

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	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 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VQFN Exposed Pad
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f0130qh020eg">https://www.e-xfl.com/product-detail/zilog/z8f0130qh020eg</a>

# **Table of Contents**

Revision History .....	iii
List of Figures .....	x
List of Tables .....	xii
Overview .....	1
Features .....	1
Part Selection Guide .....	2
Block Diagram .....	3
CPU and Peripheral Overview .....	4
General Purpose Input/Output .....	4
Flash Controller .....	4
Nonvolatile Data Storage .....	5
Internal Precision Oscillator .....	5
External Crystal Oscillator .....	5
10-Bit Analog-to-Digital Converter .....	5
Analog Comparator .....	5
Timers .....	5
Interrupt Controller .....	5
Reset Controller .....	6
On-Chip Debugger .....	6
Acronyms and Expansions .....	6
Pin Description .....	7
Available Packages .....	7
Pin Configurations .....	7
Signal Descriptions .....	11
Pin Characteristics .....	13
Address Space .....	14
Register File .....	14
Program Memory .....	15
Data Memory .....	16
Flash Information Area .....	16
Register Map .....	17
Reset and Stop Mode Recovery .....	21
Reset Types .....	21
Reset Sources .....	23
Power-On Reset .....	23

# Address Space

The eZ8 CPU can access the following three distinct address spaces:

- The register file addresses access for the general purpose registers and the eZ8 CPU, peripheral and general purpose I/O port control registers
- The 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 information about the eZ8 CPU and its address space, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download at [www.zilog.com](http://www.zilog.com).

## Register File

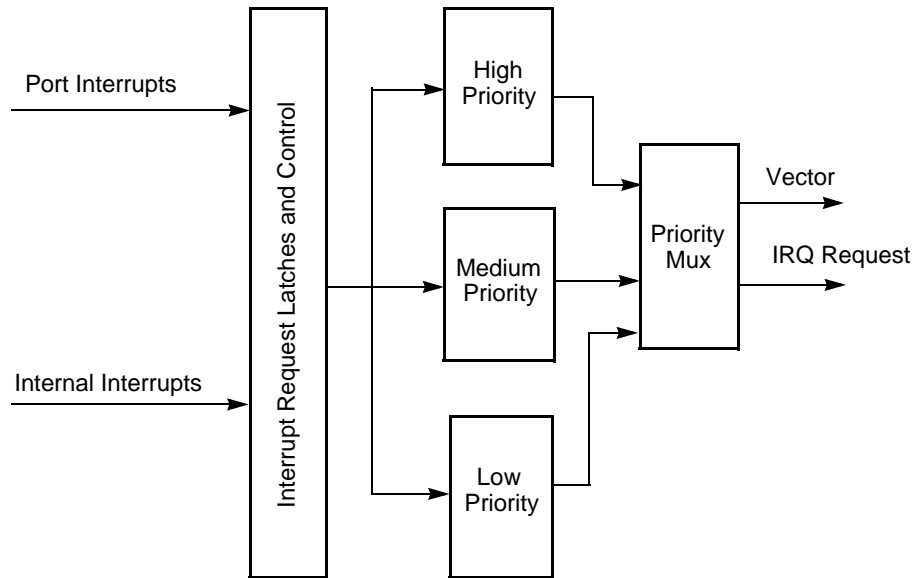
The register file address space in the Z8 Encore! MCU is 4KB (4096 bytes). The register file consists of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as *source* are read and registers defined as *destinations* are written. The architecture 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 controlling 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 256B 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 can produce unpredictable results.

The on-chip RAM always begins at address 000H in the register file address space. The Z8 Encore! F0830 Series devices contain up to 256B 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.

## Architecture

Figure 9 displays the Interrupt Controller block diagram.



**Figure 9. Interrupt Controller Block Diagram**

## Operation

This section describes the operational aspects of the following functions.

Master Interrupt Enable: see page 55

Interrupt Vectors and Priority: see page 56

Interrupt Assertion: see page 56

Software Interrupt Assertion: see page 57

### Master Interrupt Enable

The master interrupt enable bit (IRQE) in the Interrupt Control Register globally enables and disables the interrupts.

