

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	5MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	24
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0113pj005sc

Operation	152
OCD Interface	152
DEBUG Mode	153
OCD Data Format	154
OCD Auto-Baud Detector/Generator	154
OCD Serial Errors	155
OCD Unlock Sequence (8-Pin Devices Only)	156
Breakpoints	156
Runtime Counter	156
On-Chip Debugger Commands	157
On-Chip Debugger Control Register Definitions	161
OCD Control Register	161
OCD Status Register	163
Oscillator Control	165
Operation	165
System Clock Selection	165
Clock Failure Detection and Recovery	166
Oscillator Control Register Definitions	167
Internal Precision Oscillator	169
Operation	169
eZ8 CPU Instruction Set	171
Assembly Language Programming Introduction	171
Assembly Language Syntax	172
eZ8 CPU Instruction Notation	172
eZ8 CPU Instruction Classes	174
eZ8 CPU Instruction Summary	179
Opcode Maps	188
Electrical Characteristics	193
Absolute Maximum Ratings	193
DC Characteristics	194
AC Characteristics	197
On-Chip Peripheral AC and DC Electrical Characteristics	199
General Purpose I/O Port Input Data Sample Timing	202
General Purpose I/O Port Output Timing	204
On-Chip Debugger Timing	205
UART Timing	206
Packaging	209

Table 9. Reset and Stop Mode Recovery Characteristics and Latency

Reset Type	Reset Characteristics and Latency		
	Control Registers	eZ8 CPU	Reset Latency (Delay)
System Reset	Reset (as applicable)	Reset	66 Internal Precision Oscillator Cycles
Stop Mode Recovery	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	66 Internal Precision Oscillator Cycles + IPO startup time

During a System Reset or Stop Mode Recovery, the IPO requires 4 μ s to start up. Then the Z8 Encore! XP F0823 Series device is held in Reset for 66 cycles of the Internal Precision Oscillator. If the crystal oscillator is enabled in the Flash option bits, this reset period is increased to 5000 IPO cycles. When a reset occurs because of a low voltage condition or Power-On Reset, this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the reset period, the device remains in reset until the pin is deasserted.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address.

When the control registers are re-initialized by a system reset, the system clock after reset is always the IPO. The software must reconfigure the oscillator control block, such that the correct system clock source is enabled and selected.

Reset Sources

Table 10 lists the possible sources of a System Reset.

Table 16. Port Alternate Function Mapping (8-Pin Parts)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Select Register AFS1	Alternate Function Select Register AFS2
Port A	PA0	T0IN	Timer 0 Input	AFS1[0]: 0	AFS2[0]: 0
		Reserved		AFS1[0]: 0	AFS2[0]: 1
		Reserved		AFS1[0]: 1	AFS2[0]: 0
		$\overline{T0OUT}$	Timer 0 Output Complement	AFS1[0]: 1	AFS2[0]: 1
	PA1	T0OUT	Timer 0 Output	AFS1[1]: 0	AFS2[1]: 0
		Reserved		AFS1[1]: 0	AFS2[1]: 1
		CLKIN	External Clock Input	AFS1[1]: 1	AFS2[1]: 0
		Analog Functions*	ADC Analog Input/VREF	AFS1[1]: 1	AFS2[1]: 1
	PA2	DE0	UART 0 Driver Enable	AFS1[2]: 0	AFS2[2]: 0
		\overline{RESET}	External Reset	AFS1[2]: 0	AFS2[2]: 1
		T1OUT	Timer 1 Output	AFS1[2]: 1	AFS2[2]: 0
		Reserved		AFS1[2]: 1	AFS2[2]: 1
	PA3	$\overline{CTS0}$	UART 0 Clear to Send	AFS1[3]: 0	AFS2[3]: 0
		COUT	Comparator Output	AFS1[3]: 0	AFS2[3]: 1
		T1IN	Timer 1 Input	AFS1[3]: 1	AFS2[3]: 0
		Analog Functions*	ADC Analog Input	AFS1[3]: 1	AFS2[3]: 1
	PA4	RXD0	UART 0 Receive Data	AFS1[4]: 0	AFS2[4]: 0
		Reserved		AFS1[4]: 0	AFS2[4]: 1
		Reserved		AFS1[4]: 1	AFS2[4]: 0
		Analog Functions*	ADC/Comparator Input (N)	AFS1[4]: 1	AFS2[4]: 1
	PA5	TXD0	UART 0 Transmit Data	AFS1[5]: 0	AFS2[5]: 0
		$\overline{T1OUT}$	Timer 1 Output Complement	AFS1[5]: 0	AFS2[5]: 1
		Reserved		AFS1[5]: 1	AFS2[5]: 0
		Analog Functions*	ADC/Comparator Input (P)	AFS1[5]: 1	AFS2[5]: 1

