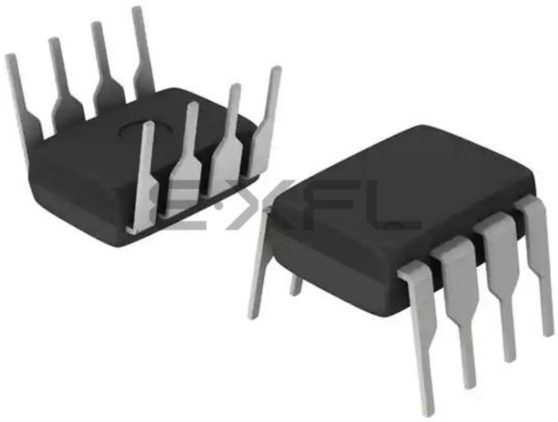


Welcome to E-XFL.COM

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	Active
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	6
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	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0123pb005sg

Note: *Analog input alternate functions (ANA) are not available on Z8F0x13 devices.

Signal Descriptions

Table 3 lists the Z8 Encore! XP F0823 Series signals. To determine the signals available for the specific package styles, see **the** Pin Configurations **section on page 7**.

Table 3. Signal Descriptions

Signal Mnemonic	I/O	Description
General-Purpose I/O Ports A–D		
PA[7:0]	I/O	Port A. These pins are used for general-purpose I/O.
PB[7:0][1]	I/O	Port B. These pins are used for general-purpose I/O. PB6 and PB7 are available only in those devices without an ADC.
PC[7:0]	I/O	Port C. These pins are used for general-purpose I/O.
UART Controllers		
TXD0	O	Transmit Data. This signal is the transmit output from the UART and IrDA.
RXD0	I	Receive Data. This signal is the receive input for the UART and IrDA.
$\overline{\text{CTS0}}$	I	Clear To Send. This signal is the flow control input for the UART.
DE	O	Driver Enable. This signal allows automatic control of external RS-485 drivers. This signal is approximately the inverse of the TXE (Transmit Empty) bit in the UART Status 0 Register. The DE signal can be used to ensure the external RS-485 driver is enabled when data is transmitted by the UART.
Timers		
T0OUT/T1OUT	O	Timer Output 0–1. These signals are output from the timers.
$\overline{\text{T0OUT}}$/$\overline{\text{T1OUT}}$	O	Timer Complement Output 0–1. These signals are output from the timers in PWM DUAL OUTPUT Mode.
T0IN/T1IN	I	Timer Input 0–1. These signals are used as the capture, gating and counter inputs. The T0IN signal is multiplexed $\overline{\text{T0OUT}}$ signals.
Comparator		
CINP/CINN	I	Comparator Inputs. These signals are the positive and negative inputs to the comparator.

Notes:
1. PB6 and PB7 are only available in 28-pin packages without ADC. In 28-pin packages with ADC, they are replaced by AV_{DD} and AV_{SS}.
2. The AV_{DD} and AV_{SS} signals are available only in 28-pin packages with ADC. They are replaced by PB6 and PB7 on 28-pin packages without ADC.

Table 6. Z8 Encore! XP F0823 Series Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
Z8F0123 and Z8F0113 Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–003D	Oscillator Fail Traps*
003E–03FF	Program Memory

Note: *See the Trap and Interrupt Vectors in Order of Priority section on page 55 for a list of the interrupt vectors and traps.

Data Memory

Z8 Encore! XP F0823 Series does not use the eZ8 CPU's 64KB Data Memory address space.

Flash Information Area

Table 7 lists the F0823 Series Flash Information Area. This 128B Information Area is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash Information Area is mapped into the Program Memory and overlays the 128 bytes at addresses FE00H to FF7FH. When the Information Area access is enabled, all reads from these Program Memory addresses return the Information Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

Table 7. F0823 Series Flash Memory Information Area Map

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog Option Bits.
FE40–FE53	Part Number. 20-character ASCII alphanumeric code Left-justified and filled with FH.
FE54–FE5F	Reserved.
FE60–FE7F	Zilog Calibration Data.
FE80–FFFF	Reserved.

