



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	5MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	16
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0223ph005ec

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Table of Contents

Overview	. 1
Features	. 1
Part Selection Guide	. 2
Block Diagram	. 3
CPU and Peripheral Overview	. 4
eZ8 CPU Features	. 4
General-Purpose I/O	. 4
Flash Controller	. 4
Internal Precision Oscillator	. 5
	. 5 5
Universal Asynchronous Receiver/Transmitter	. 5
Timers	. 5
Interrupt Controller	. 5
Reset Controller	. 6
On-Chip Debugger	. 6
Pin Description	. 7
Available Packages	. 7
Pin Configurations	. 7
Signal Descriptions	. 9
Pin Characteristics	10
Address Space	13
Register File	13
Program Memory	13
Data Memory	15
Flash Information Area	15
Register Map	17
Reset and Stop Mode Recovery	21
Reset Types	21
Reset Sources	22
Power-On Reset	23
Voltage Brownout Reset	24
	25
External Reset Indicator	20 26
	20

Z8 Encore! XP[®] F0823 Series Product Specification

PA0 and PA6 contain two different timer functions, a timer input and a complementary timer output. Both of these functions require the same GPIO configuration, the selection between the two is based on the timer mode. For more details, see Timers on page 67.

Caution: For pin with multiple alternate functions, it is recommended to write to the AFS1 and AFS2 sub-registers before enabling the alternate function via the AF sub-register. This prevents spurious transitions through unwanted alternate function modes.

Direct LED Drive

The Port C pins provide a current sinked output capable of driving an LED without requiring an external resistor. The output sinks current at programmable levels of 3 mA, 7 mA, 13 mA, and 20 mA. This mode is enabled through the Alternate Function sub-register AFS1 and is programmable through the LED control registers. The LED Drive Enable (LEDEN) register turns on the drivers. The LED Drive Level (LEDLVLH and LEDLVLL) registers select the sink current.

For correct function, the LED anode must be connected to V_{DD} and the cathode to the GPIO pin. Using all Port C pins in LED drive mode with maximum current can result in excessive total current. For the maximum total current for the applicable package, see Electrical Characteristics on page 193.

Shared Reset Pin

On the 8-pin product versions, the reset pin is shared with PA2, but the pin is not limited to output-only when in GPIO mode.

Caution: If PA2 on the 8-pin product is reconfigured as an input, take care that no external stimulus drives the pin Low during any reset sequence. Since PA2 returns to its RESET alternate function during system resets, driving it Low holds the chip in a reset state until the pin is released.

Shared Debug Pin

On the 8-pin version of this device only, the Debug pin shares function with the PA0 GPIO pin. This pin performs as a general purpose input pin on power-up, but the debug logic monitors this pin during the reset sequence to determine if the unlock sequence occurs. If the unlock sequence is present, the debug function is unlocked and the pin no longer func-

PSMRE[7:0]—Port Stop Mode Recovery Source Enabled.

0 = The Port pin is not configured as a Stop Mode Recovery source. Transitions on this pin during STOP mode do not initiate Stop Mode Recovery.

1 = The Port pin is configured as a Stop Mode Recovery source. Any logic transition on this pin during STOP mode initiates Stop Mode Recovery.

Port A–C Pull-up Enable Sub-Registers

The Port A–C Pull-up Enable sub-register (Table 25) is accessed through the Port A–C Control register by writing 06H to the Port A–C Address register. Setting the bits in the Port A–C Pull-up Enable sub-registers enables a weak internal resistive pull-up on the specified Port pins.

Table 25. Port A–C Pull-Up Enable Sub-Registers (PxPUE)

BITS	7	6	5	4	3	2	1	0
FIELD	PPUE7	PPUE6	PPUE5	PPUE4	PPUE3	PPUE2	PPUE1	PPUE0
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	lf 06H i	n Port A–C	Address Reg	gister, acces	sible throug	h the Port A-	-C Control F	Register

PPUE[7:0]—Port Pull-up Enabled

0 = The weak pull-up on the Port pin is disabled.

1 = The weak pull-up on the Port pin is enabled.