Note: * Analog Functions include ADC inputs, ADC reference and comparator inputs. Also, alternate function selection as described in Port A–C Alternate Function Sub-Registers must be enabled.

Table 19. Port A–C 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							

PCTL[7:0]—Port Control

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

Port A–C Data Direction Sub-Registers

The Port A–C Data Direction sub-register is accessed through the Port A–C Control register by writing 01H to the Port A–C Address register (Table 20).

Table 20. Port A–C 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–C Address Register, accessible through the Port A–C 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–C Output Data register is driven onto the port pin.

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

Port A–C Alternate Function Sub-Registers

The Port A–C Alternate Function sub-register (Table 21) is accessed through the Port A–C Control register by writing 02H to the Port A–C Address register. The Port A–C 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–C Alternate Function Set 1 Sub-Registers on page 48 and Port A–C Alternate Function Set 2 Sub-Registers on

page 49. See GPIO Alternate Functions on page 36 to determine the alternate function associated with each port pin.

! 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 21. Port A–C 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); 04H (Port A of 8-pin device)							
R/W	R/W							
ADDR	If 02H in Port A–C Address Register, accessible through the Port A–C 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–C 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–C Output Control Sub-Registers

The Port A–C Output Control sub-register (Table 22) is accessed through the Port A–C Control register by writing 03H to the Port A–C Address register. Setting the bits in the Port A–C Output Control 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 22. Port A–C Output Control Sub-Registers (PxOC)

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

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	If 06H in Port A–C Address Register, accessible through the Port A–C Control 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.*

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

BITS	7	6	5	4	3	2	1	0
FIELD	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
ADDR	FD2H, FD6H, FDAH							

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.

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.

! **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

Table 41. 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] 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

For selection of Port A as the interrupt source, see Shared Interrupt Select Register on page 64.

Table 42. 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							

PA7VENH—Port A Bit[7] Interrupt Request Enable Low Bit

PA6CENH—Port A Bit[6] or Comparator Interrupt Request Enable Low Bit

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

IRQ2 Enable High and Low Bit Registers

Table 43 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers (Table 44 and Table 45) form a priority encoded enabling for interrupts in the Interrupt Request 2 register. Priority is generated by setting bits in each register.

Table 43. IRQ2 Enable and Priority Encoding

IRQ2ENH[x]	IRQ2ENL[x]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal

Table 43. IRQ2 Enable and Priority Encoding (Continued)

IRQ2ENH[x]	IRQ2ENL[x]	Priority	Description
1	1	Level 3	High

where x indicates the register bits from 0–7.

Table 44. IRQ2 Enable High Bit Register (IRQ2ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved				C3ENH	C2ENH	C1ENH	C0ENH
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	FC7H							

Reserved—Must be 0

C3ENH—Port C3 Interrupt Request Enable High Bit

C2ENH—Port C2 Interrupt Request Enable High Bit

C1ENH—Port C1 Interrupt Request Enable High Bit

C0ENH—Port C0 Interrupt Request Enable High Bit

Table 45. IRQ2 Enable Low Bit Register (IRQ2ENL)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved				C3ENL	C2ENL	C1ENL	C0ENL
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	FC8H							

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 46) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A or Port D input pin.

CAPTURE/COMPARE Mode

0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal.

1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.

PWM DUAL OUTPUT Mode

0 = Timer Output is forced Low (0) and Timer Output Complement is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon Reload. When enabled, the Timer Output Complement is forced Low (0) upon PWM count match and forced High (1) upon Reload. The PWMD field in TxCTL0 register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to High (1).

1 = Timer Output is forced High (1) and Timer Output Complement is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon Reload. When enabled, the Timer Output Complement is forced High (1) upon PWM count match and forced Low (0) upon Reload. The PWMD field in TxCTL0 register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to Low (0).

