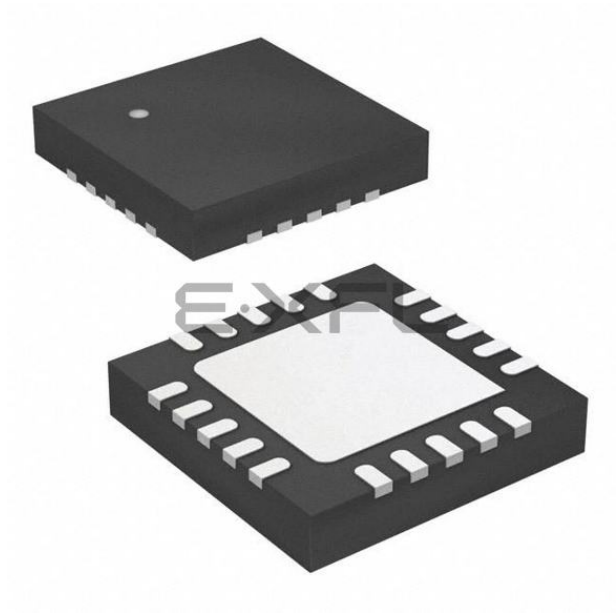

Welcome to E-XFL.COM

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	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K 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 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0830qh020sg

Table 29. Port A–C Input Data Registers (PxIN) 49
Table 30. Port A–D Output Data Register (PxOUT) 50
Table 31. LED Drive Enable (LEDEN) 51
Table 32. LED Drive Level High Register (LEDLVLH) 51
Table 33. LED Drive Level Low Register (LEDLVLL) 52
Table 34. Trap and Interrupt Vectors in Order of Priority 54
Table 35. Interrupt Request 0 Register (IRQ0) 58
Table 36. Interrupt Request 1 Register (IRQ1) 59
Table 37. Interrupt Request 2 Register (IRQ2) 60
Table 38. IRQ0 Enable and Priority Encoding 60
Table 39. IRQ0 Enable Low Bit Register (IRQ0ENL) 61
Table 40. IRQ0 Enable High Bit Register (IRQ0ENH) 61
Table 41. IRQ1 Enable and Priority Encoding 62
Table 42. IRQ1 Enable High Bit Register (IRQ1ENH) 62
Table 43. IRQ2 Enable and Priority Encoding 63
Table 44. IRQ1 Enable Low Bit Register (IRQ1ENL) 63
Table 45. IRQ2 Enable Low Bit Register (IRQ2ENL) 64
Table 46. IRQ2 Enable High Bit Register (IRQ2ENH) 64
Table 47. Interrupt Edge Select Register (IRQES) 65
Table 48. Shared Interrupt Select Register (IRQSS) 66
Table 49. Interrupt Control Register (IRQCTL) 67
Table 50. Timer 0–1 High Byte Register (TxH) 83
Table 51. Timer 0–1 Low Byte Register (TxL) 83
Table 52. Timer 0–1 Reload High Byte Register (TxRH) 85
Table 53. Timer 0–1 Reload Low Byte Register (TxRL) 85
Table 54. Timer 0–1 PWM High Byte Register (TxPWMH) 86
Table 55. Timer 0–1 PWM Low Byte Register (TxPWML) 86
Table 56. Timer 0–1 Control Register 0 (TxCTL0) 87
Table 57. Timer 0–1 Control Register 1 (TxCTL1) 88
Table 58. Watchdog Timer Approximate Time-Out Delays 92

