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, WDT
Number of I/O	25
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f021asj020sc00tr

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

vector address. Following Stop Mode Recovery, the STOP bit in the Reset Status (RSTSTAT) Register is set to 1. Table 10 lists the Stop Mode Recovery sources and resulting actions. The text following provides more detailed information about each of the Stop Mode Recovery sources.

Table 10. 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 Reset Status (RSTSTAT) register, the WDT and STOP bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and the Z8 Encore! XP F082A 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 may 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 (see Table 131 on page 229). 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*

Low-Power Modes

The Z8 Encore! XP F082A Series products contain power-saving features. The highest level of power reduction is provided by the STOP mode, in which nearly all device functions are powered down. The next lower level of power reduction is provided by the HALT mode, in which the CPU is powered down.

Further power savings can be implemented by disabling individual peripheral blocks while in Active mode (defined as being in neither STOP nor HALT mode).

STOP Mode

Executing the eZ8 CPU's STOP instruction places the device into STOP mode, powering down all peripherals except the Voltage Brownout detector, the Low-power Operational Amplifier and the Watchdog Timer. These three blocks may also be disabled for additional power savings. Specifically, the operating characteristics are:

- Primary crystal oscillator and internal precision oscillator are stopped; XIN and XOUT (if previously enabled) are disabled, and PA0/PA1 revert to the states programmed by the GPIO registers.

- System clock is stopped.

- eZ8 CPU is stopped.

- Program counter (PC) stops incrementing.

- Watchdog Timer's internal RC oscillator continues to operate if enabled by the Oscillator Control register.

- If enabled, the Watchdog Timer logic continues to operate.

- If enabled for operation in STOP mode by the associated Flash Option Bit, the Voltage Brownout protection circuit continues to operate.

- Low-power operational amplifier continues to operate if enabled by the Power Control register to do so.

- All other on-chip peripherals are idle.

To minimize current in STOP mode, all GPIO pins that are configured as digital inputs must be driven to one of the supply rails (V_{CC} or GND). Additionally, any GPIOs configured as outputs must also be driven to one of the supply rails. The device can be brought out of STOP mode using Stop Mode Recovery. For more information on Stop Mode Recovery, see Reset, Stop Mode Recovery, and Low Voltage Detection on page 23.

Table 14. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port B	PB0	Reserved		AFS1[0]: 0
		ANA0/AMPOUT	ADC Analog Input/LPO Output	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1/AMPINN	ADC Analog Input/LPO Input (N)	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2/AMPINP	ADC Analog Input/LPO Input (P)	AFS1[2]: 1
	PB3	CLKIN	External Clock Input	AFS1[3]: 0
		ANA3	ADC Analog Input	AFS1[3]: 1
	PB4	Reserved		AFS1[4]: 0
		ANA7	ADC Analog Input	AFS1[4]: 1
	PB5	Reserved		AFS1[5]: 0
		VREF*	ADC Voltage Reference	AFS1[5]: 1
	PB6	Reserved		AFS1[6]: 0
		Reserved		AFS1[6]: 1
	PB7	Reserved		AFS1[7]: 0
		Reserved		AFS1[7]: 1

Note: Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set register AFS2 is not used to select the function. Also, alternate function selection as described in Port A–D Alternate Function Sub-Registers on page 47 must also be enabled.

* VREF is available on PB5 in 28-pin products only.

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

Table 20. Port A–D Alternate Function Sub-Registers (PxAF)

BITS	7	6	5	4	3	2	1	0
FIELD	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
RESET	00H (Ports A–C); 01H (Port D); 04H (Port A of 8-pin device)							
R/W	R/W							
ADDR	If 02H in Port A–D Address Register, accessible through the Port A–D Control Register							

AF[7:0]—Port Alternate Function enabled
0 = The port pin is in normal mode and the DDx bit in the Port A–D Data Direction sub-register determines the direction of the pin.
1 = The alternate function selected through Alternate Function Set sub-registers is enabled. Port pin operation is controlled by the alternate function.

