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	22
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
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/z8f0423sj005sc

Interrupt Controller

The interrupt controller on the Z8 Encore! XP® F0823 Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 unique interrupt vectors
  - 12 GPIO port pin interrupt sources (two are shared)
  - 8 on-chip peripheral interrupt sources (two are shared)
- Flexible GPIO interrupts
  - Eight selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Watchdog Timer can be configured to generate an interrupt

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt controller has no effect on operation. For more information on interrupt servicing by the eZ8 CPU, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com.

Interrupt Vector Listing

Table 33 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

Note: *Some port interrupts are not available on the 8- and 20-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.*

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

External Driver Enable

The UART provides a Driver Enable (DE) signal for off-chip bus transceivers. This feature reduces the software overhead associated with using a GPIO pin to control the transceiver when communicating on a multi-transceiver bus, such as RS-485.

Driver Enable is an active High signal that envelopes the entire transmitted data frame including parity and Stop bits as displayed in Figure 14. The Driver Enable signal asserts when a byte is written to the UART Transmit Data register. The Driver Enable signal asserts at least one UART bit period and no greater than two UART bit periods before the Start bit is transmitted. This allows a setup time to enable the transceiver. The Driver Enable signal deasserts one system clock period after the final Stop bit is transmitted. This one system clock delay allows both time for data to clear the transceiver before disabling it, as well as the ability to determine if another character follows the current character. In the event of back to back characters (new data must be written to the Transmit Data Register before the previous character is completely transmitted) the DE signal is not deasserted between characters. The DEPOL bit in the UART Control Register 1 sets the polarity of the Driver Enable signal.

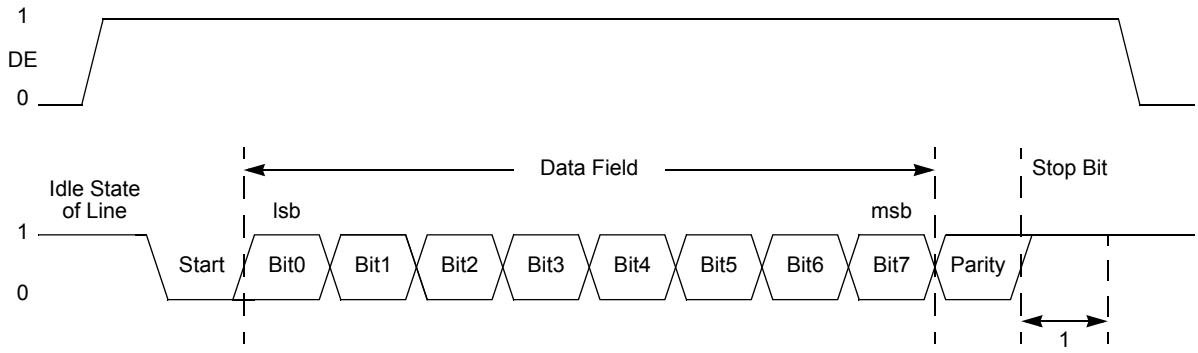


Figure 14. UART Driver Enable Signal Timing (shown with 1 Stop Bit and Parity)

The Driver Enable to Start bit setup time is calculated as follows:

$$\left(\frac{1}{\text{Baud Rate (Hz)}}\right) \leq \text{DE to Start Bit Setup Time (s)} \leq \left(\frac{2}{\text{Baud Rate (Hz)}}\right)$$

UART Interrupts

The UART features separate interrupts for the transmitter and the receiver. In addition, when the UART primary functionality is disabled, the Baud Rate Generator can also function as a basic timer with interrupt capability.

Endec, and passed to the UART. Communication is half-duplex, which means simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART's baud rate generator and supports IrDA standard baud rates from 9600 baud to 115.2 kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the Infrared Endec. The Infrared Endec data rate is calculated using the following equation:

$$\text{Infrared Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR_TXD) that drives the infrared transceiver. Each UART/Infrared data bit is 16 clocks wide. If the data to be transmitted is 1, the IR_TXD signal remains low for the full 16 clock period. If the data to be transmitted is 0, the transmitter first outputs a 7 clock low period, followed by a 3 clock high pulse. Finally, a 6 clock low pulse is output to complete the full 16 clock data period. Figure 17 displays IrDA data transmission. When the Infrared Endec is enabled, the UART's TXD signal is internal to Z8 Encore! XP® F0823 Series products while the IR_TXD signal is output through the TXD pin.