CAPTURE RESTART Mode

0 = Count is captured on the rising edge of the Timer Input signal

1 = Count is captured on the falling edge of the Timer Input signal

COMPARATOR COUNTER 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.

! Caution: *When the Timer Output alternate function TxOUT on a GPIO port pin is enabled, TxOUT changes to whatever state the TPOL bit is in. The timer does not need to be enabled for that to happen. Also, the port data direction sub register is not needed to be set to output on TxOUT. Changing the TPOL bit with the timer enabled and running does not immediately change the TxOUT.*

PRES—Prescale value.

The timer input clock is divided by 2^{PRES} , where PRES can be set from 0 to 7. The prescaler is reset each time the Timer is disabled. This reset ensures proper clock division each time the Timer is restarted.

000 = Divide by 1

001 = Divide by 2

REFSELL—Voltage Reference Level Select Low Bit; in conjunction with the High bit (REFSELH) in ADC Control/Status Register 1, this determines the level of the internal voltage reference; the following details the effects of {REFSELH, REFSELL};

► **Note:** *This reference is independent of the Comparator reference.*

00= Internal Reference Disabled, reference comes from external pin.

01= Internal Reference set to 1.0 V

10= Internal Reference set to 2.0 V (default)

REFEXT—External Reference Select

0 = External reference buffer is disabled; V_{ref} pin is available for GPIO functions

1 = The internal ADC reference is buffered and connected to the V_{ref} pin

CONT

0 = Single-shot conversion. ADC data is output once at completion of the 5129 system clock cycles.

1 = Continuous conversion. ADC data updated every 256 system clock cycles.

ANAIN[3:0]—Analog Input Select

These bits select the analog input for conversion. Not all port pins in this list are available in all packages for Z8 Encore! XP[®] F0823 Series. For information on the port pins available with each package style, see Pin Description on page 7. Do not enable unavailable analog inputs. Usage of these bits changes depending on the buffer mode selected in ADC Control/Status Register 1.

For the reserved values, all input switches are disabled to avoid leakage or other undesirable operation. ADC samples taken with reserved bit settings are undefined.

Single-Ended:

0000 = ANA0

0001 = ANA1

0010 = ANA2

0011 = ANA3

0100 = ANA4

0101 = ANA5

0110 = ANA6

0111 = ANA7

1000 = Reserved

1001 = Reserved

1010 = Reserved

1011 = Reserved

1100 = Reserved

1101 = Reserved

1110 = Reserved

1111 = Reserved

ADC Data Low Bits Register

The ADC Data Low Byte register contains the lower bits of the ADC output as well as an overflow status bit. The output is a 11-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 75. ADC Data Low Bits 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							

ADCDL—ADC Data Low Bits

These bits are the least significant three bits of the 11-bits of the ADC output. These bits are undefined after a Reset.

Reserved—Undefined when read

OVF—Overflow Status

0= An overflow did not occur in the digital filter for the current sample

1= An overflow did occur in the digital filter for the current sample

Table 97. Randomized Lot ID Locations (Continued)

Info Page Address	Memory Address	Usage
5C	FE5C	Randomized Lot ID Byte 23
5D	FE5D	Randomized Lot ID Byte 22
5E	FE5E	Randomized Lot ID Byte 21
5F	FE5F	Randomized Lot ID Byte 20
61	FE61	Randomized Lot ID Byte 19
62	FE62	Randomized Lot ID Byte 18
64	FE64	Randomized Lot ID Byte 17
65	FE65	Randomized Lot ID Byte 16
67	FE67	Randomized Lot ID Byte 15
68	FE68	Randomized Lot ID Byte 14
6A	FE6A	Randomized Lot ID Byte 13
6B	FE6B	Randomized Lot ID Byte 12
6D	FE6D	Randomized Lot ID Byte 11
6E	FE6E	Randomized Lot ID Byte 10
70	FE70	Randomized Lot ID Byte 9
71	FE71	Randomized Lot ID Byte 8
73	FE73	Randomized Lot ID Byte 7
74	FE74	Randomized Lot ID Byte 6
76	FE76	Randomized Lot ID Byte 5
77	FE77	Randomized Lot ID Byte 4
79	FE79	Randomized Lot ID Byte 3
7A	FE7A	Randomized Lot ID Byte 2
7C	FE7C	Randomized Lot ID Byte 1
7D	FE7D	Randomized Lot ID Byte 0 (least significant)

