



Welcome to [E-XFL.COM](https://www.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	4KB (4K 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-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f043ahh020sg

Assembly Language Programming Introduction	162
Assembly Language Syntax	163
eZ8 CPU Instruction Notation	164
eZ8 CPU Instruction Classes	166
eZ8 CPU Instruction Summary	171
Op Code Maps	181
Electrical Characteristics	184
Absolute Maximum Ratings	184
DC Characteristics	185
AC Characteristics	188
On-Chip Peripheral AC and DC Electrical Characteristics	189
General Purpose I/O Port Input Data Sample Timing	194
GPIO Port Output Timing	195
On-Chip Debugger Timing	196
Packaging	198
Ordering Information	199
Part Number Suffix Designations	201
Appendix A. Register Tables	203
General Purpose RAM	203
Timer 0	203
Analog-to-Digital Converter (ADC)	208
Low Power Control	212
LED Controller	212
Oscillator Control	213
Comparator 0	214
Interrupt Controller	214
GPIO Port A	218
Watchdog Timer (WDT)	222
Trim Bit Control	224
Flash Memory Controller	224
Index	227
Customer Support	235

Nonvolatile Data Storage

The nonvolatile data storage (NVDS) uses a hybrid hardware/software scheme to implement a byte programmable data memory and is capable of storing about 100,000 write cycles.

Internal Precision Oscillator

The internal precision oscillator (IPO) with accuracy of $\pm 4\%$ full voltage/temperature range is a trimable clock source that requires no external components.

External Crystal Oscillator

The external crystal oscillator circuit provides highly accurate clock frequencies with the use of an external crystal, ceramic resonator or RC network.

10-Bit Analog-to-Digital Converter

The analog-to-digital converter (ADC) converts an analog input signal to a 10-bit binary number. The ADC accepts inputs from eight different analog input pins. It has a fast 2.8 μ s conversion speed.

Analog Comparator

The analog comparator compares the signal at an input pin with either an internal programmable reference voltage or with a signal at the second input pin. The comparator output is used either to drive a logic output pin or to generate an interrupt.

Timers

Two enhanced 16-bit reloadable timers are used for timing/counting events or for motor control operations. These timers provide a 16-bit programmable reload counter and operate in ONE-SHOT, CONTINUOUS, GATED, CAPTURE, CAPTURE RESTART, COMPARE, CAPTURE and COMPARE, PWM SINGLE OUTPUT and PWM DUAL OUTPUT modes.

Interrupt Controller

The Z8 Encore! F083A Series products support seventeen interrupt sources with sixteen interrupt vectors: up to five internal peripheral interrupts and up to twelve GPIO interrupts. These interrupts have three levels of programmable interrupt priority.

Address Space

The eZ8 CPU accesses three distinct address spaces as follows:

- The Register File addresses access for the general purpose registers and the eZ8 CPU, peripheral and GPIO port control registers
- Program Memory addresses access for all of the memory locations having executable code and/or data
- The Data Memory addresses access for all of the memory locations containing only the data

The following sections describe these three address spaces. For more detailed information about the eZ8 CPU and its address space, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#), available for download on www.zilog.com.

Register File

The Register File address space in the Z8 Encore! MCU is 4KB (4096 bytes). The Register File consists of two sections: control registers and general purpose registers. When instructions are executed, registers defined as source are read and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general purpose registers to function as accumulators, address pointers, index registers, stack areas or scratch pad memory.

The upper 256 bytes of the 4KB Register File address space are reserved for control of the eZ8 CPU, on-chip peripherals and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256 B Control Register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and produces unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The Z8 Encore! F083A Series devices contain up to 256 B of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the Control Register address space), returns an undefined value. Writing to these Register File addresses has no effect.

Table 7. Z8 Encore! F083A 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.

address. Following Stop Mode Recovery, the STOP bit in the Reset Status (RSTSTAT) Register is set to 1. Table 11 lists the Stop Mode Recovery sources and resulting actions. The following sections provide more detailed information about each of the Stop Mode Recovery sources.

Table 11. Stop Mode Recovery Sources and Resulting Action

Operating Mode	Stop Mode Recovery Source	Action
STOP Mode	WDT time-out when configured for Reset.	Stop Mode Recovery.
	WDT time-out when configured for interrupt.	Stop Mode Recovery followed by interrupt (if interrupts are enabled).
	Data transition on any GPIO port pin enabled as a Stop Mode Recovery source.	Stop Mode Recovery.
	Assertion of external RESET pin.	System reset.
	Debug pin driven Low.	System reset.