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No.
LED Controller (cont'd)				
F84	LED Drive Level Low Byte	LEDLVLL	00	53
F85	Reserved	—	XX	
Oscillator Control				
F86	Oscillator Control	OSCCTL	A0	172
F87–F8F	Reserved	—	XX	
Comparator 0				
F90	Comparator 0 Control	CMP0	14	133
F91–FBF	Reserved	—	XX	
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	59
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	62
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	62
FC3	Interrupt Request 1	IRQ1	00	60
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	64
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	64
FC6	Interrupt Request 2	IRQ2	00	61
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	65
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	66
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	67
FCE	Shared Interrupt Select	IRQSS	00	67
FCF	Interrupt Control	IRQCTL	00	68
GPIO Port A				
FD0	Port A Address	PAADDR	00	40
FD1	Port A Control	PACTL	00	42
FD2	Port A Input Data	PAIN	XX	43
FD3	Port A Output Data	PAOUT	00	43
GPIO Port B				
FD4	Port B Address	PBADDR	00	40
FD5	Port B Control	PBCTL	00	42

Note: XX=Undefined.

clock and reset signals, the required reset duration can be as short as three clock periods and as long as four. A reset pulse three clock cycles in duration might trigger a reset; a pulse four cycles in duration always triggers a reset.

While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z8 Encore! XP F0823 Series devices remain in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the System Reset time-out, the device exits the Reset state on the system clock rising edge following $\overline{\text{RESET}}$ pin deassertion. Following a System Reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the WDT Control (WDTCTL) register is set to 1.

External Reset Indicator

During System Reset or when enabled by the GPIO logic (see **the** Port A–C Control Registers **section on page 42**), the $\overline{\text{RESET}}$ pin functions as an open-drain (active Low) reset mode indicator in addition to the input functionality. This reset output feature allows an Z8 Encore! XP F0823 Series device to reset other components to which it is connected, even if that reset is caused by internal sources such as POR, VBO, or WDT events.

After an internal reset event occurs, the internal circuitry begins driving the $\overline{\text{RESET}}$ pin Low. The $\overline{\text{RESET}}$ pin is held Low by the internal circuitry until the appropriate delay listed in Table 9 has elapsed.

On-Chip Debugger Initiated Reset

A POR is initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The OCD block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the System Reset. Following the System Reset, the POR bit in the Reset Status (RSTSTAT) Register is set.

Stop Mode Recovery

The device enters into STOP Mode when eZ8 CPU executes a `STOP` instruction. For more details about STOP Mode, see **the** Low-Power Modes **section on page 30**. During Stop Mode Recovery, the CPU is held in reset for 66 IPO cycles if the crystal oscillator is disabled or 5000 cycles if it is enabled. The SMR delay also included the time required to start up the IPO.

Stop Mode Recovery does not affect on-chip registers other than the Watchdog Timer Control Register (WDTCTL) and the Oscillator Control Register (OSCCTL). After any Stop Mode Recovery, the IPO is enabled and selected as the system clock. If another system clock source is required or IPO disabling is required, the Stop Mode Recovery code must reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

Port A–C Input Data Registers

Reading from the Port A–C Input Data registers (Table 30) returns the sampled values from the corresponding port pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8- and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

Table 30. Port A–C Input Data Registers (PxIN)

Bit	**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
Address	FD2H, FD6H, FDAH							

Bit	**Description**
[7:0] PxIN	**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).

Note: x indicates the specific GPIO port pin number (7–0).

Interrupt Control Register

The Interrupt Control (IRQCTL) Register (Table 50) contains the master enable bit for all interrupts.

Table 50. Interrupt Control Register (IRQCTL)

Bit	7	6	5	4	3	2	1	0
Field	IRQE	Reserved						
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R
Address	FCFH							

Bit	Description
[7] IRQE	**Interrupt Request Enable** This bit is set to 1 by executing an Enable Interrupts (EI) or Interrupt Return (IRET) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, reset or by a direct register write of a 0 to this bit. 0 = Interrupts are disabled. 1 = Interrupts are enabled.
[6:0]	**Reserved** These bits are reserved and must be programmed to 0000000 when read.

input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.