Interrupts are globally enabled by any of the following actions:

- Execution of an EI (enable interrupt) instruction
- Execution of an IRET (return from interrupt) instruction

---

**! 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:

$$\text{Comparator Output Transitions} = \text{Current Count Value} - \text{Start Value}$$

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 the 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 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 whether 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 the TxCTL1 Register is set to indicate that the timer interrupt has been caused by an input capture event.

If no capture event occurs, the timer counts up to 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 the TxCTL1 Register is cleared to indicate that the timer interrupt has not been caused by an input capture event.

Observe the following steps for configuring a timer for CAPTURE RESTART Mode and for 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 the 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).

## Watchdog Timer Refresh

Upon first enable, the Watchdog Timer is loaded with the value in the Watchdog Timer Reload registers. The Watchdog Timer counts down to 000000H unless a WDT instruction is executed by the eZ8 CPU. Execution of the WDT instruction causes the downcounter to be reloaded with the WDT reload value stored in the Watchdog Timer Reload registers. Counting resumes following the Reload operation.

When the Z8 Encore! F0830 Series devices are operating in DEBUG Mode (using the On-Chip Debugger), the Watchdog Timer must be continuously refreshed to prevent any WDT time-outs.

## Watchdog Timer Time-Out Response

The Watchdog Timer times out when the counter reaches 000000H. A time-out of the Watchdog Timer generates either an interrupt or a system reset. The WDT\_RES Flash option bit determines the time-out response of the Watchdog Timer. See *the Flash Option Bits* chapter on page 124 for information about programming the WDT\_RES Flash option bit.

### WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the Watchdog Timer issues an interrupt request to the Interrupt Controller and sets the WDT status bit in the Reset Status Register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executing code from the vector address. After time-out and interrupt generation, the Watchdog Timer counter resets to its maximum value of FFFFFFFH and continues counting. The Watchdog Timer counter will not automatically return to its reload value.

The Reset Status Register (see *Table 12 on page 29*) must be read before clearing the WDT interrupt. This read clears the WDT time-out flag and prevents further WDT interrupts occurring immediately.

### WDT Interrupt in STOP Mode

If configured to generate an interrupt when a time-out occurs and the Z8 Encore! F0830 Series devices are in STOP Mode, the Watchdog Timer automatically initiates a Stop Mode Recovery and generates an interrupt request. Both the WDT status bit and the STOP bit in the Watchdog Timer Control Register are set to 1 following a WDT time-out in STOP Mode. See *the Reset and Stop Mode Recovery* chapter on page 21 for more information about Stop Mode Recovery operations.

If interrupts are enabled, following completion of the Stop Mode Recovery, the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executes the code from the vector address.

## ADC Data High Byte Register

The ADC Data High Byte Register, listed in Table 64, contains the upper eight bits of the ADC output. Access to the ADC Data High Byte Register is read-only. Reading the ADC Data High Byte Register latches data in the ADC Low Bits Register.

**Table 64. ADC Data High Byte Register (ADCD\_H)**

Bit	7	6	5	4	3	2	1	0
Field	ADCDH							
RESET	X							
R/W	R							
Address	F72H							

Bit	Description
[7:0] ADCDH	<b>ADC High Byte</b> 00h–FFh = The last conversion output is held in the data registers until the next ADC conversion is completed.

## ADC Data Low Bits Register

The ADC Data Low Bits Register, shown in Table 65, contains the lower bits of the ADC output. Access to the ADC Data Low Bits Register is read-only. Reading the ADC Data High Byte Register latches lower bits of the ADC in the ADC Data Low Bits Register.

**Table 65. ADC Data Low Bits Register (ADCD\_L)**

Bit	7	6	5	4	3	2	1	0
Field	ADCDL		Reserved					
RESET	X		X					
R/W	R		R					
Address	F73H							

Bit	Description
[7:6] ADCDL	<b>ADC Low Bits</b> 00–11b = These bits are the two least-significant bits of the 10-bit ADC output. These bits are undefined after a reset. The low bits are latched into this register whenever the ADC Data High Byte Register is read.
[5:0]	<b>Reserved</b> These bits are reserved and must be programmed to 000000.