Stop Mode Recovery using Watchdog Timer Time-Out

If the WDT times out during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Reset Status (RSTSTAT) Register, the WDT and STOP bits are set to 1. If the WDT is configured to generate an interrupt upon time-out and the Z8 Encore! F083A Series device is configured to respond to interrupts, the eZ8 CPU services the WDT interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery using GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. If any GPIO pin is enabled as a Stop Mode Recovery source, a change in the input pin value (from high to low or from low to high) initiates Stop Mode Recovery. In the Reset Status (RSTSTAT) Register, the STOP bit is set to 1.

! Caution: In STOP Mode, the GPIO port input data registers (PxIN) are disabled. The port input data registers record the port transition only if the signal stays on the port pin through the end of the Stop Mode Recovery delay. As a result, short pulses on the port pin initiates Stop Mode Recovery without being written to the port Input Data Register or without initiating an interrupt (if enabled for that pin).

Power Control Register 0

Each bit in this register disables a peripheral block, either by gating its system clock input or by removing power from the block.

► **Note:** This register is only reset during a POR sequence. Other system Reset events do not affect it.

Table 14. Power Control Register 0 (PWRCTL0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			VBO	Reserved	Reserved	COMP	Reserved
RESET	1	0	0	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F80H							

Bit	Description
[7:5]	Reserved These bits are reserved and must be programmed to 000.
[4] VBO	Voltage Brown-Out detector Disable This bit is only effect when the VBO_AO Flash option bit is disabled. In STOP Mode, VBO is always disabled when the VBO_AO Flash option bit is disabled. For VBO_AO Flash option bit function, see the Flash Option Bits chapter on page 124. 0 = VBO enabled. 1 = VBO disabled.
[3]	Reserved This bit is reserved and must be programmed to 0.
[2]	Reserved This bit is reserved and must be programmed to 0.
[1] COMP	Comparator Disable 0 = Comparator is enabled. 1 = Comparator is disabled.
[0]	Reserved This bit is reserved and must be programmed to 0.

```
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 generates 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.

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. This allows user software to determine if interrupts are generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt were 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. The user must 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 between the timer start and the capture event is 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). Additionally, 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 resets to 0000H and continues 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
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.

5. Configure the associated GPIO port pin for the timer input alternate function.
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 is 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 are read (i.e., enabled) while counting. 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 is 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, there is no timer input available.

Timer Control Register Definitions

This section defines the features of the following Timer Control registers.

Timer 0–1 High and Low Byte Registers: see page 84

Timer Reload High and Low Byte Registers: see page 84

Timer 0–1 PWM High and Low Byte Registers: see page 86

Timer 0–1 Control Registers: see page 87

Bit	Description (Continued)
[6] TPOL	<p>Timer Input/Output Polarity Operation of this bit is a function of the current operating mode of the timer.</p> <p>ONE-SHOT 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 on timer reload.</p> <p>CONTINUOUS Mode When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled and reloaded, the timer output signal is complemented.</p> <p>COUNTER Mode If the timer is disabled, the timer output signal is set to the value of this bit. If the timer is enabled the timer output signal is complemented after timer reload. 0 = Count occurs on the rising edge of the timer input signal. 1 = Count occurs on the falling edge of the timer input signal.</p> <p>PWM SINGLE OUTPUT Mode 0 = Timer output is forced low (0), when the timer is disabled. The timer output is forced high (1), when the timer is enabled and the PWM count matches and the timer output is forced low (0), when the timer is enabled and reloaded. 1 = Timer output is forced high (1), when the timer is disabled. The timer output is forced low(0), when the timer is enabled and the PWM count matches and forced high (1) when the timer is enabled and reloaded.</p> <p>CAPTURE 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.</p> <p>COMPARE Mode When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled and reloaded, the timer output signal is complemented.</p> <p>GATED Mode 0 = Timer counts when the timer input signal is high (1) and interrupts are generated on the falling edge of the timer input. 1 = Timer counts when the timer input signal is low (0) and interrupts are generated on the rising edge of the timer input.</p> <p>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.</p>

Flash Memory

The products in the Z8 Encore! F083A Series features either 4 KB (4096 bytes with NVDS) or 8 KB (8192 bytes with NVDS) of nonvolatile Flash memory with read/write/erase capability. The Flash memory is 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 is 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! F083A Series, each sector maps to one page for 4KB devices and two pages for 8KB devices.

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

Table 70 describes the Flash memory configuration for each device in the Z8 Encore! F083A Series. Figures 14 and 15 display the Flash memory arrangement.

Table 70. Z8 Encore! F083A Series Flash Memory Configurations

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash Sector Size (bytes)
Z8F083A	8 (8196)	16	0000H–1FFFH	1024
Z8F043A	4 (4096)	8	0000H–0FFFH	512

Table 72. Flash Code Protection Using Flash Option Bits

FHSWP	FWP	Flash Code Protection Description
0	0	Programming and erasing disabled for all Flash program memory. In user code programming, page erase and mass erase are all disabled. Mass erase is available through the On-Chip Debugger.
0 or 1	1	Programming, page erase and mass erase are enabled for all of the Flash program memory.

At reset, the Flash Controller is locked to prevent accidental program or erasure of Flash memory. To program or erase Flash memory, first write the target page to the Page Select Register. 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 details, see [Figure 16](#) on page 114.