Table 89. Trim Option Bits at 0006H (TCLKFLT) 132
Table 90. ClkFlt Delay Control Definition 133
Table 91. Write Status Byte 135
Table 92. Read Status Byte 136
Table 93. NVDS Read Time 137
Table 94. OCD Baud-Rate Limits 142
Table 95. On-Chip Debugger Command Summary 144
Table 96. OCD Control Register (OCDCTL) 149
Table 97. OCD Status Register (OCDSTAT) 150
Table 98. Oscillator Configuration and Selection 152
Table 99. Oscillator Control Register (OSCCTL) 154
Table 100. Recommended Crystal Oscillator Specifications 158
Table 101. Assembly Language Syntax Example 1 163
Table 102. Assembly Language Syntax Example 2 164
Table 103. Notational Shorthand 164
Table 104. Additional Symbols 165
Table 105. Arithmetic Instructions 166
Table 106. Bit Manipulation Instructions 167
Table 107. Block Transfer Instructions 167
Table 108. CPU Control Instructions 168
Table 109. Load Instructions 168
Table 110. Rotate and Shift Instructions 169
Table 111. Logical Instructions 169
Table 112. Program Control Instructions 169
Table 113. eZ8 CPU Instruction Summary 171
Table 114. Op Code Map Abbreviations 181
Table 115. Absolute Maximum Ratings 184
Table 116. DC Characteristics 185
Table 117. AC Characteristics 189
Table 118. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing 190

Data Memory

The Z8 Encore! F0830 Series does not use the eZ8 CPU's 64KB data memory address space.

Flash Information Area

Table 7 maps the Z8 Encore! F0830 Series Flash information area. The 128-byte 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 program memory and overlays these 128 bytes at addresses FE00H to FE7FH. When information area access is enabled, all reads from these program memory addresses return information area data rather than program memory data. Access to the Flash information area is read-only.

Table 7. Z8 Encore! F0830 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	Reserved
FE80–FFFF	Reserved

clock and reset signals, the required reset duration may be three or four clock periods. A reset pulse of three clock cycles in duration might trigger a reset and a reset pulse of four cycles in duration always triggers a reset.

While the $\overline{\text{RESET}}$ input pin is asserted low, the Z8 Encore! F0830 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 Reset Status (RSTSTAT) Register is set to 1.

External Reset Indicator

During system reset or when enabled by the GPIO logic, 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! F0830 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. See the Port A–D Control Registers section on page 41.

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 (see page 22) has elapsed.

On-Chip Debugger Initiated Reset

A Power-On Reset can be initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The OCD block is not reset, but the remainder 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 the STOP Mode when the STOP instruction is executed by the eZ8 CPU. See the Low-Power Modes chapter on page 30 for detailed STOP Mode information. During Stop Mode Recovery, the CPU is held in reset for about 66 IPO cycles if the crystal oscillator is disabled or about 5000 cycles if it is enabled.

Stop Mode Recovery does not affect the on-chip registers other than the Reset Status (RSTSTAT) Register 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.

```
LDX r0, IRQ0
AND r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 2 to clear bits in the Interrupt Request 0 Register:

Example 2. A good coding style that avoids lost interrupt requests:

```
ANDX IRQ0, MASK
```

Software Interrupt Assertion

Program code can generate interrupts directly. Writing 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: Zilog recommends not using a coding style to generate software interrupts by setting bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 3, which follows.

Example 3. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 4 to set bits in the Interrupt Request registers:

Example 4. A good coding style that avoids lost interrupt requests:

```
ORX IRQ0, MASK
```

Interrupt Control Register Definitions

The Interrupt Control registers enable individual interrupts, set interrupt priorities and indicate interrupt requests for all of the interrupts other than the Watchdog Timer interrupt, the primary oscillator fail trap and the Watchdog Oscillator fail trap interrupts.

Table 45. IRQ2 Enable High Bit Register (IRQ2ENH)

Bit	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
Address	FC7H							

Bit	Description
[7:4]	**Reserved** These registers are reserved and must be programmed to 0000.
[3] C3ENH	**Port C3 Interrupt Request Enable High Bit**
[2] C2ENH	**Port C2 Interrupt Request Enable High Bit**
[1] C1ENH	**Port C1 Interrupt Request Enable High Bit**
[0] C0ENH	**Port C0 Interrupt Request Enable High Bit**

Table 46. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	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
Address	FC8H							

Bit	Description
[7:4]	**Reserved** These registers are reserved and must be programmed to 0000.
[3] C3ENL	**Port C3 Interrupt Request Enable Low Bit**
[2] C2ENL	**Port C2 Interrupt Request Enable Low Bit**
[1] C1ENL	**Port C1 Interrupt Request Enable Low Bit**
[0] C0ENL	**Port C0 Interrupt Request Enable Low Bit**

Caution: The frequency of the comparator output signal must not exceed one-fourth the system clock frequency.

After reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Additionally, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps for configuring a timer for COMPARATOR COUNTER Mode and for initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COMPARATOR COUNTER Mode.
   - Select either the rising edge or falling edge of the comparator output signal for the count. This also sets the initial logic level (High or Low) for the timer output alternate function. However, the timer output function is not required to be enabled.
2. Write to the Timer High and Low Byte registers to set the starting count value. This action only affects the first pass in COMPARATOR COUNTER Mode. After the first timer reload in COMPARATOR COUNTER Mode, counting always begins at the reset value 0001H. Generally, in COMPARATOR COUNTER Mode, the Timer High and Low Byte registers must be written with the value 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the timer output function, configure the associated GPIO port pin for the timer output alternate function.
6. Write to the Timer Control Register to enable the timer.

In COMPARATOR COUNTER Mode, the number of comparator output transitions is calculated with the following equation:

Comparator Output Transitions = Current Count Value – Start Value

6. Write to the Timer Control Register to enable the timer.
7. Counting begins on the first appropriate transition of the timer input signal. No interrupt is generated by the first edge.

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

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

Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on Timer operation. When the timer is enabled and the Timer High Byte Register is read, the contents of the timer low byte register are placed in a holding register. A subsequent read from the timer low byte register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value when enabled. When the timers are not enabled, a read from the timer low byte register returns the actual value in the counter.

Timer Pin Signal Operation

Timer output is a GPIO port pin alternate function. The timer output is toggled every time the counter is reloaded.

The timer input can be used as a selectable counting source. It shares the same pin as the complementary timer output. When selected by the GPIO alternate function registers, this pin functions as a timer input in all modes except for the DUAL PWM OUTPUT Mode. For this mode, no timer input is available.

Timer 0–1 Control Registers

The Timer Control registers are 8-bit read/write registers that control the operation of their associated counter/timers.

Time 0–1 Control Register 0

The Timer Control 0 (TxCTL0) and Timer Control 1 (TxCTL1) registers determine the timer operating mode. These registers also include a programmable PWM deadband delay, two bits to configure the timer interrupt definition, and a status bit to identify if the most recent timer interrupt is caused by an input capture event.

Table 56. Timer 0–1 Control Register 0 (TxCTL0)

Bit	7	6	5	4	3	2	1	0
Field	TMODEHI	TICONFIG		Reserved	PWMD			INPCAP
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	F06H, F0EH							

Bit	Description