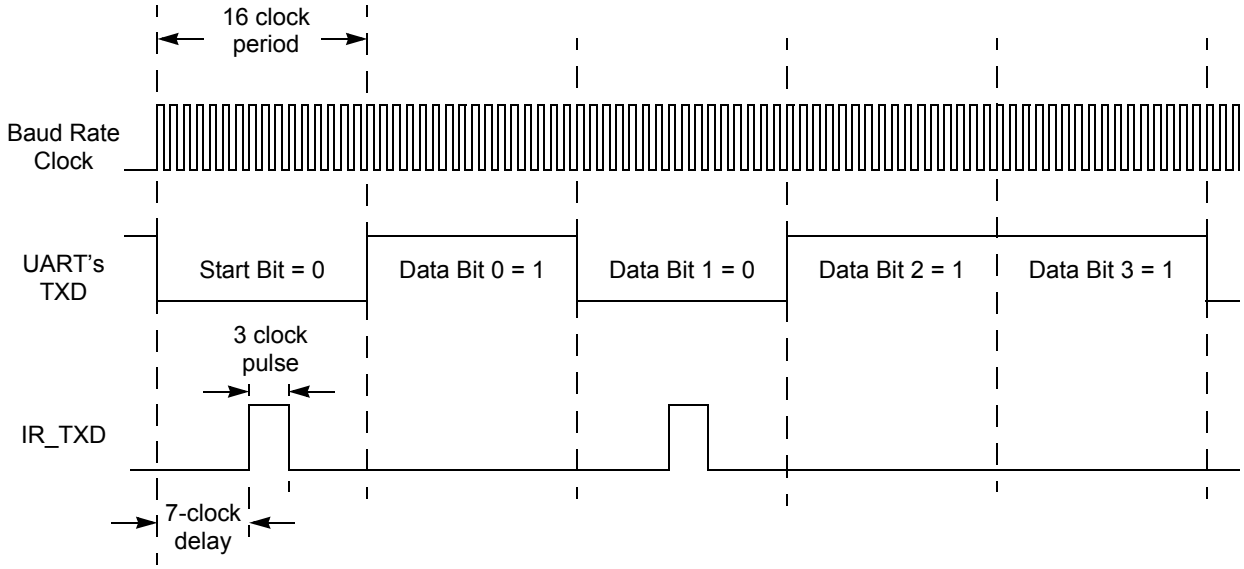


Figure 17. Infrared Data Transmission

5. When the conversion is complete, the ADC control logic performs the following operations:
   - 11-bit two's-complement result written to {ADCD_H[7:0], ADCD_L[7:5]}.
   - CEN resets to 0 to indicate the conversion is complete.
6. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

Continuous Conversion

When configured for continuous conversion, the ADC continuously performs an analog-to-digital conversion on the selected analog input. Each new data value over-writes the previous value stored in the ADC Data registers. An interrupt is generated after each conversion.

Caution: *In CONTINUOUS mode, ADC updates are limited by the input signal bandwidth of the ADC and the latency of the ADC and its digital filter. Step changes at the input are not detected at the next output from the ADC. The response of the ADC (in all modes) is limited by the input signal bandwidth and the latency.*

Follow the steps below for setting up the ADC and initiating continuous conversion:

1. Enable the acceptable analog input by configuring the general-purpose I/O pins for alternate function. This action disables the digital input and output driver.
2. Write the ADC Control/Status Register 1 to configure the ADC:
   - Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in the ADC Control/Status Register 1.
3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control register can be written simultaneously:
   - Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
   - Set CONT to 1 to select continuous conversion.
   - If the internal VREF must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
   - Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in ADC Control Register 0.
   - Set CEN to 1 to start the conversions.

Operation

The Flash Controller programs and erases Flash memory. The Flash Controller provides the proper Flash controls and timing for Byte Programming, Page Erase, and Mass Erase of Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels.

The Flowchart in Figure 21 displays basic Flash Controller operation. The following sub-sections provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select Page Erase, and Mass Erase) displayed in Figure 21.