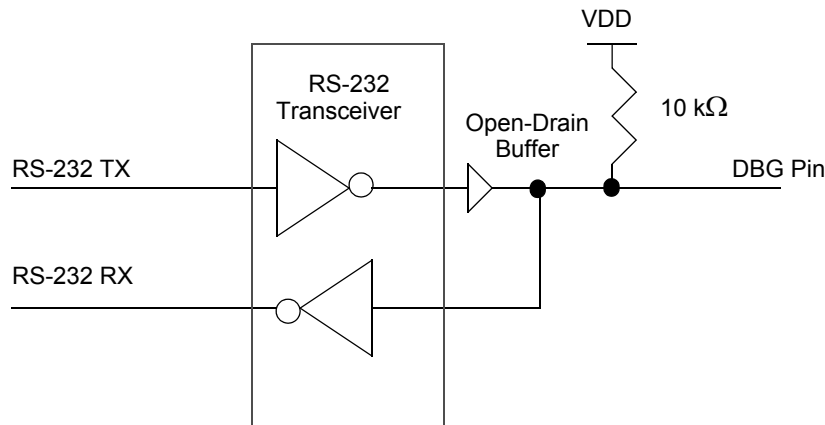


Figure 24. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (2)

DEBUG Mode

The operating characteristics of the devices in DEBUG mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions
- The system clock operates unless in STOP mode
- All enabled on-chip peripherals operate unless in STOP mode
- Automatically exits HALT mode
- Constantly refreshes the Watchdog Timer, if enabled.

Entering DEBUG Mode


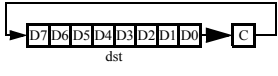
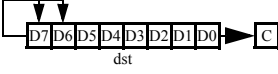
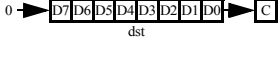
The device enters DEBUG mode following the operations below:

- The device enters DEBUG mode after the eZ8 CPU executes a BRK (breakpoint) instruction
- If the DBG pin is held Low during the most recent clock cycle of System Reset, the part enters DEBUG mode upon exiting System Reset

► **Note:** Holding the DBG pin Low for an additional 5000 (minimum) clock cycles after reset (making sure to account for any specified frequency error if using an internal oscillator) prevents a false interpretation of an Autobaud sequence (see OCD Auto-Baud Detector/Generator on page 154).

- If the PA2/RESET pin is held Low while a 32-bit key sequence is issued to the PA0/DBG pin, the DBG feature is unlocked. After releasing PA2/RESET, it is pulled high. At this

Table 115. 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		
RR dst		R		E0	*	*	*	*	-	-	2	2
		IR		E1							2	3
RRC dst		R		C0	*	*	*	*	-	-	2	2
		IR		C1							2	3
SBC dst, src	$dst \leftarrow dst - src - C$	r	r	32	*	*	*	*	1	*	2	3
		r	lr	33							2	4
		R	R	34							3	3
		R	IR	35							3	4
		R	IM	36							3	3
		IR	IM	37							3	4
SBCX dst, src	$dst \leftarrow dst - src - C$	ER	ER	38	*	*	*	*	1	*	4	3
		ER	IM	39							4	3
SCF	$C \leftarrow 1$			DF	1	-	-	-	-	-	1	2
SRA dst		R		D0	*	*	*	0	-	-	2	2
		IR		D1							2	3
SRL dst		R		1F C0	*	*	0	*	-	-	3	2
		IR		1F C1							3	3
SRP src	$RP \leftarrow src$		IM	01	-	-	-	-	-	-	2	2
STOP	STOP Mode			6F	-	-	-	-	-	-	1	2
SUB dst, src	$dst \leftarrow dst - src$	r	r	22	*	*	*	*	1	*	2	3
		r	lr	23							2	4
		R	R	24							3	3
		R	IR	25							3	4
		R	IM	26							3	3
		IR	IM	27							3	4

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

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.