After unlocking a specific page, the user must 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 is allowed through the debug port.

After unlocking a specific page, the user must 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 causes 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! devices are divided into maximum number of eight sectors. A sector is one-eighth of the total size of Flash memory, unless this value is smaller than the page size, in which case the sector and page sizes are equal. On the Z8 Encore! F083A Series devices, the sector size is varied; see [Table 70](#) and Figures 14 and 15.

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 is lost. User code must write this register in their initialization routine if they want 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, bits can only be set to 1. Thus, sectors can be protected, but not unprotected, via register

write operations. Writing a value other than 5EH to the Flash Control Register deselects the Flash Sector Protect Register and reenables access to the Page Select Register. Observe the following procedure to setup the Flash Sector Protect Register from user code:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write 5EH to the Flash Control Register to select the Flash Sector Protect Register.
3. Read and/or write the Flash Sector Protect Register which is now at Register File address FF9H.
4. Write 00H to the Flash Control Register to return the Flash Controller to its reset state.

The Sector Protect Register is initialized to 0 upon reset, putting each sector into an unprotected state. When a bit in the Sector Protect Register is written to 1, the corresponding sector is no longer be written or erased. After setting a bit in the Sector Protect Register, the bit cannot be cleared by the user.

Byte Programming

The Flash memory is enabled for byte programming after unlocking the Flash Controller and successfully enabling either mass erase or page erase. When the Flash Controller is unlocked and mass erase is successfully enabled, all of the program memory locations are available for byte programming. In contrast, when the Flash Controller is unlocked and page erase is successfully enabled, only the locations of the selected page are available for byte programming. An erased Flash byte contains all 1's (FFH). The programming operation is used only to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires execution of either the page erase or mass erase commands.

Byte programming is accomplished using the On-Chip Debugger's write memory command or eZ8 CPU execution of the LDC or LDCI instructions. For the description of the LDC and LDCI instructions, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#), available for download on www.zilog.com. While the Flash Controller programs Flash memory, the eZ8 CPU idles, but the system clock and on-chip peripherals continue to operate. To exit programming mode and lock the Flash, write any value to the Flash Control Register, except the mass erase or page erase commands.

! **Caution:** The byte at each Flash memory address cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs.

Table 81. Trim Bit Data Register (TRMDR)

Bit	7	6	5	4	3	2	1	0
Field	TRMDR: Trim Bit Data							
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	FF7H							

Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses 0000H and 0001H are reserved for the user-programmable Flash option bits.

Flash Program Memory Address 0000H

Table 82. Flash Option Bits at Program Memory Address 0000H

Bit	7	6	5	4	3	2	1	0
Field	WDT_RES	WDT_AO	OSC_SEL[1:0]		VBO_AO	FRP	Reserved	FWP
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
Address	Program Memory 0000H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description
[7] WDT_RES	Watchdog Timer Reset 0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request. 1 = Watchdog Timer time-out causes a system reset. This is the default setting for unprogrammed (erased) Flash.
[6] WDT_AO	Watchdog Timer Always On 0 = On application of system power, Watchdog Timer is automatically enabled. Watchdog Timer cannot be disabled. 1 = Watchdog Timer is enabled on execution of the WDT instruction. Once enabled, the Watchdog Timer is disabled only by a reset. This is the default setting for unprogrammed (erased) Flash.

Bit	Description
[7] VBO_RES	Voltage Brown-Out Reset 1 = VBO detection causes a system reset. This setting is the default setting for unprogrammed (erased) Flash.
[6:5]	Reserved These bits are reserved and must be programmed to 11.
[4] XTLDIS	State of the Crystal Oscillator at Reset This bit only enables the crystal oscillator. Its selection as a system clock must be performed manually. 0 = The crystal oscillator is enabled during reset, resulting in longer reset timing. 1 = The crystal oscillator is disabled during reset, resulting in shorter reset timing.
[3:0]	Reserved These bits are reserved and must be programmed to 1111.

Trim Bit Address Space

Table 84. Trim Bit Address Space

Address	Function
00h	ADC reference voltage
01h	ADC and comparator
02h	Internal precision oscillator
03h	Oscillator and VBO
06h	ClkFiltr

Trim Bit Address 0000H

Table 85. Trim Option Bits at 0000H (ADCREF)

Bit	7	6	5	4	3	2	1	0
Field	ADCREF_TRIM					Reserved		
RESET	U					U		
R/W	R/W					R/W		
Address	Information Page Memory 0020H							
Note: U = Unchanged by Reset; R/W = Read/Write.								

Bit	Description (Continued)
[1] IGADDR	Illegal Address When NVDS byte reads from invalid addresses occur (those exceeding the NVDS array size), this bit is set to 1.
[0]	Reserved This bit is reserved and must be programmed to 0.

Power Failure Protection

The NVDS routines employ error-checking mechanisms to ensure that any power failure will only endanger the most recently written byte. Bytes previously written to the array are not perturbed. For this protection to function, the VBO must be enabled (See *the Low-Power Modes* chapter on page 30) and configured for a threshold voltage of 2.4V or greater (See *the Trim Bit Address Space* section on page 129).

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

Optimizing NVDS Memory Usage for Execution Speed

As Table 94 shows, NVDS read times vary drastically, this discrepancy being a trade-off for minimizing the frequency of writes that require post-write page erases. The NVDS read time of address N is a function of the number of writes to addresses other than N since the most recent write to address N, as well as the number of writes since the most recent page erase. Neglecting effects caused by page erases and results caused by the initial condition in which the NVDS is blank, a rule of thumb is that every write since the most recent page erase causes read times of unwritten addresses to increase by 0.8 μ s, up to a maximum of 258 μ s.

Table 94. NVDS Read Time

Operation	Minimum Latency (μ s)	Maximum Latency (μ s)
Read	71	258
Write	126	136
Illegal Read	6	6
Illegal Write	7	7