## Sample Settling Time Register

The Sample Settling Time Register, shown in Table 66, is used to program a delay after the  $\overline{\text{SAMPLE/HOLD}}$  signal is asserted and before the START signal is asserted; an ADC conversion then begins. The number of clock cycles required for settling will vary from system to system depending on the system clock period used. The system designer should program this register to contain the number of clocks required to meet a 0.5  $\mu\text{s}$  minimum settling time.

**Table 66. Sample Settling Time (ADCSST)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved				SST			
RESET	0				1	1	1	1
R/W	R				R/W			
Address	F74H							

Bit	Description
[7:4]	<b>Reserved</b> These bits are reserved and must be programmed to 0000.
[3:0] SST	0h–Fh = Sample settling time in number of system clock periods to meet 0.5 $\mu\text{s}$ minimum.

# Flash Memory

The products in the Z8 Encore! F0830 Series features either 1 KB (1024 bytes with NVDS), 2 KB (2048 bytes with NVDS), 4 KB (4096 bytes with NVDS), 8 KB (8192 bytes with NVDS) or 12 KB (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.

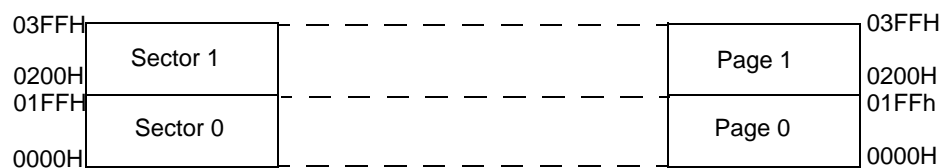
For program/data protection, Flash memory is also divided into sectors. In the Z8 Encore! F0830 Series, each sector maps to one page (for 1 KB, 2 KB and 4 KB devices), two pages (8 KB device) or three pages (12 KB device).

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**

**Table 71. Flash Code Protection using the Flash Option Bits**

<b>FHSWP</b>	<b>FWP</b>	<b>Flash Code Protection Description</b>
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. See Figure 19 for details.

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 is allowed through the debug port.

After unlocking a specific page, the user 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 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 Z8 Encore! F0830 Series devices, the sector size is varied according to the Z8 Encore! F0830 Series Flash Memory Configuration shown in [Table 69](#) on page 108 and in Figures 14 through 18, which follow the table

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,

## Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte registers, shown in Tables 76 and 77, combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation:

$$\text{FFREQ}[15:0] = \{\text{FFREQH}[7:0], \text{FFREQL}[7:0]\} = \frac{\text{System Clock Frequency}}{1000}$$

**!** **Caution:** Flash programming and erasure is not supported for system clock frequencies below 10kHz or above 20MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device.

**Table 76. Flash Frequency High Byte Register (FFREQH)**

Bit	7	6	5	4	3	2	1	0
Field	FFREQH							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FFAH							

Bit	Description
[7:0]	<b>Flash Frequency High Byte</b>
FFREQH	High byte of the 16-bit Flash frequency value.

**Table 77. Flash Frequency Low Byte Register (FFREQL)**

Bit	7	6	5	4	3	2	1	0
Field	FFREQL							
RESET	0							
R/W	R/W							
Address	FFBH							

Bit	Description
[7:0]	<b>Flash Frequency High Byte</b>
FFREQL	Low byte of the 16-bit Flash frequency value.

Table 80. 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. See Tables 81 and 82.

Table 81. 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.

