[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F82H							

Interrupt Controller

The interrupt controller on the Z8 Encore! XP F082A Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 possible interrupt sources with 18 unique interrupt vectors:
 - Twelve GPIO port pin interrupt sources (two interrupt vectors are shared).
 - Eight on-chip peripheral interrupt sources (two interrupt vectors 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 and LVD can be configured to generate an interrupt.
- Supports vectored as well as polled interrupts

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 on interrupt servicing by the eZ8 CPU, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com.

Interrupt Vector Listing

Table 32 on page 56 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.*

Timers

These Z8 Encore! XP® F082A Series products contain two 16-bit reloadable timers that can be used for timing, event counting, or generation of pulse-width modulated (PWM) signals. The timers' feature 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) may also provide basic timing functionality. For information on using the Baud Rate Generator as an additional timer, see [Universal Asynchronous Receiver/Transmitter](#) on page 97.

Architecture

[Figure 9](#) on page 70 displays the architecture of the timers.

- Set the prescale value.
 - If using the Timer Output alternate function, set the initial output level (High or Low).
2. Write to the Timer High and Low Byte registers to set the starting count value (usually 0001H). This action only affects the first pass in CONTINUOUS mode. After the first timer Reload in CONTINUOUS mode, counting always begins at the reset value of 0001H.
 3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
 4. Enable the timer interrupt (if appropriate) and set the timer interrupt priority by writing to the relevant interrupt registers.
 5. Configure the associated GPIO port pin (if using the Timer Output function) for the Timer Output alternate function.
 6. Write to the Timer Control register to enable the timer and initiate counting.

In CONTINUOUS mode, the system clock always provides the timer input. The timer period is given by the following equation:

$$\text{CONTINUOUS Mode Time-Out Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT mode equation to determine the first time-out period.

COUNTER Mode

In COUNTER mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO Port pin Timer Input alternate function. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the Timer Input signal. In COUNTER mode, the prescaler is disabled.



Caution: *The input frequency of the Timer Input signal must not exceed one-fourth the system clock frequency. Further, the high or low state of the input signal pulse must be no less than twice the system clock period. A shorter pulse may not be captured.*

Upon reaching the Reload value stored in the Timer Reload High and Low Byte registers, 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.

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.

Follow the steps below for configuring 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 SINGLE OUTPUT 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 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:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT mode equation to determine the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is represented by:

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

If TPOL is set to 1, the ratio of the PWM output High time to the total period is represented by:

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

Follow the steps below for configuring a timer for COMPARE mode and initiating 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.
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
6. Write to the Timer Control register to enable the timer and initiate counting.

In COMPARE mode, the system clock always provides the timer input. The Compare time can be calculated by the following equation:

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

GATED Mode

In GATED mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is deasserted or a timer reload occurs. To determine if a Timer Input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. When reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes (assuming the Timer Input signal remains asserted). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reset.

Follow the steps below for configuring a timer for GATED mode and initiating the count:

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for GATED mode.
 - Set the prescale value.

110 = 64 cycles delay

111 = 128 cycles delay

INPCAP—Input Capture Event

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 (Table 49).

Table 49. Timer 0–1 Control Register 1 (TxCTL1)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	TPOL	PRES			TMODE		
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F07H, F0FH							

TEN—Timer Enable

0 = Timer is disabled.

1 = Timer enabled to count.

TPOL—Timer Input/Output Polarity

Operation of this bit is a function of the current operating mode of the timer.

ONE-SHOT mode

When the timer is disabled, the Timer Output signal is set to the value of this bit.

When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

CONTINUOUS mode

When the timer is disabled, the Timer Output signal is set to the value of this bit.

When the timer is enabled, the Timer Output signal is complemented upon timer Reload.

COUNTER mode

If the timer is enabled the Timer Output signal is complemented after timer reload.

0 = Count occurs on the rising edge of the Timer Input signal.

1 = Count occurs on the falling edge of the Timer Input signal.

Analog-to-Digital Converter

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

- 11-bit resolution in DIFFERENTIAL mode.
- 10-bit resolution in SINGLE-ENDED mode.
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports.
- 9th analog input obtained from temperature sensor peripheral.
- 11 pairs of differential inputs also multiplexed with general-purpose I/O ports.
- Low-power operational amplifier (LPO).
- Interrupt on conversion complete.
- Bandgap generated internal voltage reference with two selectable levels.
- Manual in-circuit calibration is possible employing user code (offset calibration).
- Factory calibrated for in-circuit error compensation.