Port A–D Output Control Sub-Registers

The Port A–D Output Control sub-register (Table 21) is accessed through the Port A–D Control register by writing 03H to the Port A–D Address register. Setting the bits in the Port A–D Output Control sub-registers to 1 configures the specified port pins for open-drain operation. These sub-registers affect the pins directly and, as a result, alternate functions are also affected.

Table 21. Port A–D Output Control Sub-Registers (PxOC)

BITS	7	6	5	4	3	2	1	0
FIELD	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
RESET	00H (Ports A-C); 01H (Port D)							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	If 03H in Port A–D Address Register, accessible through the Port A–D Control Register							

POC[7:0]—Port Output Control
These bits function independently of the alternate function bit and always disable the drains if set to 1.
0 = The source current is enabled for any output mode (unless overridden by the alternate

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) register (Table 35) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 2 register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 2 Register (IRQ2)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved				PC3I	PC2I	PC1I	PC0I
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	FC6H							

Reserved—Must be 0.

PC*x*I—Port C Pin *x* Interrupt Request
0 = No interrupt request is pending for GPIO Port C pin *x*.
1 = An interrupt request from GPIO Port C pin *x* is awaiting service.

where *x* indicates the specific GPIO Port C pin number (0–3).

IRQ0 Enable High and Low Bit Registers

Table 36 describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers (Table 37 and Table 38) form a priority encoded enabling for interrupts in the Interrupt Request 0 register.

Table 36. IRQ0 Enable and Priority Encoding

IRQ0ENH[*x*]	IRQ0ENL[*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.

Reserved—Must be 0.

C3ENL—Port C3 Interrupt Request Enable Low Bit
C2ENL—Port C2 Interrupt Request Enable Low Bit
C1ENL—Port C1 Interrupt Request Enable Low Bit
C0ENL—Port C0 Interrupt Request Enable Low Bit

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register (Table 45) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A input pin.

Table 45. Interrupt Edge Select Register (IRQES)

BITS	7	6	5	4	3	2	1	0
FIELD	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
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	FCDH							

IESx—Interrupt Edge Select x
0 = An interrupt request is generated on the falling edge of the PAx input.
1 = An interrupt request is generated on the rising edge of the PAx input.
where x indicates the specific GPIO Port pin number (0 through 7).

Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) register (Table 46) determines the source of the PADxS interrupts. The Shared Interrupt Select register selects between Port A and alternate sources for the individual interrupts.

Because these shared interrupts are edge-triggered, it is possible to generate an interrupt just by switching from one shared source to another. For this reason, an interrupt must be disabled before switching between sources.
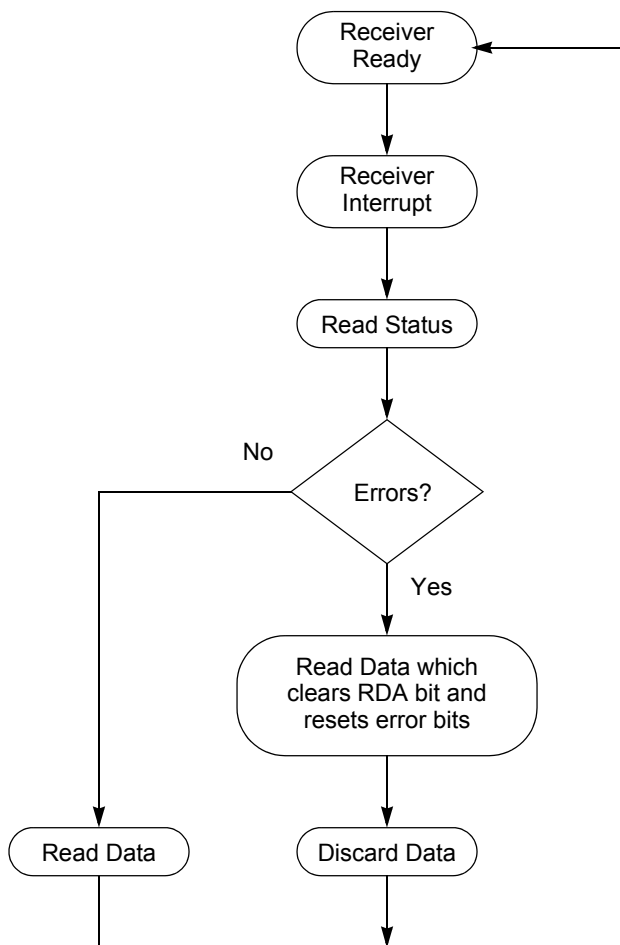
Figure 15. UART Receiver Interrupt Service Routine Flow