6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

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

CAPTURE RESTART Mode

In CAPTURE RESTART Mode, the current timer count value is recorded when the acceptable external Timer Input transition occurs. The capture count value is written to the Timer PWM High and Low Byte registers. The timer input is the system clock. The `TPOL` bit in the Timer Control Register determines if the capture occurs on a rising edge or a falling edge of the Timer Input signal. When the capture event occurs, an interrupt is generated and the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The `INPCAP` bit in TxCTL1 Register is set to indicate the timer interrupt is because of an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The `INPCAP` bit in TxCTL1 Register is cleared to indicate the timer interrupt is not caused by an input capture event.

Observe the following steps to configure a timer for CAPTURE RESTART Mode and initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE RESTART Mode; setting the mode also involves writing to `TMODEHI` bit in TxCTL1 Register
   - Set the prescale value
   - Set the capture edge (rising or falling) for the Timer Input

2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.

4. Clear the Timer PWM High and Low Byte registers to `0000H`. Clearing these registers allows the software to determine if interrupts were generated by either a capture or a reload event. If the PWM High and Low Byte registers still contain `0000H` after the interrupt, the interrupt was generated by a reload.
5. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.
6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

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

COMPARE Mode

In COMPARE Mode, the timer counts up to the 16-bit maximum compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the compare value, the timer generates an interrupt and counting continues (the timer value is not reset to `0001H`). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) upon compare.

If the Timer reaches `FFFFH`, the timer rolls over to `0000H` and continue counting. Observe the following steps to configure a timer for COMPARE Mode and to initiate the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COMPARE Mode
   - Set the prescale value
   - Set the initial logic level (High or Low) for the Timer Output alternate function, if appropriate
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the compare value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers.

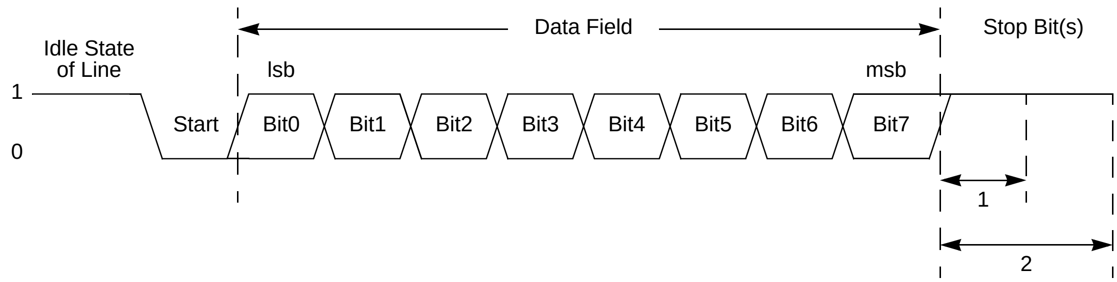


Figure 11. UART Asynchronous Data Format without Parity

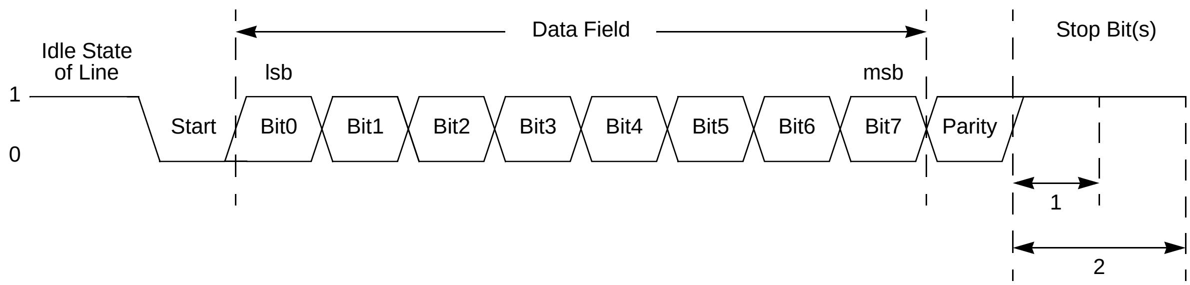


Figure 12. UART Asynchronous Data Format with Parity

Transmitting Data Using the Polled Method

Observe the following steps to transmit data using the polled method of operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the required baud rate.
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, if MULTIPROCESSOR Mode is appropriate, to enable MULTIPROCESSOR (9-bit) Mode functions.
4. Set the Multiprocessor Mode Select (`MPEN`) bit to enable MULTIPROCESSOR Mode.
5. Write to the UART Control 0 Register to:
   – Set the transmit enable bit (`TEN`) to enable the UART for data transmission
   – Set the parity enable bit (`PEN`), if parity is appropriate and MULTIPROCESSOR Mode is not enabled, and select either even or odd parity (`PSEL`)

- Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{CTS}$ pin

6. Check the `TDRE` bit in the UART Status 0 Register to determine if the Transmit Data Register is empty (indicated by a 1). If empty, continue to Step 7. If the Transmit Data Register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data Register becomes available to receive new data.
7. Write the UART Control 1 Register to select the outgoing address bit.
8. Set the Multiprocessor Bit Transmitter (`MPBT`) if sending an address byte, clear it if sending a data byte.
9. 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.
10. Make any changes to the Multiprocessor Bit Transmitter (`MPBT`) value, if appropriate and MULTIPROCESSOR Mode is enabled,.
11. To transmit additional bytes, return to Step 5.

Transmitting Data Using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data Register to accept new data for transmission. Observe the following steps to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a `DI` instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the acceptable priority.
5. Write to the UART Control 1 Register to enable MULTIPROCESSOR (9-bit) Mode functions, if MULTIPROCESSOR Mode is appropriate.
6. Set the MULTIPROCESSOR Mode Select (`MPEN`) to Enable MULTIPROCESSOR Mode.
7. Write to the UART Control 0 Register to:
   - Set the transmit enable bit (`TEN`) to enable the UART for data transmission.
   - Enable parity, if appropriate and if MULTIPROCESSOR Mode is not enabled, and select either even or odd parity.

Bit	Description (Continued)
[1] STOP	**Stop Bit Select** 0 = The transmitter sends one stop bit. 1 = The transmitter sends two stop bits.
[0] LBEN	**Loop Back Enable** 0 = Normal operation. 1 = All transmitted data is looped back to the receiver.

Table 69. UART Control 1 Register (U0CTL1)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	MPMD[1]	MPEN	MPMD[0]	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
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
Address	F43H							

Bit	Description
[7,5] MPMD[1:0]	**MULTIPROCESSOR Mode** If MULTIPROCESSOR (9-bit) Mode is enabled. 00 = The UART generates an interrupt request on all received bytes (data and address). 01 = The UART generates an interrupt request only on received address bytes. 10 = The UART generates an interrupt request when a received address byte matches the value stored in the Address Compare Register and on all successive data bytes until an address mismatch occurs. 11 = The UART generates an interrupt request on all received data bytes for which the most recent address byte matched the value in the Address Compare Register.
[6] MPEN	**MULTIPROCESSOR (9-bit) Enable** This bit is used to enable MULTIPROCESSOR (9-bit) Mode. 0 = Disable MULTIPROCESSOR (9-bit) Mode. 1 = Enable MULTIPROCESSOR (9-bit) Mode.
[4] MPBT	**Multiprocessor Bit Transmit** This bit is applicable only when MULTIPROCESSOR (9-bit) Mode is enabled. The 9th bit is used by the receiving device to determine if the data byte contains address or data information. 0 = Send a 0 in the multiprocessor bit location of the data stream (data byte). 1 = Send a 1 in the multiprocessor bit location of the data stream (address byte).
[3] DEPOL	**Driver Enable Polarity** 0 = DE signal is Active High. 1 = DE signal is Active Low.

4. When the first conversion in continuous operation is complete (after 5129 system clock cycles, plus the 40 cycles for power-up, if necessary), the ADC control logic performs the following operations:
   – CEN resets to 0 to indicate the first conversion is complete. CEN remains 0 for all subsequent conversions in continuous operation
   – An interrupt request is sent to the Interrupt Controller to indicate the conversion is complete

5. The ADC writes a new data result every 256 system clock cycles. For each completed conversion, the ADC control logic performs the following operations:
   – Writes the 11-bit two's complement result to {ADCD_H[7:0], ADCD_L[7:5]}
   – An interrupt request to the Interrupt Controller denoting conversion complete

6. To disable continuous conversion, clear the CONT bit in the ADC Control Register to 0.

Interrupts

The ADC is able to interrupt the CPU whenever a conversion has been completed and the ADC is enabled.