Port A–C Alternate Function Set 1 Sub-Registers

The Port A–C Alternate Function Set1 sub-register (Table 26) is accessed through the Port A–C Control register by writing 07H to the Port A–C Address register. The Alternate Function Set 1 sub-registers selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register are defined in GPIO Alternate Functions on page 36.

Note:

Alternate function selection on port pins must also be enabled as described in Port A–C Alternate Function Sub-Registers *on page 45*.

50

0 PIN0 X R

			j	(*****)			
BITS	7	6	5	4	3	2	1
FIELD	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1
RESET	Х	Х	Х	Х	Х	Х	Х
R/W	R	R	R	R	R	R	R

Table 28. Port A–C Input Data Registers (PxIN)

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–C Output Data Register

The Port A–C Output Data register (Table 29) controls the output data to the pins.

FD2H, FD6H, FDAH

Table 29. Port A–C 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							

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 30) activates the controlled current drive. The Port C pin must first be enabled by setting the Alternate Function register to select the LED function.

ADDR

Priority	Program Memory Vector Address	Interrupt or Trap Source
Lowest	0036H	Port C Pin 0, both input edges
	0038H	Reserved

Table 33. Trap and Interrupt Vectors in Order of Priority (Continued)

Architecture

Figure 8 displays the interrupt controller block diagram.



Figure 8. Interrupt Controller Block Diagram

Operation

Master Interrupt Enable

The master interrupt enable bit (IRQE) in the Interrupt Control register globally enables and disables interrupts.

Interrupts are globally enabled by any of the following actions:

- Execution of an Enable Interrupt (EI) instruction
- Execution of an Return from Interrupt (IRET) instruction

Caution:

To avoid missing interrupts, use the following coding style to clear bits in the Interrupt Request 0 register:

Good coding style that avoids lost interrupt requests: ANDX IRQ0, MASK

Software Interrupt Assertion

Program code generates interrupts directly. Writing a 1 to the correct bit in the Interrupt Request register triggers an interrupt (assuming that interrupt is enabled). When the interrupt request is acknowledged by the eZ8 CPU, the bit in the Interrupt Request register is automatically cleared to 0.

Caution: The following coding style used to generate software interrupts by setting bits in the Interrupt Request registers is not recommended. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost.

Poor coding style that can result in lost interrupt requests: LDX r0, IRQ0 OR r0, MASK LDX IRQ0, r0

Caution: To avoid missing interrupts, use the following coding style to set bits in the Interrupt Request registers:

Good coding style that avoids lost interrupt requests: ORX IRQ0, MASK

Watchdog Timer Interrupt Assertion

The Watchdog Timer interrupt behavior is different from interrupts generated by other sources. The Watchdog Timer continues to assert an interrupt as long as the timeout condition continues. As it operates on a different (and usually slower) clock domain than the rest of the device, the Watchdog Timer continues to assert this interrupt for many system clocks until the counter rolls over.

Caution: To avoid re-triggerings of the Watchdog Timer interrupt after exiting the associated interrupt service routine, it is recommended that the service routine continues to read from the RSTSTAT register until the WDT bit is cleared as given in the following coding sample:

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

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 given by the following equation:

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

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.

Follow the steps below to configure a timer for CONTINUOUS mode and to initiate the count:

- 1. Write to the Timer Control register to:
 - Disable the timer
 - Configure the timer for CONTINUOUS mode
 - 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:

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.

- 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 PWM Control register to set the PWM dead band delay value. The deadband delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM high and low byte registers). It must also be less than the duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).
- 5. 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.
- 6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
- 8. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation: $PWM \text{ Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the ONE-SHOT mode equation determines 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:

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:

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

CAPTURE Mode

In CAPTURE mode, the current timer count value is recorded when the appropriate external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control register determines if the Capture occurs on a rising edge or a falling edge

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

Follow the steps below 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.
- 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].
- 7. Return to step 4 to receive additional data.

Z8 Encore! XP[®] F0823 Series Product Specification

118



Figure 19. Analog-to-Digital Converter Block Diagram

Operation

Data Format

The output of the ADC is an 11-bit, signed, two's complement digital value. The output generally ranges from 0 to +1023, but offset errors can cause small negative values.