[7] TMODEHI	**Timer Mode High Bit** This bit along with the TMODE field in the TxCTL1 Register determines the operating mode of the timer. This is the most significant bit of the timer mode selection value. See the TxCTL1 Register description on the next page for additional details.
[6:5] TICONFIG	**Timer Interrupt Configuration** This field configures timer interrupt definition. 0x = Timer interrupt occurs on all of the defined reload, compare and input events. 10 = Timer interrupt occurs only on defined input capture/deassertion events. 11 = Timer interrupt occurs only on defined reload/compare events.
[4]	**Reserved** This bit is reserved and must be programmed to 0.
[3:1] PWMD	**PWM Delay Value** This field is a programmable delay to control the number of system clock cycles delay before the timer output and the timer output complement are forced to their Active state. 000 = No delay. 001 = 2 cycles delay. 010 = 4 cycles delay. 011 = 8 cycles delay. 100 = 16 cycles delay. 101 = 32 cycles delay. 110 = 64 cycles delay. 111 = 128 cycles delay.

Flash Memory

The products in the Z8 Encore! F0830 Series features either 1KB (1024 bytes with NVDS), 2KB (2048 bytes with NVDS), 4KB (4096 bytes with NVDS), 8KB (8192 bytes with NVDS) or 12KB (12288 bytes with no NVDS) of nonvolatile Flash memory with read/write/erase capability. Flash memory can be programmed and erased in-circuit by either user code or through the On-Chip Debugger.

The Flash memory array is arranged in pages with 512 bytes per page. The 512-byte page is the minimum Flash block size that can be erased. Each page is divided into eight rows of 64 bytes.

For program/data protection, Flash memory is also divided into sectors. In the Z8 Encore! F0830 Series, each sector maps to one page (for 1KB, 2KB and 4KB devices), two pages (8KB device) or three pages (12KB device).

The first two bytes of Flash program memory is used as Flash option bits. For more information, see *the* Flash Option Bits chapter on page 124.

Table 69 lists the Flash memory configuration for each device in the Z8 Encore! F0830 Series. Figures 14 through 18 display the memory arrangements for each Flash memory size.

Table 69. Z8 Encore! F0830 Series Flash Memory Configuration

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash Sector Size (bytes)
Z8F123x	12 (12,288)	24	0000H–2FFFH	1536
Z8F083x	8 (8196)	16	0000H–1FFFH	1024
Z8F043x	4 (4096)	8	0000H–0FFFH	512
Z8F023x	2 (2048)	4	0000H–07FFH	512
Z8F013x	1 (1024)	2	0000H–03FFH	512

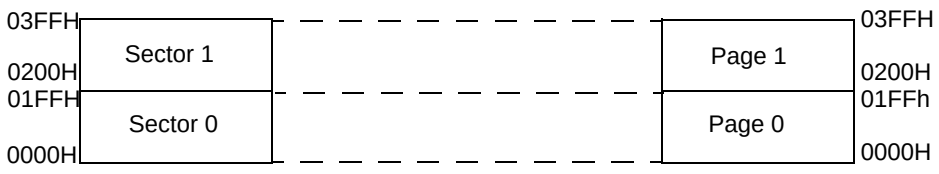

Figure 14. 1K Flash with NVDS

Page Erase

Flash memory can be erased one page (512 bytes) at a time. Page erasing Flash memory sets all bytes in that page to the value FFH. The Flash Page Select Register identifies the page to be erased. Only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95h to the Flash Control Register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the page erase operation completes. If the Page Erase operation is performed using the On-Chip Debugger, poll the Flash Status Register to determine when the Page Erase operation is complete. When the page erase is complete, the Flash Controller returns to its Locked state.

Mass Erase

Flash memory can also be mass erased using the Flash Controller, but only by using the On-Chip Debugger. Mass erasing Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked and the mass erase successfully enabled, writing the 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; instead, the control signals for Flash memory can be brought out to the GPIO pins. Bypassing the Flash Controller allows faster row programming algorithms by controlling the Flash programming signals directly.

Row programing is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of Flash memory. Mass Erase and Page Erase operations are also supported, when the Flash Controller is bypassed.

For more information about bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore!*. This document is available for download at www.zilog.com.

Flash Controller Behavior in Debug Mode

The following behavioral changes can be observed in the Flash Controller when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash write protect option bit is ignored.

Byte Read

To read a byte from the NVDS array, user code must first push the address onto the stack. User code issues a `CALL` instruction to the address of the byte-read routine (`0x2000`). At the return from the subroutine, the read byte resides in working register R0 and the read status byte resides in working register R1. The bit fields of this status byte are defined in Table 92. Additionally, the user code should pop the address byte off the stack.