Baud Rate Generator Interrupts

If the baud rate generator (BRG) interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This condition allows the Baud Rate Generator to function as an additional counter if the UART functionality is not employed.

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UART Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value

baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the Endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the Endec clock counter is reset, resynchronizing the Endec to the incoming signal, allowing the Endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the Endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All Infrared Endec configuration and status information is set by the UART control registers as defined in Universal Asynchronous Receiver/Transmitter on page 97.

⚠ **Caution:** *To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.*

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.*

read operations to illegal addresses. Also, the user code must pop the address byte off the stack.

The read routine uses 9 bytes of stack space in addition to the one byte of address pushed by the user. Sufficient memory must be available for this stack usage.

Because of the Flash memory architecture, NVDS reads exhibit a non-uniform execution time. A read operation takes between 44 µs and 489 µs (assuming a 20 MHz system clock). Slower system clock speeds result in proportionally higher execution times.

NVDS byte reads from invalid addresses (those exceeding the NVDS array size) return 0xff. Illegal read operations have a 2 µs execution time.

The status byte returned by the NVDS read routine is zero for successful read, as determined by a CRC check. If the status byte is non-zero, there was a corrupted value in the NVDS array at the location being read. In this case, the value returned in R0 is the byte most recently written to the array that does not have a CRC error.

Power Failure Protection

The NVDS routines employ error checking mechanisms to ensure a power failure endangers only the most recently written byte. Bytes previously written to the array are not perturbed.

A system reset (such as a pin reset or Watchdog Timer reset) that occurs during a write operation also perturbs the byte currently being written. All other bytes in the array are unperturbed.

Optimizing NVDS Memory Usage for Execution Speed

The NVDS read time varies drastically, this discrepancy being a trade-off for minimizing the frequency of writes that require post-write page erases (see Table 104). The NVDS read time of address N is a function of the number of writes to addresses other than N since the most recent write to address N, as well as the number of writes since the most recent page erase. Neglecting effects caused by page erases and results caused by the initial condition in which the NVDS is blank, a rule of thumb is that every write since the most recent page erase causes read times of unwritten addresses to increase by 1 µs, up to a maximum of (511-NVDS_SIZE) µs.

Table 104. NVDS Read Time

Operation	Minimum Latency	Maximum Latency
Read (16 byte array)	875	9961
Read (64 byte array)	876	8952

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 116 through Table 123 lists the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instruction can be considered as a subset of more than one category. Within these tables, the source operand is identified as 'src', the destination operand is 'dst' and a condition code is 'cc'.