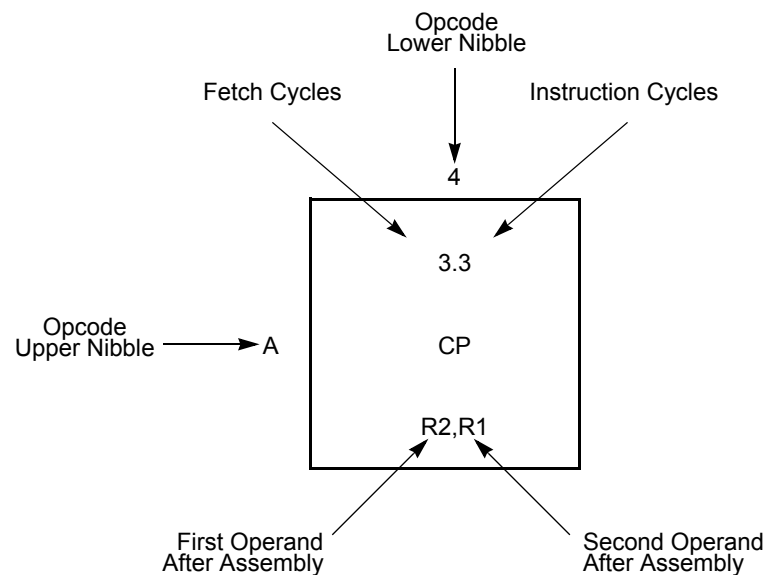


Figure 26. Opcode Map Cell Description

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

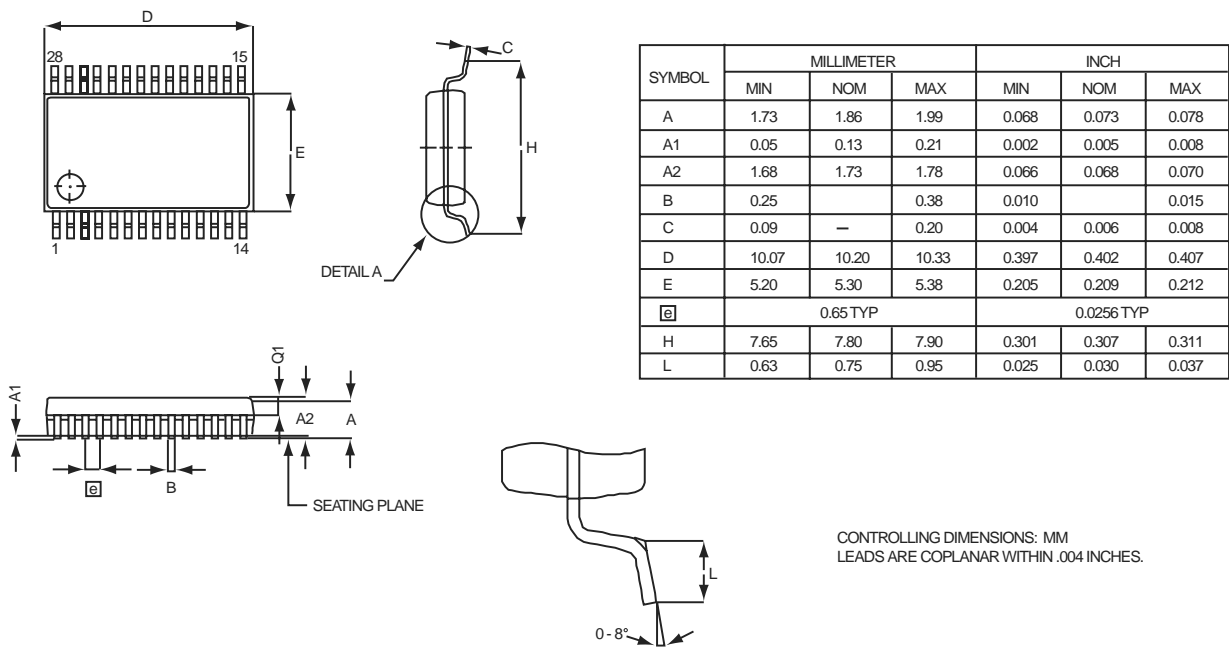


Figure 42. 28-Pin Small Shrink Outline Package (SSOP)

