

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	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	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f012apb020sc

Watchdog Timer Time-Out Response	92
Watchdog Timer Reload Unlock Sequence	93
Watchdog Timer Calibration	93
Watchdog Timer Control Register Definitions	94
Watchdog Timer Control Register	94
Watchdog Timer Reload Upper, High and Low Byte Registers	94
Universal Asynchronous Receiver/Transmitter	97
Architecture	97
Operation	98
Data Format	98
Transmitting Data using the Polled Method	99
Transmitting Data using the Interrupt-Driven Method	100
Receiving Data using the Polled Method	101
Receiving Data using the Interrupt-Driven Method	102
Clear To Send (CTS) Operation	103
MULTIPROCESSOR (9-bit) Mode	103
External Driver Enable	104
UART Interrupts	105
UART Baud Rate Generator	107
UART Control Register Definitions	108
UART Control 0 and Control 1 Registers	108
UART Status 0 Register	111
UART Status 1 Register	112
UART Transmit Data Register	113
UART Receive Data Register	113
UART Address Compare Register	114
UART Baud Rate High and Low Byte Registers	114
Infrared Encoder/Decoder	117
Architecture	117
Operation	117
Transmitting IrDA Data	118
Receiving IrDA Data	119
Infrared Encoder/Decoder Control Register Definitions	120
Analog-to-Digital Converter	121
Architecture	121
Operation	122
Data Format	122

Table 3. Pin Characteristics (20- and 28-pin Devices) (Continued)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt-Trigger Input	Open Drain Output	5 V Tolerance
PC[7:0]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	PC[7:3] unless pullups enabled
RESET/PD0	I/O	I/O (defaults to RESET)	Low (in Reset mode)	Yes (PD0 only)	Programmable for PD0; always on for RESET	Yes	Programmable for PD0; always on for RESET	Yes, unless pullups enabled
VDD	N/A	N/A	N/A	N/A			N/A	N/A
VSS	N/A	N/A	N/A	N/A			N/A	N/A

► **Note:** PB6 and PB7 are available only in those devices without ADC.

Table 4. Pin Characteristics (8-Pin Devices)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt-Trigger Input	Open Drain Output	5 V Tolerance
PA0/DBG	I/O	I (but can change during reset if key sequence detected)	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
PA1	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
RESET/PA2	I/O	I/O (defaults to RESET)	Low (in Reset mode)	Yes	Programmable for PA2; always on for RESET	Yes	Programmable for PA2; always on for RESET	Yes, unless pull-ups enabled
PA[5:3]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
V _{DD}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
V _{SS}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 18. Port A–D Control Registers (PxCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	PCTL							
RESET	00H							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FD1H, FD5H, FD9H, FDDH							

PCTL[7:0]—Port Control

The Port Control register provides access to all sub-registers that configure the GPIO Port operation.

Port A–D Data Direction Sub-Registers

The Port A–D Data Direction sub-register is accessed through the Port A–D Control register by writing 01H to the Port A–D Address register ([Table 19](#)).

Table 19. Port A–D Data Direction Sub-Registers (PxDD)

BITS	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
ADDR	If 01H in Port A–D Address Register, accessible through the Port A–D Control Register							

DD[7:0]—Data Direction

These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction register setting.

0 = Output. Data in the Port A–D Output Data register is driven onto the port pin.

1 = Input. The port pin is sampled and the value written into the Port A–D Input Data Register. The output driver is tristated.

Port A–D Alternate Function Sub-Registers

The Port A–D Alternate Function sub-register ([Table 20](#)) is accessed through the Port A–D Control register by writing 02H to the Port A–D Address register. The Port A–D Alternate Function sub-registers enable the alternate function selection on pins. If disabled, pins functions as GPIO. If enabled, select one of four alternate functions using alternate function set subregisters 1 and 2 as described in the [Port A–D Alternate Function](#)

Table 39. IRQ1 Enable and Priority Encoding

IRQ1ENH[x]	IRQ1ENL[x]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Medium
1	1	Level 3	High

where x indicates the register bits from 0–7.

Table 40. IRQ1 Enable High Bit Register (IRQ1ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VENH	PA6CENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
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	FC4H							

PA7VENH—Port A Bit[7] or LVD Interrupt Request Enable High Bit

PA6CENH—Port A Bit[7] or Comparator Interrupt Request Enable High Bit

PAxENH—Port A Bit[x] Interrupt Request Enable High Bit

See Shared Interrupt Select (IRQSS) register for selection of either the LVD or the comparator as the interrupt source.