Table 116. Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst, src	Add with Carry
ADCX	dst, src	Add with Carry using Extended Addressing
ADD	dst, src	Add
ADDX	dst, src	Add using Extended Addressing
CP	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using Extended Addressing
CPX	dst, src	Compare using Extended Addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word
MULT	dst	Multiply
SBC	dst, src	Subtract with Carry
SBCX	dst, src	Subtract with Carry using Extended Addressing
SUB	dst, src	Subtract
SUBX	dst, src	Subtract using Extended Addressing

Table 123. Rotate and Shift Instructions (Continued)

Mnemonic	Operands	Instruction
SRA	dst	Shift Right Arithmetic
SRL	dst	Shift Right Logical
SWAP	dst	Swap Nibbles

eZ8 CPU Instruction Summary

Table 124 summarizes the eZ8 CPU instructions. The table identifies the addressing modes employed by the instruction, the effect upon the Flags register, the number of CPU clock cycles required for the instruction fetch, and the number of CPU clock cycles required for the instruction execution.

Table 124. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	2	3
		r	Ir	13							2	4
		R	R	14							3	3
		R	IR	15							3	4
		R	IM	16							3	3
		IR	IM	17							3	4
ADCX dst, src	dst ← dst + src + C	ER	ER	18	*	*	*	*	0	*	4	3
		ER	IM	19							4	3
ADD dst, src	dst ← dst + src	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	03							2	4
		R	R	04							3	3
		R	IR	05							3	4
		R	IM	06							3	3
		IR	IM	07							3	4
ADDX dst, src	dst ← dst + src	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							4	3

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

Electrical Characteristics

The data in this chapter is pre-qualification and pre-characterization and is subject to change. Additional electrical characteristics may be found in the individual chapters.

Absolute Maximum Ratings

Stresses greater than those listed in Table 126 may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, tie unused inputs to one of the supply voltages (V_{DD} or V_{SS}).

Table 126. Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units	Notes
Ambient temperature under bias	-40	+105	°C	
Storage temperature	-65	+150	°C	
Voltage on any pin with respect to V_{SS}	-0.3	+5.5	V	1
	-0.3	+3.9	V	2
Voltage on V_{DD} pin with respect to V_{SS}	-0.3	+3.6	V	
Maximum current on input and/or inactive output pin	-5	+5	µA	
Maximum output current from active output pin	-25	+25	mA	
8-pin Packages Maximum Ratings at 0 °C to 70 °C				
Total power dissipation		220	mW	
Maximum current into V_{DD} or out of V_{SS}		60	mA	
20-pin Packages Maximum Ratings at 0 °C to 70 °C				
Total power dissipation		430	mW	
Maximum current into V_{DD} or out of V_{SS}		120	mA	
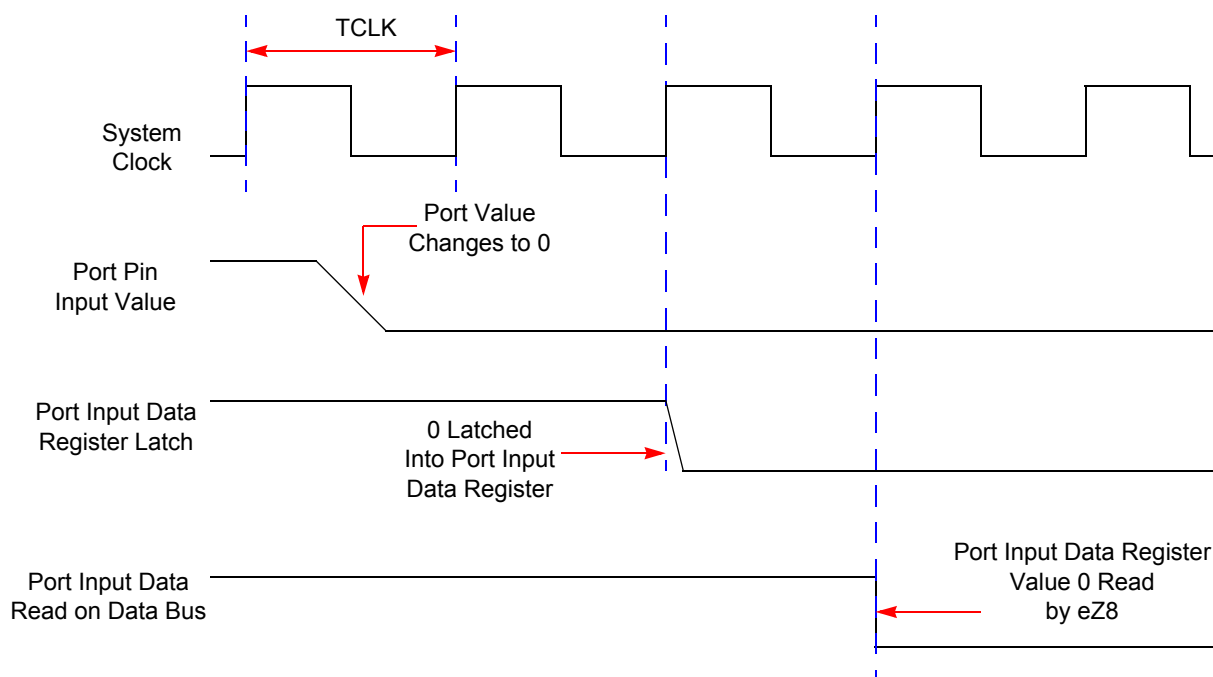
Figure 34. Port Input Sample Timing