Architecture

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

The input stage of the ADC allows both differential gain and buffering. The following input options are available:

- Unbuffered input (SINGLE-ENDED and DIFFERENTIAL modes).
- Buffered input with unity gain (SINGLE-ENDED and DIFFERENTIAL modes).
- LPO output with full pin access to the feedback path.

#5 MSB	#5 LSB
--------	--------

6. Add the gain correction factor to the original offset corrected value.

#5 MSB	#5 LSB
--------	--------

+

#1 MSB	#1 LSB
--------	--------

=

#6 MSB	#6 LSB
--------	--------

7. Shift the result to the right, using the sign bit determined in [Step 1](#). This allows for the detection of computational overflow.

S->	#6 MSB	#6 LSB
-----	--------	--------

Output Data

The following is the output format of the corrected ADC value.

MSB								LSB							
s	v	b	a	9	8	7	6	5	4	3	2	1	0	-	-

The overflow bit in the corrected output indicates that the computed value was greater than the maximum logical value (+1023) or less than the minimum logical value (-1024). Unlike the hardware overflow bit, this is not a simple binary Flag. For a normal sample (non-overflow), the sign and the overflow bit matches. If the sign bit and overflow bit do not match, a computational overflow has occurred.

Input Buffer Stage

Many applications require the measurement of an input voltage source with a high output impedance. This ADC provides a buffered input for such situations. The drawback of the buffered input is a limitation of the input range. When using unity gain buffered mode, the input signal must be prevented from coming too close to either V_{SS} or V_{DD} . See [Table 135](#) on page 231 for details.

This condition applies only to the input voltage level (with respect to ground) of each differential input signal. The actual differential input voltage magnitude may be less than 300 mV.

The input range of the unbuffered ADC swings from V_{SS} to V_{DD} . Input signals smaller than 300 mV must use the unbuffered input mode. If these signals do not contain low output impedances, they might require off-chip buffering.

Signals outside the allowable input range can be used without instability or device damage. Any ADC readings made outside the input range are subject to greater inaccuracy than specified.

1001 = 1.8 V
1010–1111 = Reserved

For 8-pin devices:

000000 = 0.00 V
000001 = 0.05 V
000010 = 0.10 V
000011 = 0.15 V
000100 = 0.20 V
000101 = 0.25 V
000110 = 0.30 V
000111 = 0.35 V
001000 = 0.40 V
001001 = 0.45 V
001010 = 0.50 V
001011 = 0.55 V
001100 = 0.60 V
001101 = 0.65 V
001110 = 0.70 V
001111 = 0.75 V
010000 = 0.80 V
010001 = 0.85 V
010010 = 0.90 V
010011 = 0.95 V
010100 = 1.00 V (Default)
010101 = 1.05 V
010110 = 1.10 V
010111 = 1.15 V
011000 = 1.20 V
011001 = 1.25 V
011010 = 1.30 V
011011 = 1.35 V
011100 = 1.40 V
011101 = 1.45 V
011110 = 1.50 V
011111 = 1.55 V
100000 = 1.60 V
100001 = 1.65 V
100010 = 1.70 V
100011 = 1.75 V
100100 = 1.80 V

value 63H to the Flash Control register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. Using the On-Chip Debugger, poll the Flash Status register to determine when the Mass Erase operation is complete. When the Mass Erase is complete, the Flash Controller returns to its locked state.

Flash Controller Bypass

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Row Programming algorithms by controlling the Flash programming signals directly.

Row programming is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of the Flash memory. Page Erase operations are also supported when the Flash Controller is bypassed.

For more information on bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore![®] MCU Application Note (AN0117)* available for download at www.zilog.com.

Flash Controller Behavior in DEBUG Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored.
- The Flash Sector Protect register is ignored for programming and erase operations.
- Programming operations are not limited to the page selected in the Page Select register.
- Bits in the Flash Sector Protect register can be written to one or zero.
- The second write of the Page Select register to unlock the Flash Controller is not necessary.
- The Page Select register can be written when the Flash Controller is unlocked.
- The Mass Erase command is enabled through the Flash Control register.