Table 78. Flash Code Protection Using the Flash Option Bits

FWP	Flash Code Protection Description
0	Programming and erasing disabled for all of Flash Program Memory. In user code programming, Page Erase, and Mass Erase are all disabled. Mass Erase is available through the On-Chip Debugger.
1	Programming, Page Erase, and Mass Erase are enabled for all of Flash Program Memory.

Flash Code Protection Using the Flash Controller

At Reset, the Flash Controller locks to prevent accidental program or erasure of the Flash memory. To program or erase the Flash memory, first write the Page Select Register with the target page. Unlock the Flash Controller by making two consecutive writes to the Flash Control register with the values 73H and 8CH, sequentially. The Page Select Register must be rewritten with the same page previously stored there. If the two Page Select writes do not match, the controller reverts to a locked state. If the two writes match, the selected page becomes active. For more details, see Figure 21.

After unlocking a specific page, you can enable either Page Program or Erase. Writing the value 95H causes a Page Erase only if the active page resides in a sector that is not protected. Any other value written to the Flash Control register locks the Flash Controller. Mass Erase is not allowed in the user code but only in through the Debug Port.

After unlocking a specific page, you can also write to any byte on that page. After a byte is written, the page remains unlocked, allowing for subsequent writes to other bytes on the same page. Further writes to the Flash Control Register cause the active page to revert to a locked state.

Sector Based Flash Protection

The final protection mechanism is implemented on a per-sector basis. The Flash memories of Z8 Encore! XP devices are divided into maximum number of 8 sectors. A sector is 1/8 of the total size of the Flash memory, unless this value is smaller than the page size, in which case the sector and page sizes are equal.

The Sector Protect Register controls the protection state of each Flash sector. This register is shared with the Page Select Register. It is accessed by writing 73H followed by 5EH to the Flash controller. The next write to the Flash Control Register targets the Sector Protect Register.

The Sector Protect Register is initialized to 0 on reset, putting each sector into an unprotected state. When a bit in the Sector Protect Register is written to 1, the corresponding sector can no longer be written or erased by the CPU. External Flash programming through the OCD or via the Flash Controller Bypass mode are unaffected. After

Table 83. Flash Frequency High Byte Register (FFREQH)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQH							
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	FFAH							

FFREQH—Flash Frequency High Byte
High byte of the 16-bit Flash Frequency value

Table 84. Flash Frequency Low Byte Register (FFREQL)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQL							
RESET	0							
R/W	R/W							
ADDR	FFBH							

FFREQL—Flash Frequency Low Byte
Low byte of the 16-bit Flash Frequency value

Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

Reserved—R/W bits must be 1 during writes; 1 when read.

VBO_AO—Voltage Brownout Protection Always ON
0 = Voltage Brownout Protection can be disabled in STOP mode to reduce total power consumption. For the block to be disabled, the power control register bit must also be written (see Power Control Register 0 on page 32).
1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

FRP—Flash Read Protect
0 = User program code is inaccessible. Limited control features are available through the On-Chip Debugger.
1 = User program code is accessible. All On-Chip Debugger commands are enabled. This setting is the default for unprogrammed (erased) Flash.

Reserved—Must be 1

FWP—Flash Write Protect
This Option Bit provides Flash Program Memory protection:
0 = Programming and erasure disabled for all of Flash Program Memory. Programming, Page Erase, and Mass Erase through User Code is disabled. Mass Erase is available using the On-Chip Debugger.
1 = Programming, Page Erase, and Mass Erase are enabled for all of Flash program memory.

Flash Program Memory Address 0001H

Table 88. Flash Options Bits at Program Memory Address 0001H

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved			XTLDIS	Reserved			
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 0001H							

Note: U = Unchanged by Reset. R/W = Read/Write.