Table 139. GPIO Port Input Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
T_{S_PORT}	Port Input Transition to XIN Rise Setup Time (Not pictured)	5	–
T_{H_PORT}	XIN Rise to Port Input Transition Hold Time (Not pictured)	0	–
T_{SMR}	GPIO Port Pin Pulse Width to ensure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources)	1 µs	

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.
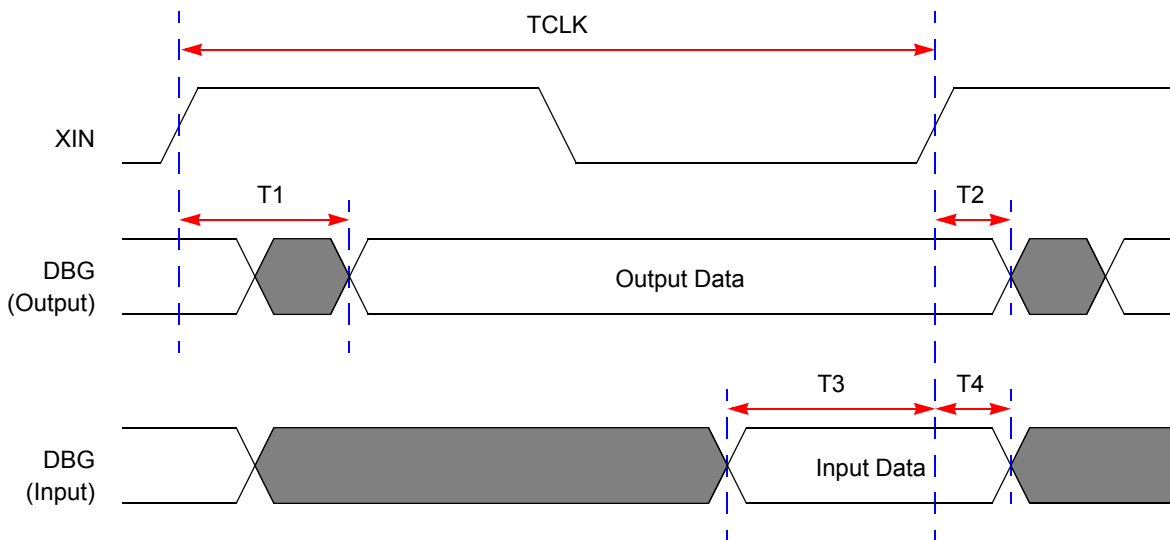
Figure 36. On-Chip Debugger Timing

Table 141. On-Chip Debugger Timing

		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	–

Figure 46 displays the 28-pin Small Outline Integrated Circuit package (SOIC) available in the Z8 Encore! XP F082A Series devices.