Caution: *For security reasons, the Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the flash controller must go through the unlock sequence again to select another page.*

Flash Status Register

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

Table 79. Flash Status Register (FSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved		FSTAT					
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
ADDR	FF8H							

Reserved—Must be 0.

FSTAT—Flash Controller Status

000000 = Flash Controller locked

000001 = First unlock command received (73H written)

000010 = Second unlock command received (8CH written)

000011 = Flash Controller unlocked

000100 = Sector protect register selected

001xxx = Program operation in progress

010xxx = Page erase operation in progress

100xxx = Mass erase operation in progress

Flash Page Select Register

The Flash Page Select (FPS) register shares address space with the Flash Sector Protect Register. Unless the Flash controller is unlocked and written with 5EH, writes to this address target the Flash Page Select Register.

The register is used to select one of the available Flash memory pages to be programmed or erased. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory having addresses with the most significant 7 bits given by FPS[6:0] are chosen for program/erase operation.

Flash Option Bits

Programmable Flash option bits allow user configuration of certain aspects of Z8 Encore! XP[®] F082A Series operation. The feature configuration data is stored in the Flash program memory and loaded into holding registers during Reset. The features available for control through the Flash Option Bits include:

- Watchdog Timer time-out response selection—interrupt or system reset
- Watchdog Timer always on (enabled at Reset)
- The ability to prevent unwanted read access to user code in Program Memory
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Program Memory
- Voltage Brownout configuration—always enabled or disabled during STOP mode to reduce STOP mode power consumption
- Oscillator mode selection—for high, medium, and low power crystal oscillators, or external RC oscillator
- Factory trimming information for the internal precision oscillator and low voltage detection
- Factory calibration values for ADC, temperature sensor, and Watchdog Timer compensation
- Factory serialization and randomized lot identifier (optional)

Operation

Option Bit Configuration By Reset

Each time the Flash Option Bits are programmed or erased, the device must be Reset for the change to take effect. During any reset operation (System Reset, Power-On Reset, or Stop Mode Recovery), the Flash Option Bits are automatically read from the Flash Program Memory and written to Option Configuration registers. The Option Configuration registers control operation of the devices within the Z8 Encore! XP F082A Series. Option Bit control is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access.

When selecting a new clock source, the system clock oscillator failure detection circuitry and the Watchdog Timer oscillator failure circuitry must be disabled. If SOFEN and WOFEN are not disabled prior to a clock switch-over, it is possible to generate an interrupt for a failure of either oscillator. The Failure detection circuitry can be enabled any-time after a successful write of OSCSEL in the OSCCTL register.

The internal precision oscillator is enabled by default. If the user code changes to a different oscillator, it may be appropriate to disable the IPO for power savings. Disabling the IPO does not occur automatically.

Clock Failure Detection and Recovery

System Clock Oscillator Failure

The Z8F04xA family devices can generate non-maskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer oscillator to drive the system clock. The Watchdog Timer oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer is selected as the system clock oscillator. It is also unavailable if the Watchdog Timer oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function (see [Watchdog Timer](#) on page 91).

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below $1 \text{ kHz} \pm 50\%$. If an external signal is selected as the system oscillator, it is possible that a very slow but non-failing clock can generate a failure condition. Under these conditions, do not enable the clock failure circuitry (SOFEN must be deasserted in the OSCCTL register).

Watchdog Timer Failure

In the event of a Watchdog Timer oscillator failure, a similar non-maskable interrupt-like event is issued. This event does not trigger an attendant clock switch-over, but alerts the CPU of the failure. After a Watchdog Timer failure, it is no longer possible to detect a primary oscillator failure. The failure detection circuitry does not function if the Watchdog Timer is used as the system clock oscillator or if the Watchdog Timer oscillator has been disabled. For either of these cases, it is necessary to disable the detection circuitry by deasserting the WDFEN bit of the OSCCTL register.

The Watchdog Timer oscillator failure detection circuit counts system clocks while looking for a Watchdog Timer clock. The logic counts 8004 system clock cycles before determining that a failure has occurred. The system clock rate determines the speed at which the Watchdog Timer failure can be detected. A very slow system clock results in very slow detection times.