When the ADC is disabled, an interrupt is not asserted; however, an interrupt pending when the ADC is disabled is not cleared.

Calibration and Compensation

Z8 Encore! XP F0823 Series ADC can be factory calibrated for offset error and gain error, with the compensation data stored in Flash memory. Alternatively, user code can perform its own calibration, storing the values into Flash themselves.

Factory Calibration

Devices that have been factory calibrated contain nine bytes of calibration data in the Flash option bit space. This data consists of three bytes for each reference type. For a list of input modes for which calibration data exists, see the Zilog Calibration Data section on page 152. There is 1 byte for offset, and there are 2 bytes for gain correction.

User Calibration

If you have precision references available, its own external calibration can be performed, storing the values into Flash themselves.

Comparator Control Register Definition

The Comparator Control Register (CMPCTL) configures the comparator inputs and sets the value of the internal voltage reference.

Table 78. Comparator Control Register (CMP0)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	INPSEL	INNSEL	REFLVL				Reserved	
RESET	0	0	0	1	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F90H							

Bit	**Description**
[7] INPSEL	**Signal Select for Positive Input** 0 = GPIO pin used as positive comparator input. 1 = temperature sensor used as positive comparator input.
[6] INNSEL	**Signal Select for Negative Input** 0 = internal reference disabled, GPIO pin used as negative comparator input. 1 = internal reference enabled as negative comparator input.
[5:2] REFLVL	**Internal Reference Voltage Level** 0000 = 0.0V. 0001 = 0.2V. 0010 = 0.4V. 0011 = 0.6V. 0100 = 0.8V. 0101 = 1.0V (Default). 0110 = 1.2V. 0111 = 1.4V. 1000 = 1.6V. 1001 = 1.8V. 1010–1111 = Reserved. **Note:** This reference is independent of the ADC voltage reference.
[1:0]	**Reserved** These bits are reserved; R/W bits must be programmed to 00 during writes and to 00 when read.

.

Table 80. 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 Flash memory size unless this value is smaller than the page size – in which case, the sector and page sizes are equal. On Z8 Encore! F0823 Series devices, the sector size is varied according to the Flash memory configuration shown in Table 79 on page 134.

The Flash Sector Protect Register can be configured to prevent sectors from being programmed or erased. After a sector is protected, it cannot be unprotected by user code. The Flash Sector Protect Register is cleared after reset, and any previously-written protection values are lost. User code must write this register in their initialization routine if they prefer to enable sector protection.

The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register is accessed by writing the Flash Control Register with `5EH`. After the Flash Sector Protect Register is selected, it can be accessed at the Page Select Register address. When user code writes the Flash Sector Protect Register,

Assembly Language Source Program Example

`JP START`	; Everything after the semicolon is a comment.
`START:`	; A label called 'START'. The first instruction (`JP START`) in this ; example causes program execution to jump to the point within the ; program where the `START` label occurs.
`LD R4, R7`	; A Load (LD) instruction with two operands. The first operand, ; Working Register R4, is the destination. The second operand, ; Working Register R7, is the source. The contents of R7 is ; written into R4.
`LD 234H, #%01`	; Another Load (LD) instruction with two operands. ; The first operand, Extended Mode Register Address `234H`, ; identifies the destination. The second operand, Immediate Data ; value `01H`, is the source. The value `01H` is written into the ; Register at address `234H`.

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 shown in Table 106.

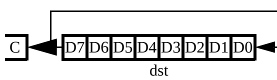
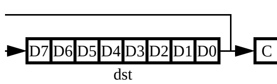
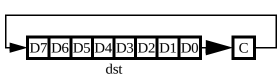
Table 106. Assembly Language Syntax Example 1

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 shown in Table 107.

Table 118. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
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
RR dst		R		E0	*	*	*	*	–	–	2	2
		IR		E1							2	3
RRC dst		R		C0	*	*	*	*	–	–	2	2
		IR		C1							2	3

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

Ordering Information

Order your F0823 Series products from Zilog using the part numbers shown in Table 135. For more information about ordering, please consult your local Zilog sales office. The Sales Location page on the Zilog website lists all regional offices.