Reserved—R/W must be 1 during writes; 1 when read

XTLDIS—State of Crystal Oscillator at Reset

- **Read Program Memory CRC (0EH)**—The Read Program Memory Cyclic Redundancy Check (CRC) command computes and returns the CRC of Program Memory using the 16-bit CRC-CCITT polynomial. If the device is not in DEBUG mode, this command returns FFFFH for the CRC value. Unlike most other OCD Read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads the Program Memory, calculates the CRC value, and returns the result. The delay is a function of the Program Memory size and is approximately equal to the system clock period multiplied by the number of bytes in the Program Memory.

  ```
  DBG ← 0EH
  DBG → CRC[15:8]
  DBG → CRC[7:0]
  ```

- **Step Instruction (10H)**—The Step Instruction command steps one assembly instruction at the current Program Counter (PC) location. If the device is not in DEBUG mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

  ```
  DBG ← 10H
  ```

- **Stuff Instruction (11H)**—The Stuff Instruction command steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0-4 bytes of the instruction are read from Program Memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a Breakpoint. If the device is not in DEBUG mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

  ```
  DBG ← 11H
  DBG ← opcode[7:0]
  ```

- **Execute Instruction (12H)**—The Execute Instruction command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over breakpoints. The number of bytes to send for the instruction depends on the opcode. If the device is not in DEBUG mode or the Flash Read Protect Option bit is enabled, this command reads and discards one byte.

  ```
  DBG ← 12H
  DBG ← 1-5 byte opcode
  ```

On-Chip Debugger Control Register Definitions

OCD Control Register

The OCD Control register controls the state of the OCD. This register is used to enter or exit DEBUG mode and to enable the BRK instruction. It also resets Z8 Encore! XP® F0823 Series device.

Caution: *Unintentional accesses to the oscillator control register can actually stop the chip by switching to a non-functioning oscillator. To prevent this condition, the oscillator control block employs a register unlocking/locking scheme.*

OSC Control Register Unlocking/Locking

To write the oscillator control register, unlock it by making two writes to the OSCCTL register with the values E7H followed by 18H. A third write to the OSCCTL register changes the value of the actual register and returns the register to a locked state. Any other sequence of oscillator control register writes has no effect. The values written to unlock the register must be ordered correctly, but are not necessarily consecutive. It is possible to write to or read from other registers within the unlocking/locking operation.

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

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

Clock Failure Detection and Recovery

Primary Oscillator Failure

Z8 Encore! XP® F0823 Series devices can generate non-maskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer oscillator to drive the system clock. The Watchdog Timer oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer is the primary oscillator. It is also unavailable if the Watchdog Timer oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function outlined in the Watchdog Timer on page 87.

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

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as ‘destination, source’. After assembly, the object code usually has the operands in the order ‘source, destination’, but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. You must follow this binary format if you prefer manual program coding or intend to implement your own assembler.

Example 1

If the contents of Registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 103. Assembly Language Syntax Example 1

Assembly Language Code	ADD	43H,	08H	(ADD dst, src)
Object Code	04	08	43	(OPC src, dst)

Example 2

In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0–R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is:

Table 104. Assembly Language Syntax Example 2

Assembly Language Code	ADD	43H,	R8	(ADD dst, src)
Object Code	04	E8	43	(OPC src, dst)

See the device-specific Product Specification to determine the exact register file range available. The register file size varies, depending on the device type.

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status flags, and address modes are represented by a notational shorthand that is described in Table 105.

Table 112. Logical Instructions (Continued)

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

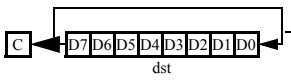
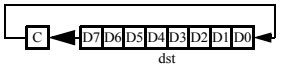
Table 113. Program Control Instructions

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

Table 114. Rotate and Shift Instructions

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

Table 115. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
OR dst, src	dst ← dst OR src	r	r	42	–	*	*	0	–	–	2	3
		r	Ir	43							2	4
		R	R	44							3	3
		R	IR	45							3	4
		R	IM	46							3	3
		IR	IM	47							3	4
ORX dst, src	dst ← dst OR src	ER	ER	48	–	*	*	0	–	–	4	3
		ER	IM	49							4	3
POP dst	dst ← @SP SP ← SP + 1	R		50	–	–	–	–	–	–	2	2
		IR		51							2	3
POPX dst	dst ← @SP SP ← SP + 1	ER		D8	–	–	–	–	–	–	3	2
PUSH src	SP ← SP – 1 @SP ← src	R		70	–	–	–	–	–	–	2	2
		IR		71							2	3
		IM		IF70							3	2
PUSHX src	SP ← SP – 1 @SP ← src	ER		C8	–	–	–	–	–	–	3	2
RCF	C ← 0			CF	0	–	–	–	–	–	1	2
RET	PC ← @SP SP ← SP + 2			AF	–	–	–	–	–	–	1	4
RL dst		R		90	*	*	*	*	–	–	2	2
		IR		91							2	3
RLC dst		R		10	*	*	*	*	–	–	2	2
		IR		11							2	3