Table 41. IRQ1 Enable Low Bit Register (IRQ1ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
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	FC5H							

PA7VENL—Port A Bit[7] or LVD Interrupt Request Enable Low Bit

PA6CENL—Port A Bit[6] or Comparator Interrupt Request Enable Low Bit

PAxENL—Port A Bit[x] Interrupt Request Enable Low Bit

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

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for COMPARATOR COUNTER mode.
 - Select either the rising edge or falling edge of the comparator output signal for the count. This also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function is not required to be enabled.
2. Write to the Timer High and Low Byte registers to set the starting count value. This action only affects the first pass in COMPARATOR COUNTER mode. After the first timer Reload in COMPARATOR COUNTER mode, counting always begins at the reset value of 0001H. Generally, in COMPARATOR COUNTER mode the Timer High and Low Byte registers must be written with the value 0001H.
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.

In COMPARATOR COUNTER mode, the number of comparator output transitions since the timer start is given by the following equation:

$$\text{Comparator Output Transitions} = \text{Current Count Value} - \text{Start Value}$$

PWM SINGLE OUTPUT Mode

In PWM SINGLE OUTPUT mode, the timer outputs a Pulse-Width Modulator (PWM) output signal through a GPIO Port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

If the TPOL bit in the Timer Control register is set to 1, the Timer Output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the Reload value and is reset to 0001H.

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

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for CAPTURE mode.
 - Set the prescale value.
 - Set the Capture edge (rising or falling) for the Timer Input.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. Clear the Timer PWM High and Low Byte registers to 0000H. Clearing these registers allows the software to determine if interrupts were generated by either a capture event or a reload. 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 TxCTL0 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:

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

CAPTURE RESTART Mode

In CAPTURE RESTART mode, the current timer count value is recorded when the acceptable external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in TxCTL0 register is set to indicate the timer interrupt is because of an input capture event.

If no Capture event occurs, the timer counts up to the 16-bit Compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the 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. The INPCAP bit in TxCTL0 register is cleared to indicate the timer interrupt is not caused by an input capture event.

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

1. Write to the Timer Control register to:
 - Disable the timer.
 - Configure the timer for CAPTURE RESTART mode by writing the TMODE bits in the TxCTL1 register and the TMODEHI bit in TxCTL0 register.
 - Set the prescale value.
 - Set the Capture edge (rising or falling) for the Timer Input.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. Clear the Timer PWM High and Low Byte registers to 0000H. This allows the software to determine if interrupts were generated by either a capture event or a reload. 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 TxCTL0 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.

MPMD[1:0]—MULTIPROCESSOR Mode

If MULTIPROCESSOR (9-bit) mode is enabled,

00 = The UART generates an interrupt request on all received bytes (data and address).

01 = The UART generates an interrupt request only on received address bytes.

10 = The UART generates an interrupt request when a received address byte matches the value stored in the Address Compare Register and on all successive data bytes until an address mismatch occurs.

11 = The UART generates an interrupt request on all received data bytes for which the most recent address byte matched the value in the Address Compare Register.

MPEN—MULTIPROCESSOR (9-bit) Enable

This bit is used to enable MULTIPROCESSOR (9-bit) mode.

0 = Disable MULTIPROCESSOR (9-bit) mode.

1 = Enable MULTIPROCESSOR (9-bit) mode.

MPBT—Multiprocessor Bit Transmit

This bit is applicable only when MULTIPROCESSOR (9-bit) mode is enabled. The 9th bit is used by the receiving device to determine if the data byte contains address or data information.

0 = Send a 0 in the multiprocessor bit location of the data stream (data byte).

1 = Send a 1 in the multiprocessor bit location of the data stream (address byte).

DEPOL—Driver Enable Polarity

0 = DE signal is Active High.

1 = DE signal is Active Low.

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.

$\overline{\text{RDAIRQ}}$ —Receive Data Interrupt Enable

0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller.

Table 73. ADC Data High Byte Register (ADCD_H)

BITS	7	6	5	4	3	2	1	0
FIELD	ADCDH							
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	F72H							
X = Undefined.								