The read routine uses 16 bytes of stack space in addition to the one byte of address pushed by the user code. Sufficient memory must be available for this stack usage.

Due to the Flash memory architecture, NVDS reads exhibit a nonuniform execution time. A read operation takes between 71µs and 258µs (assuming a 20MHz 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 6µs execution time.

The status byte returned by the NVDS read routine is zero for a successful read. If the status byte is nonzero, there is 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 an error.

Table 92. Read Status Byte

Bit	7	6	5	4	3	2	1	0
Field	Reserved			DE	Reserved	FE	IGADDR	Reserved
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 000.
[4] DE	**Data Error** When reading an NVDS address, if an error is found in the latest data corresponding to this NVDS address, this bit is set to 1. NVDS source code steps forward until it finds valid data at this address.
[3]	**Reserved** This bit is reserved and must be programmed to 0.
[2] FE	**Flash Error** If a Flash error is detected, this bit is set to 1.
[1] IGADDR	**Illegal Address** When NVDS byte reads from invalid addresses (those exceeding the NVDS array size) occur, this bit is set to 1.
[0]	**Reserved** This bit is reserved and must be programmed to 0.

Operation

The following section describes the operation of the On-Chip Debugging function.

OCD Interface

The On-Chip Debugger uses the DBG pin for communication with an external host. This one-pin interface is a bidirectional open-drain interface that transmits and receives data. Data transmission is half-duplex, which means that transmission and data retrieval cannot occur simultaneously. The serial data on the DBG pin is sent using the standard asynchronous data format defined in RS-232.This pin creates an interface between the Z8 Encore! F0830 Series products and the serial port of a host PC using minimal external hardware. Two different methods for connecting the DBG pin to an RS-232 interface are displayed in Figures 21 and 22. The recommended method is the buffered implementation depicted in Figure 22. The DBG pin must always be connected to V_{DD} through an external pull-up resistor.

! Caution: For proper operation of the On-Chip Debugger, all power pins (V_{DD} and AV_{DD}) must be supplied with power and all ground pins (V_{SS} and AV_{SS}) must be properly grounded. The DBG pin is open-drain and must always be connected to V_{DD} through an external pull-up resistor to ensure proper operation.

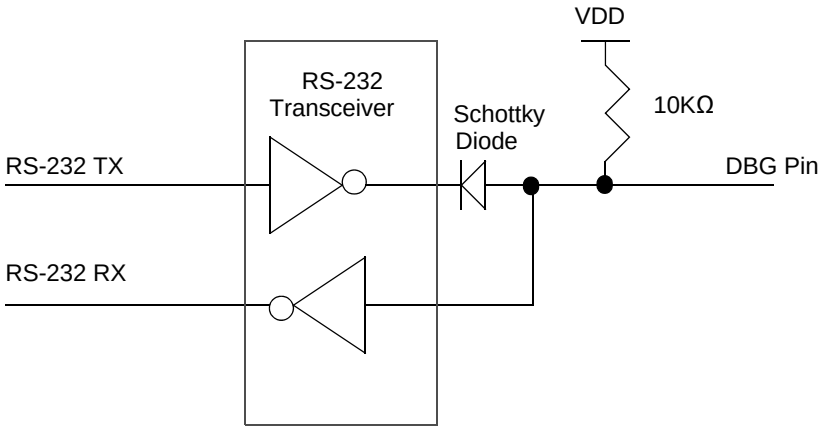

Figure 21. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #1 of 2

Runtime Counter

The OCD contains a 16-bit runtime counter. It counts system clock cycles between breakpoints. The counter starts counting when the OCD leaves DEBUG Mode and stops counting when it enters DEBUG Mode again or when it reaches the maximum count of FFFFH.

On-Chip Debugger Commands

The host communicates to the On-Chip Debugger by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG Mode, all OCD commands become available unless the user code and control registers are protected by programming the Flash read protect option bit (FRP). The FRP prevents the code in memory from being read out of the Z8 Encore! F0830 Series products. When this option is enabled, several of the OCD commands are disabled.

Table 95 summarizes the On-Chip Debugger commands. This table indicates the commands that operate when the device is not in DEBUG Mode (normal operation) and the commands that are disabled by programming the FRP.

Table 95. On-Chip Debugger Command Summary

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

Read OCD Control Register (05H). The read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

Write Program Counter (06H). The write program counter command, writes the data that follows to the eZ8 CPU's program counter (PC). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, the program counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

Read Program Counter (07H). The read program counter command, reads the value in the eZ8 CPUs program counter (PC). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

Write Register (08H). The write register command, writes data to the register file. Data can be written 1–256 bytes at a time (256 bytes can be written by setting size to 0). If the device is not in DEBUG Mode, the address and data values are discarded. If the Flash read protect option bit is enabled, only writes to the Flash control registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0,Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1–256 data bytes
```