The ADC registers return 13 bits of data, but the two LSBs are intended for compensation use only. When the compensation routine is performed on the 13 bit raw ADC value, two

Flash Control Register Definitions

Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FTCTL) register before programming or erasing the Flash memory. Writing the sequence 73H 8CH, sequentially, to the Flash Control register unlocks the Flash Controller. When the Flash Controller is unlocked, the Flash memory can be enabled for Mass Erase or Page Erase by writing the appropriate enable command to the FCTL. Page Erase applies only to the active page selected in Flash Page Select register. Mass Erase is enabled only through the On-Chip Debugger. Writing an invalid value or an invalid sequence returns the Flash Controller to its locked state. The Write-only Flash Control Register shares its Register File address with the read-only Flash Status Register.

BITS	7	6	5	4	3	2	1	0	
FIELD	FCMD								
RESET	0	0	0	0	0	0	0	0	
R/W	W	W	W	W	W	W	W	W	
ADDR				FF	8H				

Table 79. Flash Control Register (FCTL)

FCMD—Flash Command

73H = First unlock command

8CH = Second unlock command

95H = Page Erase command (must be third command in sequence to initiate Page Erase) 63H = Mass Erase command (must be third command in sequence to initiate Mass Erase)

5EH = Enable Flash Sector Protect Register Access

Flash Status Register

The Flash Status 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.

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

Trim Bit Data Register

The Trim Bid Data (TRMDR) register contains the read or write data for access to the trim option bits.

Table 86. Trim Bit Data Register (TRMDR)

BITS	7	6	5	4	3	2	1	0	
FIELD	TRMDR - Trim Bit Data								
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		FF7H							

Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses 0000H and 0001H are reserved for the user-programmable Flash option bits.

Flash Program Memory Address 0000H

 Table 87. Flash Option Bits at Program Memory Address 0000H

BITS	7	6	5	4	3	2	1	0	
FIELD	WDT_RES	WDT_AO	Reserved		VBO_AO	FRP	Reserved	FWP	
RESET	U	U	U	U	U	U	U	U	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
ADDR	Program Memory 0000H								
Note: U = Unchanged by Reset. R/W = Read/Write.									

WDT RES—Watchdog Timer Reset

0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request.

1 = Watchdog Timer time-out causes a system reset. This setting is the default for unprogrammed (erased) Flash.

WDT_AO—Watchdog Timer Always ON

0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled.

1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the

On-Chip Debugger Commands

The host communicates to the OCD by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG mode, all OCD commands become available unless the user code and control registers are protected by programming the Flash Read Protect Option bit (FRP). The Flash Read Protect Option bit prevents the code in memory from being read out of Z8 Encore! XP[®] F0823 Series products. When this option is enabled, several of the OCD commands are disabled. Table 99 on page 162 is a summary of the OCD commands. Each OCD command is described in further detail in the bulleted list following this table. Table 99 on page 162 also indicates those commands that operate when the device is not in DEBUG mode (normal operation) and those commands that are disabled by programming the Flash Read Protect Option bit.

Debug Command	Command Byte	Enabled when NOT in DEBUG mode?	Disabled by Flash Read Protect Option Bit
Read OCD Revision	00H	Yes	-
Reserved	01H	_	_
Read OCD Status Register	02H	Yes	-
Read Runtime Counter	03H	_	-
Write OCD Control Register	04H	Yes	Cannot clear DBGMODE bit.
Read OCD Control Register	05H	Yes	-
Write Program Counter	06H	_	Disabled.
Read Program Counter	07H	_	Disabled.
Write Register	08H	_	Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control register.
Read Register	09H	-	Disabled.
Write Program Memory	0AH	_	Disabled.
Read Program Memory	0BH	_	Disabled.
Write Data Memory	0CH	_	Yes.
Read Data Memory	0DH	-	_
Read Program Memory CRC	OEH	_	-
Reserved	0FH	_	-
Step Instruction	10H	_	Disabled.