ADCDH—ADC Data High Byte

This byte contains the upper eight bits of the ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the most recent conversion output is held in this register. These bits are undefined after a Reset.

ADC Data Low Byte Register

The ADC Data Low Byte (ADCD_L) register contains the lower bits of the ADC output as well as an overflow status bit. The output is a 13-bit two's complement value. During a single-shot conversion, this value is invalid. Access to the ADC Data Low Byte register is read-only. Reading the ADC Data High Byte register latches data in the ADC Low Bits register.

Table 74. ADC Data Low Byte Register (ADCD_L)

BITS	7	6	5	4	3	2	1	0
FIELD	ADCDL					Reserved		OVF
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	F73H							
X = Undefined.								

ADCDL—ADC Data Low Bits

These bits are the least significant five bits of the 13-bits of the ADC output. These bits are undefined after a Reset.

Reserved—Must be undefined.

OVF—Overflow Status

0= A hardware overflow did not occur in the ADC for the current sample.

1= A hardware overflow did occur in the ADC for the current sample, therefore the current sample is invalid.

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

Temperature Sensor

The on-chip Temperature Sensor allows you to measure temperature on the die with either the on-board ADC or on-board comparator. This block is factory calibrated for in-circuit software correction. Uncalibrated accuracy is significantly worse, therefore the temperature sensor is not recommended for uncalibrated use.

Temperature Sensor Operation

The on-chip temperature sensor is a Proportional to Absolute Temperature (PTAT) topology. A pair of Flash option bytes contain the calibration data. The temperature sensor can be disabled by a bit in the [Power Control Register 0](#) on page 34 to reduce power consumption.

The temperature sensor can be directly read by the ADC to determine the absolute value of its output. The temperature sensor output is also available as an input to the comparator for threshold type measurement determination. The accuracy of the sensor when used with the comparator is substantially less than when measured by the ADC.

If the temperature sensor is routed to the ADC, the ADC must be configured in unity-gain buffered mode (see [Input Buffer Stage](#) on page 129). The value read back from the ADC is a signed number, although it is always positive.

The sensor is factory-trimmed through the ADC using the external 2.0 V reference. Unless the sensor is re-trimmed for use with a different reference, it is most accurate when used with the external 2.0 V reference.

Because this sensor is an on-chip sensor it is recommended that the user account for the difference between ambient and die temperature when inferring ambient temperature conditions.

During normal operation, the die undergoes heating that causes a mismatch between the ambient temperature and that measured by the sensor. For best results, the Z8 Encore! XP[®] device must be placed into STOP mode for sufficient time such that the die and ambient temperatures converge (this time is dependent on the thermal design of the system). The temperature sensor measurement must then be made immediately after recovery from STOP mode.

The following equation defines the transfer function between the temperature sensor output voltage and the die temperature. This is needed for comparator threshold measurements.

$$V = 0.01 \times T + 0.65$$

where, T is the temperature in °C; V is the sensor output in volts.

High. Because of the open-drain nature of the DBG pin, the host can send a Serial Break to the OCD even if the OCD is transmitting a character.

OCD Unlock Sequence (8-Pin Devices Only)

Because of pin-sharing on the 8-pin device, an unlock sequence must be performed to access the DBG pin. If this sequence is not completed during a system reset, then the PA0/DBG pin functions only as a GPIO pin.

The following sequence unlocks the DBG pin:

1. Hold PA2/ $\overline{\text{RESET}}$ Low.
2. Wait 5ms for the internal reset sequence to complete.
3. Send the following bytes serially to the debug pin:
DBG \leftarrow 80H (autobaud)
DBG \leftarrow EBH
DBG \leftarrow 5AH
DBG \leftarrow 70H
DBG \leftarrow CDH (32-bit unlock key)
4. Release PA2/ $\overline{\text{RESET}}$. The PA0/DBG pin is now identical in function to that of the DBG pin on the 20-/28-pin device. To enter DEBUG mode, re-autobaud and write 80H to the OCD control register (see [On-Chip Debugger Commands](#) on page 179).