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix

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 F0823 Series with 8 KB Flash, 10-Bit Analog-to-Digital Converter								
Standard Temperature: 0°C to 70°C								
Z8F0823PB005SG	8 KB	1 KB	6	12	2	4	1	PDIP 8-pin package
Z8F0823QB005SG	8 KB	1 KB	6	12	2	4	1	QFN 8-pin package
Z8F0823SB005SG	8 KB	1 KB	6	12	2	4	1	SOIC 8-pin package
Z8F0823SH005SG	8 KB	1 KB	16	18	2	7	1	SOIC 20-pin package
Z8F0823HH005SG	8 KB	1 KB	16	18	2	7	1	SSOP 20-pin package
Z8F0823PH005SG	8 KB	1 KB	16	18	2	7	1	PDIP 20-pin package
Z8F0823SJ005SG	8 KB	1 KB	22	18	2	8	1	SOIC 28-pin package
Z8F0823HJ005SG	8 KB	1 KB	22	18	2	8	1	SSOP 28-pin package
Z8F0823PJ005SG	8 KB	1 KB	22	18	2	8	1	PDIP 28-pin package
Extended Temperature: –40°C to 105°C								
Z8F0823PB005EG	8 KB	1 KB	6	12	2	4	1	PDIP 8-pin package
Z8F0823QB005EG	8 KB	1 KB	6	12	2	4	1	QFN 8-pin package
Z8F0823SB005EG	8 KB	1 KB	6	12	2	4	1	SOIC 8-pin package
Z8F0823SH005EG	8 KB	1 KB	16	18	2	7	1	SOIC 20-pin package
Z8F0823HH005EG	8 KB	1 KB	16	18	2	7	1	SSOP 20-pin package
Z8F0823PH005EG	8 KB	1 KB	16	18	2	7	1	PDIP 20-pin package
Z8F0823SJ005EG	8 KB	1 KB	22	18	2	8	1	SOIC 28-pin package
Z8F0823HJ005EG	8 KB	1 KB	22	18	2	8	1	SSOP 28-pin package
Z8F0823PJ005EG	8 KB	1 KB	22	18	2	8	1	PDIP 28-pin package

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

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 F0823 Series with 1 KB Flash, 10-Bit Analog-to-Digital Converter								
Standard Temperature: 0°C to 70°C								
Z8F0123PB005SG	1 KB	256 B	6	12	2	4	1	PDIP 8-pin package
Z8F0123QB005SG	1 KB	256 B	6	12	2	4	1	QFN 8-pin package
Z8F0123SB005SG	1 KB	256 B	6	12	2	4	1	SOIC 8-pin package
Z8F0123SH005SG	1 KB	256 B	16	18	2	7	1	SOIC 20-pin package
Z8F0123HH005SG	1 KB	256 B	16	18	2	7	1	SSOP 20-pin package
Z8F0123PH005SG	1 KB	256 B	16	18	2	7	1	PDIP 20-pin package
Z8F0123SJ005SG	1 KB	256 B	22	18	2	8	1	SOIC 28-pin package
Z8F0123HJ005SG	1 KB	256 B	22	18	2	8	1	SSOP 28-pin package
Z8F0123PJ005SG	1 KB	256 B	22	18	2	8	1	PDIP 28-pin package
Extended Temperature: –40°C to 105°C								
Z8F0123PB005EG	1 KB	256 B	6	12	2	4	1	PDIP 8-pin package
Z8F0123QB005EG	1 KB	256 B	6	12	2	4	1	QFN 8-pin package
Z8F0123SB005EG	1 KB	256 B	6	12	2	4	1	SOIC 8-pin package
Z8F0123SH005EG	1 KB	256 B	16	18	2	7	1	SOIC 20-pin package
Z8F0123HH005EG	1 KB	256 B	16	18	2	7	1	SSOP 20-pin package
Z8F0123PH005EG	1 KB	256 B	16	18	2	7	1	PDIP 20-pin package
Z8F0123SJ005EG	1 KB	256 B	22	18	2	8	1	SOIC 28-pin package
Z8F0123HJ005EG	1 KB	256 B	22	18	2	8	1	SSOP 28-pin package
Z8F0123PJ005EG	1 KB	256 B	22	18	2	8	1	PDIP 28-pin package

M

N

O

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.