Read Register (09H). The read register command, reads data from the register file. Data can be read 1–256 bytes at a time (256 bytes can be read by setting size to 0). If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, this command returns FFH for all of the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG → 1–256 data bytes
```

Write Program Memory (0AH). The write program memory command, writes data to program memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG Mode or if the Flash read protect option bit is enabled, the data is discarded.

Oscillator Control

The Z8 Encore! F0830 Series device uses five possible clocking schemes. Each one of these is user-selectable.

- On-chip precision trimmed RC oscillator
- On-chip oscillator using off-chip crystal or resonator
- On-chip oscillator using external RC network
- External clock drive
- On-chip low precision Watchdog Timer Oscillator

In addition, Z8 Encore! F0830 Series devices contain clock failure detection and recovery circuitry, allowing continued operation despite a failure of the primary oscillator.

Operation

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

System Clock Selection

The oscillator control block selects from the available clocks. *Table 98* describes each clock source and its usage.

eZ8 CPU Instruction Summary

Table 113 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 113. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags 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

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

Table 121. Nonvolatile Data Storage

	V_{DD} = 2.7 to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7 to 3.6V T_A = –40°C to +105°C				
Parameter	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Notes**
NVDS Byte Read Time				71	–	258	µs	With system clock at 20 MHz
NVDS Byte Program Time				126	–	136	µs	With system clock at 20 MHz
Data Retention				10	–	–	years	25°C
Endurance				100,000	–	–	cycles	Cumulative write cycles for entire memory

Note: For every 200 writes, a maintenance operation is necessary. In this rare occurrence, the write can take up to 58 ms to complete.

Table 122. Analog-to-Digital Converter Electrical Characteristics and Timing

		V_{DD} = 2.7 to 3.6V T_A = 0°C to +70°C			V_{DD} = 2.7 to 3.6V T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
	Resolution				–	10	–	bits	
	Differential Nonlinearity (DNL) [1]				–1	–	+4	LSB	