Crystal Oscillator

The products in the Z8 Encore! XP[®] F082A Series contain an on-chip crystal oscillator for use with external crystals with 32 kHz to 20 MHz frequencies. In addition, the oscillator supports external RC networks with oscillation frequencies up to 4 MHz or ceramic resonators with frequencies up to 8 MHz. The on-chip crystal oscillator can be used to generate the primary system clock for the internal eZ8 CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin can also accept a CMOS-level clock input signal (32 kHz–20 MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected. The Z8 Encore! XP F082A Series products do not contain an internal clock divider. The frequency of the signal on the X_{IN} input pin determines the frequency of the system clock.

► **Note:** *Although the XIN pin can be used as an input for an external clock generator, the CLKIN pin is better suited for such use (see [System Clock Selection](#) on page 187).*

Operating Modes

The Z8 Encore! XP F082A Series products support four oscillator modes:

- Minimum power for use with very low frequency crystals (32 kHz–1 MHz).
- Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 8 MHz).
- Maximum power for use with high frequency crystals (8 MHz to 20 MHz).
- On-chip oscillator configured for use with external RC networks (<4 MHz).

The oscillator mode is selected using user-programmable Flash Option Bits. See [Flash Option Bits](#) on page 153 for information.

Crystal Oscillator Operation

The Flash Option bit XTLDIS controls whether the crystal oscillator is enabled during reset. The crystal may later be disabled after reset if a new oscillator has been selected as the system clock. If the crystal is manually enabled after reset through the OSCCTL register, the user code must wait at least 1000 crystal oscillator cycles for the crystal to stabilize. After this, the crystal oscillator may be selected as the system clock.

► **Note:** *The stabilization time varies depending on the crystal or resonator used, as well as on the feedback network. See [Table 111](#) for transconductance values to compute oscillator stabilization times.*

Table 121. Logical Instructions (Continued)

Mnemonic	Operands	Instruction
ORX	dst, src	Logical OR using Extended Addressing
XOR	dst, src	Logical Exclusive OR
XORX	dst, src	Logical Exclusive OR using Extended Addressing

Table 122. Program Control Instructions

Mnemonic	Operands	Instruction
BRK	—	On-Chip Debugger Break
BTJ	p, bit, src, DA	Bit Test and Jump
BTJNZ	bit, src, DA	Bit Test and Jump if Non-Zero
BTJZ	bit, src, DA	Bit Test and Jump if Zero
CALL	dst	Call Procedure
DJNZ	dst, src, RA	Decrement and Jump Non-Zero
IRET	—	Interrupt Return
JP	dst	Jump
JP cc	dst	Jump Conditional
JR	DA	Jump Relative
JR cc	DA	Jump Relative Conditional
RET	—	Return
TRAP	vector	Software Trap

Table 123. Rotate and Shift Instructions

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
RR	dst	Rotate Right
RRC	dst	Rotate Right through Carry

Table 124. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
AND dst, src	$\text{dst} \leftarrow \text{dst AND src}$	r	r	52	–	*	*	0	–	–	2	3
		r	lr	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	$\text{dst} \leftarrow \text{dst AND src}$	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	$\text{dst}[\text{bit}] \leftarrow 0$	r		E2	–	–	–	–	–	–	2	2
BIT p, bit, dst	$\text{dst}[\text{bit}] \leftarrow p$	r		E2	–	–	–	–	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	$\text{dst}[\text{bit}] \leftarrow 1$	r		E2	–	–	–	–	–	–	2	2
BSWAP dst	$\text{dst}[7:0] \leftarrow \text{dst}[0:7]$	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if $\text{src}[\text{bit}] = p$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJNZ bit, src, dst	if $\text{src}[\text{bit}] = 1$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJZ bit, src, dst	if $\text{src}[\text{bit}] = 0$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
CALL dst	$\text{SP} \leftarrow \text{SP} - 2$ $@\text{SP} \leftarrow \text{PC}$ $\text{PC} \leftarrow \text{dst}$	IRR		D4	–	–	–	–	–	–	2	6
		DA		D6							3	3
CCF	$C \leftarrow \sim C$			EF	*	–	–	–	–	–	1	2
CLR dst	$\text{dst} \leftarrow 00H$	R		B0	–	–	–	–	–	–	2	2
		IR		B1							2	3
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							

On-Chip Debugger Timing

Figure 36 and Table 141 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4 ns maximum rise and fall time.