Caution: *Between [Step 3](#) and [Step 4](#), there is an interval during which the 8-pin device is neither in RESET nor DEBUG mode. If a device has been erased or has not yet been programmed, all program memory bytes contain FFH. The CPU interprets this as an illegal instruction, so some irregular behavior can occur before entering DEBUG mode, and the register values after entering DEBUG mode differs from their specified reset values. However, none of these irregularities prevent programming the Flash memory. Before beginning system debug, it is recommended that some legal code be programmed into the 8-pin device, and that a RESET occurs.*

Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. If Breakpoints are enabled, the OCD enters DEBUG mode and idles the eZ8 CPU. If Breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

Write Memory, Read Memory, Write Register, Read Register, Read Memory CRC, Step Instruction, Stuff Instruction, and Execute Instruction commands.

```
DBG ← 03H
DBG → RuntimeCounter[15:8]
DBG → RuntimeCounter[7:0]
```

- **Write OCD Control Register (04H)**—The Write OCD Control Register command writes the data that follows to the OCDCTL register. When the Flash Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of returning the device to normal operating mode is to reset the device.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

- **Read OCD Control Register (05H)**—The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

- **Write Program Counter (06H)**—The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, the Program Counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

- **Read Program Counter (07H)**—The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

- **Write Register (08H)**—The Write Register command writes data to the Register File. Data can be written 1–256 bytes at a time (256 bytes can be written by setting size to 0). If the device is not in DEBUG mode, the address and data values are discarded. If the Flash Read Protect Option bit is enabled, only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0, Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```

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 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		
SUBX dst, src	$\text{dst} \leftarrow \text{dst} - \text{src}$	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29							4	3
SWAP dst	$\text{dst}[7:4] \leftrightarrow \text{dst}[3:0]$	R		F0	X	*	*	X	-	-	2	2
		IR		F1							2	3
TCM dst, src	(NOT dst) AND src	r	r	62	-	*	*	0	-	-	2	3
		r	lr	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	-	*	*	0	-	-	4	3
		ER	IM	69							4	3
TM dst, src	dst AND src	r	r	72	-	*	*	0	-	-	2	3
		r	lr	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	-	*	*	0	-	-	4	3
		ER	IM	79							4	3
TRAP Vector	$\text{SP} \leftarrow \text{SP} - 2$ $\text{@SP} \leftarrow \text{PC}$ $\text{SP} \leftarrow \text{SP} - 1$ $\text{@SP} \leftarrow \text{FLAGS}$ $\text{PC} \leftarrow \text{@Vector}$		Vector	F2	-	-	-	-	-	-	2	6
WDT				5F	-	-	-	-	-	-	1	2
Flags Notation:	* = Value is a function of the result of the operation. - = Unaffected X = Undefined											
					0 = Reset to 0 1 = Set to 1							

General Purpose I/O Port Output Timing

Figure 35 and Table 140 provide timing information for GPIO Port pins.

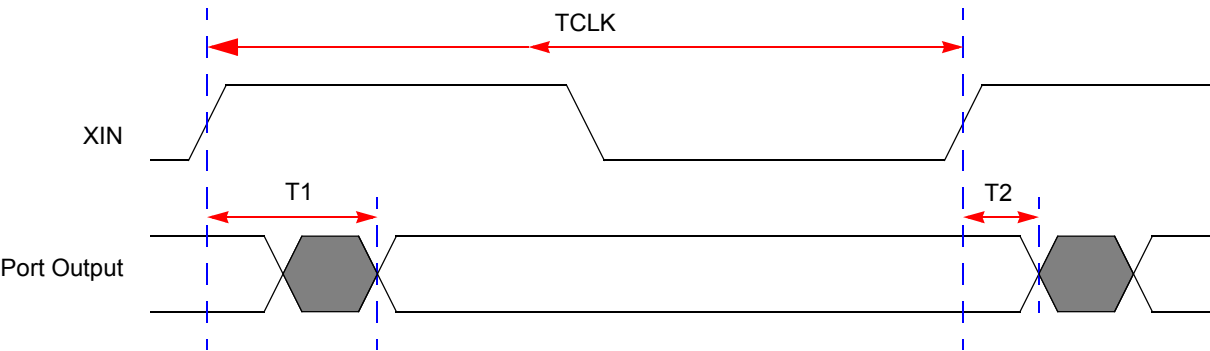


Figure 35. GPIO Port Output Timing

Table 140. GPIO Port Output Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
GPIO Port pins			
T ₁	XIN Rise to Port Output Valid Delay	–	15
T ₂	XIN Rise to Port Output Hold Time	2	–

Figure 43 displays the 20-pin Small Outline Integrated Circuit Package (SOIC) available for the Z8 Encore! XP F082A Series devices.