```
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 also steps 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

This section describes the features of the On-Chip Debugger Control and Status registers.

OCD Control Register

The OCD Control Register controls the state of the On-Chip Debugger. This register is used to enter or exit DEBUG Mode and to enable the BRK instruction. It also resets the Z8 Encore! F083A Series device.

A reset and stop function is achieved by writing 81H to this register. A reset and Go function is achieved by writing 41H to this register. If the device is in DEBUG Mode, a run function is implemented by writing 40H to this register.

Table 99. Oscillator Configuration and Selection

Clock Source	Characteristics	Required Setup
Internal precision RC oscillator	<ul style="list-style-type: none"> • 119kHz or 20MHz • $\pm 4\%$ accuracy when trimmed • No external components required 	<ul style="list-style-type: none"> • Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator at either 20MHz or 119kHz
External crystal/resonator	<ul style="list-style-type: none"> • 32kHz to 20MHz • Very high accuracy (dependent on crystal or resonator used) • Requires external components 	<ul style="list-style-type: none"> • Configure Flash option bits for correct external OSCILLATOR mode • Unlock and write OSCCTL to enable crystal oscillator, wait for it to stabilize and select as system clock (if the XTLDIS option bit has been de-asserted, no waiting is required)
External RC oscillator	<ul style="list-style-type: none"> • 32kHz to 4MHz • Accuracy dependent on external components 	<ul style="list-style-type: none"> • Configure Flash option bits for correct external OSCILLATOR Mode • Unlock and write OSCCTL to enable crystal oscillator and select as system clock
External clock drive	<ul style="list-style-type: none"> • 0 to 20MHz • Accuracy dependent on external clock source 	<ul style="list-style-type: none"> • Write GPIO registers to configure PB3 pin for external clock function • Unlock and write OSCCTL to select external system clock • Apply external clock signal to GPIO
Internal Watchdog Timer Oscillator	<ul style="list-style-type: none"> • 10kHz nominal • $\pm 40\%$ accuracy; no external components required • Low power consumption 	<ul style="list-style-type: none"> • Enable WDT if not enabled and wait until WDT oscillator is operating. • Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator

! Caution: Unintentional accesses to the Oscillator Control Register actually stop the chip by switching to a nonfunctioning 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.

Table 114. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
AND dst, src	$\text{dst} \leftarrow \text{dst AND src}$	r	r	52	–	*	*	0	–	–	2	3
		r	lr	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	$\text{dst} \leftarrow \text{dst AND src}$	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	$\text{dst}[\text{bit}] \leftarrow 0$	r		E2	–	*	*	0	–	–	2	2
BIT p, bit, dst	$\text{dst}[\text{bit}] \leftarrow p$	r		E2	–	*	*	0	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	$\text{dst}[\text{bit}] \leftarrow 1$	r		E2	–	*	*	0	–	–	2	2
BSWAP dst	$\text{dst}[7:0] \leftarrow \text{dst}[0:7]$	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if $\text{src}[\text{bit}] = p$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJNZ bit, src, dst	if $\text{src}[\text{bit}] = 1$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJZ bit, src, dst	if $\text{src}[\text{bit}] = 0$ $\text{PC} \leftarrow \text{PC} + X$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4

Note: Flags Notation:

* = Value is a function of the result of the operation.

– = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

- step instruction (10H) 148
- stuff instruction (11H) 148
- write data memory (0CH) 147
- write OCD control register (04H) 145
- write program counter (06H) 146
- write program memory (0AH) 146
- write register (08H) 146
- on-chip debugger (OCD) 139
- on-chip debugger signals 12
- on-chip oscillator 157
- one-shot mode 89
- opcode map
 - abbreviations 181
 - cell description 181
 - first 182
 - second after 1FH 183
- operation 100
 - current measurement 99
 - voltage measurement timing diagram 100
- Operational Description 21, 30, 33, 54, 69, 92, 98, 108, 110, 124, 139, 151, 157, 161
- OR 169
- ordering information 199
- ORX 169
- oscillator signals 11

P

- p 164
- Packaging 198
- part selection guide 2
- PC 165
- peripheral AC and DC electrical characteristics 189
- pin characteristics 13
- Pin Descriptions 7
- polarity 164
- POP 168
- pop using extended addressing 168
- POPX 168
- port availability, device 33
- port input timing (GPIO) 194
- port output timing, GPIO 195
- power supply signals 12
- power-on reset (POR) 23

- program control instructions 169
- program counter 165
- program memory 15
- PUSH 168
- push using extended addressing 168
- PUSHX 168
- PWM mode 89, 90
- PxADDR register 40, 218, 219, 220, 221
- PxCTL register 41, 218, 219, 220, 221

R

- R 164
- r 164
- RA
 - register address 165
- RCF 167, 168
- register 164
 - flash control (FCTL) 120, 126, 127, 224
 - flash high and low byte (FFREQH and FRE-EQL) 123
 - flash page select (FPS) 121, 122
 - flash status (FSTAT) 121
 - GPIO port A-H address (PxADDR) 40, 218, 219, 220, 221
 - GPIO port A-H alternate function sub-registers 42
 - GPIO port A-H control address (PxCTL) 41, 218, 219, 220, 221
 - GPIO port A-H data direction sub-registers 41
 - OCD control 148
 - OCD status 150
 - watch-dog timer control (WDTCTL) 29, 95, 109, 154, 213, 214, 222
 - watch-dog timer reload high byte (WDTH) 96, 223
 - watch-dog timer reload low byte (WDTL) 97, 223
 - watch-dog timer reload upper byte (WDTU) 96, 223
- register file 14
- register pair 165
- register pointer 165
- registers