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	2.40	2.64	.094	.104
A1	0.10	0.30	.004	.012
A2	2.24	2.44	.088	.096
B	0.36	0.46	.014	.018
C	0.23	0.30	.009	.012
D	17.78	18.00	.700	.710
E	7.40	7.60	.291	.299
e	1.27 BSC		.050 BSC	
H	10.00	10.65	.394	.419
h	0.30	0.71	.012	.028
L	0.61	1.00	.024	.039
Q1	0.97	1.09	.038	.043

CONTROLLING DIMENSIONS : MM
LEADS ARE COPLANAR WITHIN .004 INCH.
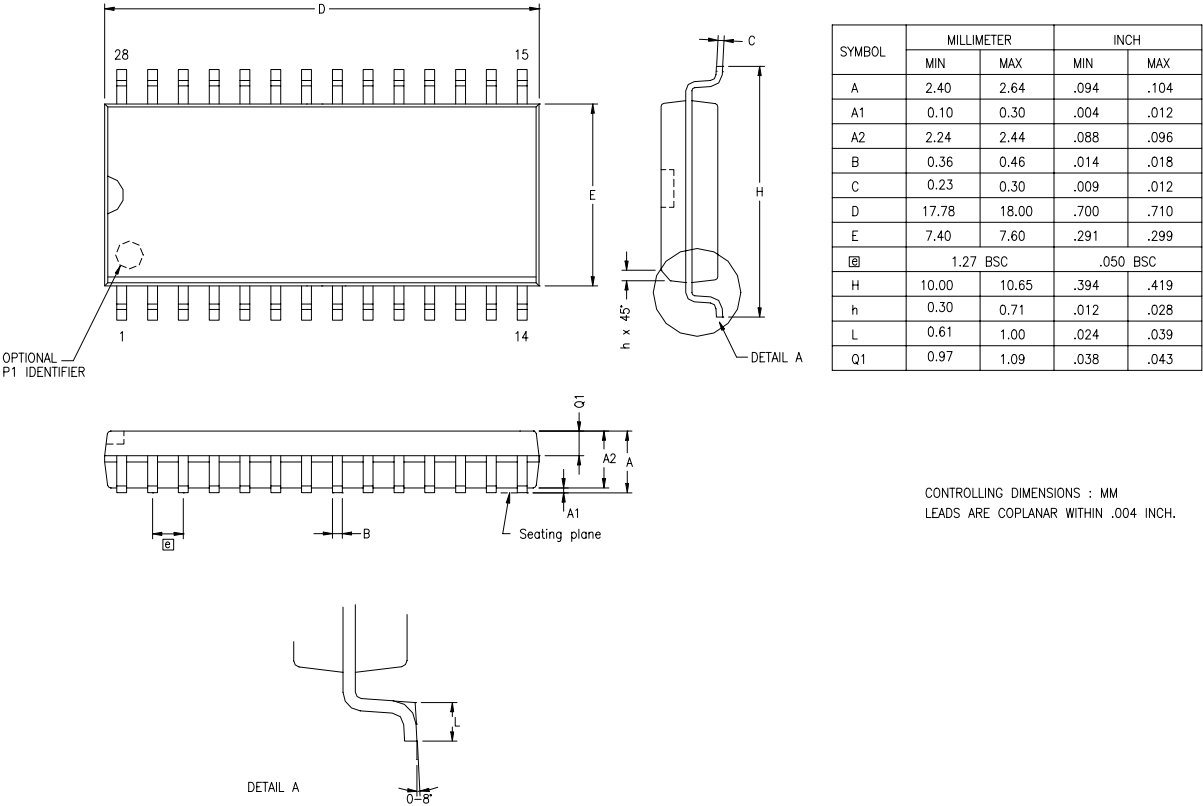
Figure 46. 28-Pin Small Outline Integrated Circuit Package (SOIC)