	Integral Nonlinearity (INL) [1]				–5	–	+5	LSB	
	Gain Error					15		LSB	
	Offset Error				–15	–	15	LSB	PDIP package
					–9	–	9	LSB	Other packages
V_{REF}	On chip reference				1.9	2.0	2.1	V	
	Active Power Consumption					4		mA	
	Power Down Current						1	µA	

Note: [1]When the input voltage is lower than 20 mV, the conversion error is out of spec.

General Purpose I/O Port Input Data Sample Timing

Figure 33 displays timing of the GPIO port input sampling. The input value on a GPIO port pin is sampled on the rising edge of the system clock. The port value is available to the eZ8 CPU on the second rising clock edge following the change of the port value.

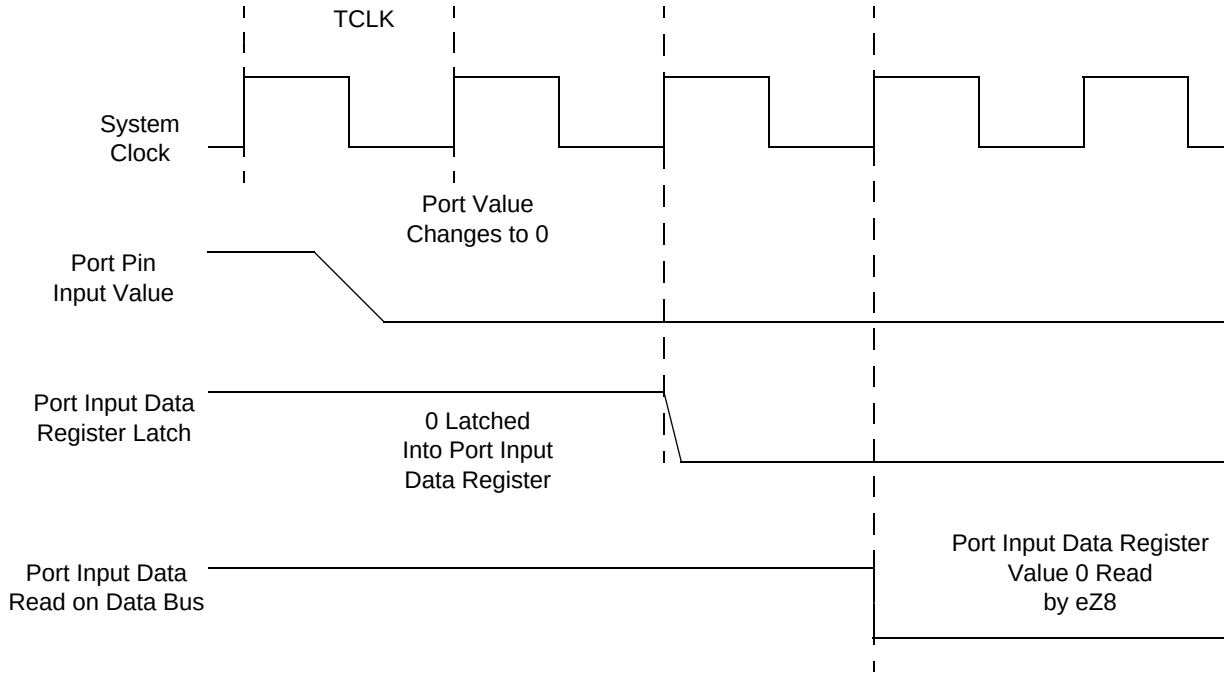

Figure 33. Port Input Sample Timing

Table 124. GPIO Port Input Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_{S_PORT}	Port Input Transition to X_{IN} Rise Setup Time (not pictured)	5	–
T_{H_PORT}	X_{IN} 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)	1µs	

Table 128. Z8 Encore! XP F0830 Series Ordering Matrix

Part Number	Flash	RAM	NVDS	ADC Channels	Description
Extended Temperature: –40°C to 105°C					
Z8F0230SH020EG	2KB	256	Yes	7	SOIC 20-pin
Z8F0230HH020EG	2KB	256	Yes	7	SSOP 20-pin
Z8F0230PH020EG	2KB	256	Yes	7	PDIP 20-pin
Z8F0230QH020EG	2KB	256	Yes	7	QFN 20-pin
Z8F0231SH020EG	2KB	256	Yes	0	SOIC 20-pin
Z8F0231HH020EG	2KB	256	Yes	0	SSOP 20-pin
Z8F0231PH020EG	2KB	256	Yes	0	PDIP 20-pin
Z8F0231QH020EG	2KB	256	Yes	0	QFN 20-pin
Z8F0230SJ020EG	2KB	256	Yes	8	SOIC 28-pin
Z8F0230HJ020EG	2KB	256	Yes	8	SSOP 28-pin
Z8F0230PJ020EG	2KB	256	Yes	8	PDIP 28-pin
Z8F0230QJ020EG	2KB	256	Yes	8	QFN 28-pin
Z8F0231SJ020EG	2KB	256	Yes	0	SOIC 28-pin
Z8F0231HJ020EG	2KB	256	Yes	0	SSOP 28-pin
Z8F0231PJ020EG	2KB	256	Yes	0	PDIP 28-pin
Z8F0231QJ020EG	2KB	256	Yes	0	QFN 28-pin
Z8 Encore! F0830 with 1KB Flash					
Standard Temperature: 0°C to 70°C					
Z8F0130SH020SG	1KB	256	Yes	7	SOIC 20-pin
Z8F0130HH020SG	1KB	256	Yes	7	SSOP 20-pin
Z8F0130PH020SG	1KB	256	Yes	7	PDIP 20-pin
Z8F0130QH020SG	1KB	256	Yes	7	QFN 20-pin
Z8F0131SH020SG	1KB	256	Yes	0	SOIC 20-pin
Z8F0131HH020SG	1KB	256	Yes	0	SSOP 20-pin
Z8F0131PH020SG	1KB	256	Yes	0	PDIP 20-pin
Z8F0131QH020SG	1KB	256	Yes	0	QFN 20-pin
Z8F0130SJ020SG	1KB	256	Yes	8	SOIC 28-pin
Z8F0130HJ020SG	1KB	256	Yes	8	SSOP 28-pin
Z8F0130PJ020SG	1KB	256	Yes	8	PDIP 28-pin
Z8F0130QJ020SG	1KB	256	Yes	8	QFN 28-pin
Z8F0131SJ020SG	1KB	256	Yes	0	SOIC 28-pin
Z8F0131HJ020SG	1KB	256	Yes	0	SSOP 28-pin