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Description
Z8 Encore! XP with 2 KB Flash, 10-Bit Analog-to-Digital Converter								
Standard Temperature: 0 °C to 70 °C								
Z8F0223PB005SC	2 KB	512 B	6	12	2	4	1	PDIP 8-pin package
Z8F0223QB005SC	2 KB	512 B	6	12	2	4	1	QFN 8-pin package
Z8F0223SB005SC	2 KB	512 B	6	12	2	4	1	SOIC 8-pin package
Z8F0223SH005SC	2 KB	512 B	16	18	2	7	1	SOIC 20-pin package
Z8F0223HH005SC	2 KB	512 B	16	18	2	7	1	SSOP 20-pin package
Z8F0223PH005SC	2 KB	512 B	16	18	2	7	1	PDIP 20-pin package
Z8F0223SJ005SC	2 KB	512 B	22	18	2	8	1	SOIC 28-pin package
Z8F0223HJ005SC	2 KB	512 B	22	18	2	8	1	SSOP 28-pin package
Z8F0223PJ005SC	2 KB	512 B	22	18	2	8	1	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C								
Z8F0223PB005EC	2 KB	512 B	6	12	2	4	1	PDIP 8-pin package
Z8F0223QB005EC	2 KB	512 B	6	12	2	4	1	QFN 8-pin package
Z8F0223SB005EC	2 KB	512 B	6	12	2	4	1	SOIC 8-pin package
Z8F0223SH005EC	2 KB	512 B	16	18	2	7	1	SOIC 20-pin package
Z8F0223HH005EC	2 KB	512 B	16	18	2	7	1	SSOP 20-pin package
Z8F0223PH005EC	2 KB	512 B	16	18	2	7	1	PDIP 20-pin package
Z8F0223SJ005EC	2 KB	512 B	22	18	2	8	1	SOIC 28-pin package
Z8F0223HJ005EC	2 KB	512 B	22	18	2	8	1	SSOP 28-pin package
Z8F0223PJ005EC	2 KB	512 B	22	18	2	8	1	PDIP 28-pin package
Replace C with G for Lead-Free Packaging								

- ADC data high byte (ADCDH) 124
- ADC data low bits (ADCDL) 125
- flash control (FCTL) 137, 143, 144
- flash high and low byte (FFREQH and FREEQL) 139
- flash page select (FPS) 138, 139
- flash status (FSTAT) 137
- GPIO port A-H address (PxADDR) 44
- GPIO port A-H alternate function sub-registers 46
- GPIO port A-H control address (PxCTL) 45
- GPIO port A-H data direction sub-registers 45
- OCD control 161
- OCD status 163
- UARTx baud rate high byte (UxBRH) 110
- UARTx baud rate low byte (UxBRL) 110
- UARTx Control 0 (UxCTL0) 107, 110
- UARTx control 1 (UxCTL1) 108
- UARTx receive data (UxRXD) 105
- UARTx status 0 (UxSTAT0) 105
- UARTx status 1 (UxSTAT1) 106
- UARTx transmit data (UxTXD) 104
- Watchdog Timer control (WDTCTL) 90, 128
- watch-dog timer control (WDTCTL) 167
- Watchdog Timer reload high byte (WDTH) 91
- Watchdog Timer reload low byte (WDTL) 91
- Watchdog Timer reload upper byte (WD-TU) 91
- register file 13
- register pair 173
- register pointer 174
- reset
 - and stop mode characteristics 22
 - and stop mode recovery 21
 - carry flag 176
 - sources 22
- RET 178
- return 178
- RL 178
- RLC 178
- rotate and shift instructions 178
- rotate left 178
- rotate left through carry 178
- rotate right 178
- rotate right through carry 178
- RP 174
- RR 173, 178
- rr 173
- RRC 178
- S**
- SBC 175
- SCF 176, 177
- second opcode map after 1FH 191
- set carry flag 176, 177
- set register pointer 177
- shift right arithmetic 179
- shift right logical 179
- signal descriptions 9
- single-sho conversion (ADC) 119
- software trap 178
- source operand 174
- SP 174
- SRA 179
- src 174
- SRL 179
- SRP 177
- stack pointer 174
- STOP 177
- STOP mode 31, 177
- Stop Mode Recovery
 - sources 26
 - using a GPIO port pin transition 27, 28
 - using Watchdog Timer time-out 27
- SUB 175
- subtract 175
- subtract - extended addressing 175
- subtract with carry 175
- subtract with carry - extended addressing 175
- SUBX 175
- SWAP 179
- swap nibbles 179
- symbols, additional 174