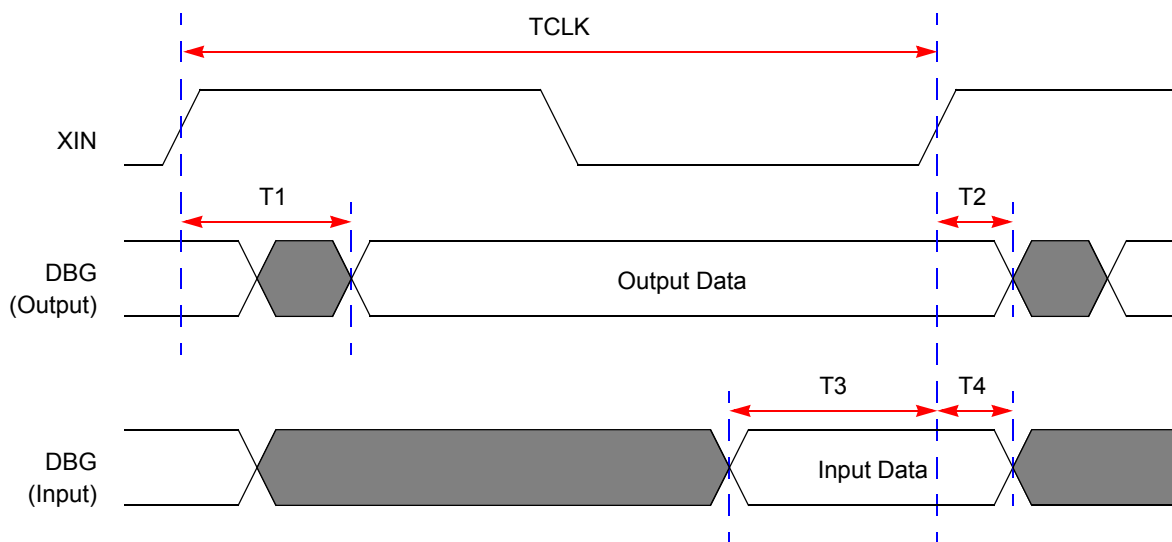


Figure 36. On-Chip Debugger Timing

Table 141. On-Chip Debugger Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
DBG			
T ₁	XIN Rise to DBG Valid Delay	–	15
T ₂	XIN Rise to DBG Output Hold Time	2	–
T ₃	DBG to XIN Rise Input Setup Time	5	–
T ₄	DBG to XIN Rise Input Hold Time	5	–

page erase 147
page select register 150, 151
FPS register 150, 151
FSTAT register 150

G

GATED mode 85
general-purpose I/O 37
GPIO 7, 37
 alternate functions 38
 architecture 38
 control register definitions 45
 input data sample timing 234
 interrupts 45
 port A-C pull-up enable sub-registers 50, 51
 port A-H address registers 46
 port A-H alternate function sub-registers 47
 port A-H control registers 46
 port A-H data direction sub-registers 47
 port A-H high drive enable sub-registers 49
 port A-H input data registers 51
 port A-H output control sub-registers 48
 port A-H output data registers 52
 port A-H stop mode recovery sub-registers 49
 port availability by device 37
 port input timing 235
 port output timing 236

H

H 202
HALT 204
halt mode 34, 204
hexadecimal number prefix/suffix 202

I

I²C 7
IM 201
immediate data 201
immediate operand prefix 202

INC 203
increment 203
increment word 203
INCW 203
indexed 201
indirect address prefix 202
indirect register 201
indirect register pair 201
indirect working register 201
indirect working register pair 201
infrared encoder/decoder (IrDA) 117
Instruction Set 199
instruction set, eZ8 CPU 199
instructions
 ADC 203
 ADCX 203
 ADD 203
 ADDX 203
 AND 205
 ANDX 205
 arithmetic 203
 BCLR 204
 BIT 204
 bit manipulation 204
 block transfer 204
 BRK 206
 BSET 204
 BSWAP 204, 206
 BTJ 206
 BTJNZ 206
 BTJZ 206
 CALL 206
 CCF 204
 CLR 205
 COM 205
 CP 203
 CPC 203
 CPCX 203
 CPU control 204
 CPX 203
 DA 203
 DEC 203
 DECW 203
 DI 204