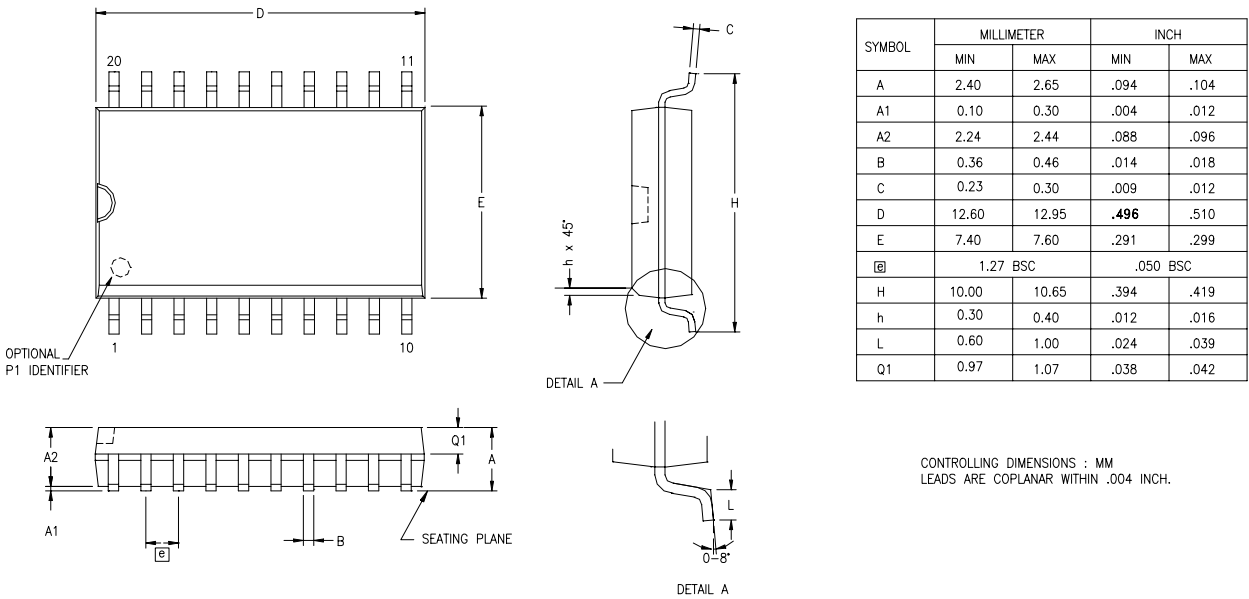


Figure 43. 20-Pin Small Outline Integrated Circuit Package (SOIC)

Index

Symbols

202
% 202
@ 202

Numerics

10-bit ADC 7
40-lead plastic dual-inline package 248, 249

A

absolute maximum ratings 221
AC characteristics 227
ADC 203
 architecture 121
 automatic power-down 122
 block diagram 122
 continuous conversion 124
 control register 130, 132
 control register definitions 130
 data high byte register 132
 data low bits register 133
 electrical characteristics and timing 231
 operation 122
 single-shot conversion 123
ADCCTL register 130, 132
ADCDH register 132
ADCDL register 133
ADCX 203
ADD 203
add - extended addressing 203
add with carry 203
add with carry - extended addressing 203
additional symbols 202
address space 15
ADDX 203
analog signals 12
analog-to-digital converter (ADC) 121
AND 205

ANDX 205
arithmetic instructions 203
assembly language programming 199
assembly language syntax 200

B

B 202
b 201
baud rate generator, UART 107
BCLR 204
binary number suffix 202
BIT 204
bit 201
 clear 204
 manipulation instructions 204
 set 204
 set or clear 204
 swap 204
 test and jump 206
 test and jump if non-zero 206
 test and jump if zero 206
bit jump and test if non-zero 206
bit swap 206
block diagram 4
block transfer instructions 204
BRK 206
BSET 204
BSWAP 204, 206
BTJ 206
BTJNZ 206
BTJZ 206

C

CALL procedure 206
CAPTURE mode 85, 86
CAPTURE/COMPARE mode 85
cc 201
CCF 204
characteristics, electrical 221
clear 205
CLR 205
COM 205