Z8 Encore! XP[®] F0823 Series Product Specification

Oscillator Control

Z8 Encore! XP[®] F0823 Series devices uses three possible clocking schemes, each user-selectable:

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

In addition, Z8 Encore! XP F0823 Series devices contain clock failure detection and recovery circuitry, allowing continued operation despite a failure of the primary oscillator.

Operation

This chapter discusses the logic used to select the system clock and handle primary oscillator failures. A description of the specific operation of each oscillator is outlined elsewhere in this document.

System Clock Selection

The oscillator control block selects from the available clocks. Table 101 details each clock source and its usage.

Clock Source	Characteristics	Required Setup
Internal Precision RC Oscillator	 32.8 kHz or 5.53 MHz ± 4% accuracy when trimmed No external components required 	 Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator at either 5.53 MHz or 32.8 kHz
External Clock Drive	 0 to 20 MHz Accuracy dependent on external clock source 	 Write GPIO registers to configure PB3 pin for external clock function Unlock and write OSCCTL to select external system clock Apply external clock signal to GPIO
Internal Watchdog Timer Oscillator	 10 kHz nominal ± 40% accuracy; no external components required Very Low power consumption 	 Enable WDT if not enabled and wait until WDT Oscillator is operating. Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator

Table 101. Oscillator Configuration and Selection

Table 105. Notational Shorthand

Notation	Description	Operand	Range	
b	Bit	b	b represents a value from 0 to 7 (000B to 111B).	
СС	Condition Code	—	See Condition Codes overview in the eZ8 CPU User Manual.	
DA	Direct Address	Addrs	Addrs represents a number in the range of 0000H to FFFFH.	
ER	Extended Addressing Register	Reg	Reg represents a number in the range of 000H to FFFH.	
IM	Immediate Data	#Data	Data is a number between 00H to FFH.	
lr	Indirect Working Register	@Rn	n = 0–15.	
IR	Indirect Register	@Reg	Reg. represents a number in the range of 00H to FFH.	
Irr	Indirect Working Register Pair	@RRp	p = 0, 2, 4, 6, 8, 10, 12, or 14.	
IRR	Indirect Register Pair	@Reg	Reg represents an even number in the range 00H to FEH	
р	Polarity	р	Polarity is a single bit binary value of either 0B or 1B.	
r	Working Register	Rn	n = 0–15.	
R	Register	Reg	Reg. represents a number in the range of 00H to FFH.	
RA	Relative Address	Х	X represents an index in the range of +127 to – 128 which is an offset relative to the address of the next instruction	
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12, or 14.	
RR	Register Pair	Reg	Reg. represents an even number in the range of 00H to FEH.	
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH.	
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to -128 range.	

Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 26. Figure 27 and Figure 28 provide information about each of the eZ8 CPU instructions. Table 116 lists Opcode Map abbreviations.



Figure 26. Opcode Map Cell Description

205

On-Chip Debugger Timing

Figure 31 and Table 129 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4 ns maximum rise and fall time.



		Delay (ns)		
Parameter	Abbreviation	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	-	

Table 129. On-Chip Debugger Timing

Figure 42 displays the 28-pin Small Shrink Outline Package (SSOP) available for Z8 Encore! XP F0823 Series devices.



SYMBOL	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
А	1.73	1.86	1.99	0.068	0.073	0.078
A1	0.05	0.13	0.21	0.002	0.005	0.008
A2	1.68	1.73	1.78	0.066	0.068	0.070
В	0.25		0.38	0.010		0.015
С	0.09	-	0.20	0.004	0.006	0.008
D	10.07	10.20	10.33	0.397	0.402	0.407
E	5.20	5.30	5.38	0.205	0.209	0.212
е	0.65 TYP			0.0256 TYP		
Н	7.65	7.80	7.90	0.301	0.307	0.311
L	0.63	0.75	0.95	0.025	0.030	0.037

CONTROLLING DIMENSIONS: MM LEADS ARE COPLANAR WITHIN .004 INCHES.

Figure 42. 28-Pin Small Shrink Outline Package (SSOP)