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

Table 119. Power Consumption

Symbol	Parameter	V_{DD} = 2.7 V to 3.6 V			Units	Conditions
		Typical[1]	Maximum[2] Std Temp	Maximum[3] Ext Temp		
I_{DD} Stop	Supply Current in STOP Mode	0.1	2	7.5	µA	No peripherals enabled. All pins driven to V_{DD} or V_{SS}.
I_{DD} Halt	Supply Current in HALT Mode (with all peripherals disabled)	35	55	65	µA	32 kHz
		520	630	700	µA	5.5 MHz
I_{DD}	Supply Current in ACTIVE Mode (with all peripherals disabled)	2.8	4.5	4.8	mA	32 kHz
		4.5	5.2	5.2	mA	5.5 MHz
I_{DD} WDT	Watchdog Timer Supply Current	0.9	1.0	1.1	µA	
I_{DD} IPO	Internal Precision Oscillator Supply Current	350	500	550	µA	
I_{DD} VBO	Voltage Brownout Supply Current	50			µA	For 20-/28-pin devices (VBO only); see Note 4
						For 8-pin devices; See Note 4
I_{DD} ADC	Analog-to-Digital Converter Supply Current (with External Reference)	2.8	3.1	3.2	mA	32 kHz
		3.1	3.6	3.7	mA	5.5 MHz
		3.3	3.7	3.8	mA	10 MHz
		3.7	4.2	4.3	mA	20 MHz
I_{DD} ADCRef	ADC Internal Reference Supply Current	0			µA	See Note 4
I_{DD} CMP	Comparator supply Current	150	180	190	µA	See Note 4

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 1 KB Flash, 10-Bit Analog-to-Digital Converter								
Standard Temperature: 0 °C to 70 °C								
Z8F0123PB005SC	1 KB	256 B	6	12	2	4	1	PDIP 8-pin package
Z8F0123QB005SC	1 KB	256 B	6	12	2	4	1	QFN 8-pin package
Z8F0123SB005SC	1 KB	256 B	6	12	2	4	1	SOIC 8-pin package
Z8F0123SH005SC	1 KB	256 B	16	18	2	7	1	SOIC 20-pin package
Z8F0123HH005SC	1 KB	256 B	16	18	2	7	1	SSOP 20-pin package
Z8F0123PH005SC	1 KB	256 B	16	18	2	7	1	PDIP 20-pin package
Z8F0123SJ005SC	1 KB	256 B	22	18	2	8	1	SOIC 28-pin package
Z8F0123HJ005SC	1 KB	256 B	22	18	2	8	1	SSOP 28-pin package
Z8F0123PJ005SC	1 KB	256 B	22	18	2	8	1	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C								
Z8F0123PB005EC	1 KB	256 B	6	12	2	4	1	PDIP 8-pin package
Z8F0123QB005EC	1 KB	256 B	6	12	2	4	1	QFN 8-pin package
Z8F0123SB005EC	1 KB	256 B	6	12	2	4	1	SOIC 8-pin package
Z8F0123SH005EC	1 KB	256 B	16	18	2	7	1	SOIC 20-pin package
Z8F0123HH005EC	1 KB	256 B	16	18	2	7	1	SSOP 20-pin package
Z8F0123PH005EC	1 KB	256 B	16	18	2	7	1	PDIP 20-pin package
Z8F0123SJ005EC	1 KB	256 B	22	18	2	8	1	SOIC 28-pin package
Z8F0123HJ005EC	1 KB	256 B	22	18	2	8	1	SSOP 28-pin package
Z8F0123PJ005EC	1 KB	256 B	22	18	2	8	1	PDIP 28-pin package
Replace C with G for Lead-Free Packaging								

D

E

F

L

M

N

O