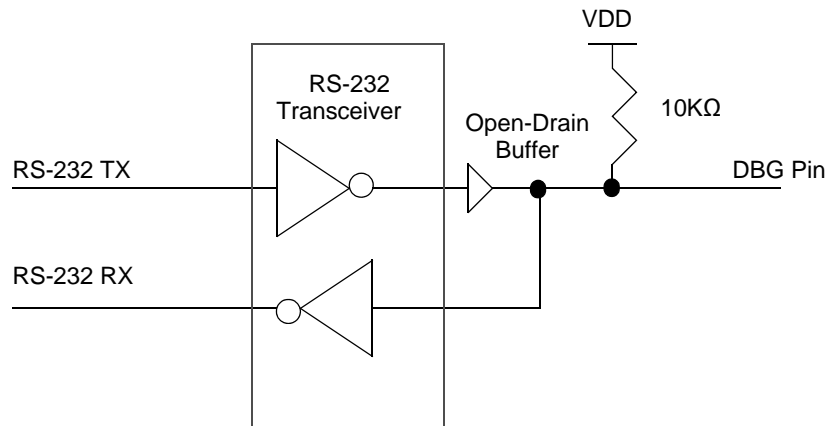
Bit	Description
[7] WDT_RES	<b>Watchdog Timer Reset</b> 0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request. 1 = Watchdog Timer time-out causes a system reset. This is the default setting for unprogrammed (erased) Flash.
[6] WDT_AO	<b>Watchdog Timer Always On</b> 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 can only be disabled by a reset. This is the default setting for unprogrammed (erased) Flash.
[5:4] OSC_SEL	<b>OSCILLATOR Mode Selection</b> 00 = On-chip oscillator configured for use with external RC networks (<4MHz). 01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0MHz). 10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5MHz to 5.0MHz). 11 = Maximum power for use with high frequency crystals (5.0MHz to 20.0MHz). This is the default setting for unprogrammed (erased) Flash.

Bit	Description (Continued)
[1:0]	<b>Filter Select</b>
FilterSely	2-bit selection for the clock filter mode. 00 = No filter. 01 = Filter low level noise on high level signal. 10 = Filter high level noise on low level signal. 11 = Filter both.
Notes: x indicates bit values 3–1; y indicates bit values 1–0.	

► **Note:** The bit values used in Table 89 are set at factory and no calibration is required.

**Table 90. ClkFlt Delay Control Definition**

DlyCtl3, DlyCtl2, DlyCtl1	Low Noise Pulse on High Signal (ns)	High Noise Pulse on Low Signal (ns)
000	5	5
001	7	7
010	9	9
011	11	11
100	13	13
101	17	17
110	20	20
111	25	25
Note: The variation is about 30%.		



**Figure 22. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #2 of 2**

## **DEBUG Mode**

The operating characteristics of the devices in DEBUG Mode are:

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

### **Entering DEBUG Mode**

- The device enters DEBUG Mode after the eZ8 CPU executes a Breakpoint (BRK) instruction
- If the DBG pin is held low during the most recent clock cycle of system reset, the device enters DEBUG Mode on exiting system reset

### **Exiting DEBUG Mode**

The device exits DEBUG Mode following any of these operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset

Figures 29 and 30 provide information about each of the eZ8 CPU instructions.

		Lower Nibble (Hex)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Upper Nibble (Hex)	0	1.1 BRK	2.2 SRP IM	2.3 ADD r1,r2	2.4 ADD r1,lr2	3.3 ADD R2,R1	3.4 ADD IR2,R1	3.3 ADD R1,IM	3.4 ADD IR1,IM	4.3 ADDX ER2,ER1	4.3 ADDX IM,ER1	2.3 DJNZ r1,X	2.2 JR cc,X	2.2 LD r1,IM	3.2 JP cc,DA	1.2 INC r1	1.2 NOP
	1	2.2 RLC R1	2.3 RLC IR1	2.3 ADC r1,r2	2.4 ADC r1,lr2	3.3 ADC R2,R1	3.4 ADC IR2,R1	3.3 ADC R1,IM	3.4 ADC IR1,IM	4.3 ADCX ER2,ER1	4.3 ADCX IM,ER1						See 2nd Op Code Map
	2	2.2 INC R1	2.3 INC IR1	2.3 SUB r1,r2	2.4 SUB r1,lr2	3.3 SUB R2,R1	3.4 SUB IR2,R1	3.3 SUB R1,IM	3.4 SUB IR1,IM	4.3 SUBX ER2,ER1	4.3 SUBX IM,ER1						
	3	2.2 DEC R1	2.3 DEC IR1	2.3 SBC r1,r2	2.4 SBC r1,lr2	3.3 SBC R2,R1	3.4 SBC IR2,R1	3.3 SBC R1,IM	3.4 SBC IR1,IM	4.3 SBCX ER2,ER1	4.3 SBCX IM,ER1						
	4	2.2 DA R1	2.3 DA IR1	2.3 OR r1,r2	2.4 OR r1,lr2	3.3 OR R2,R1	3.4 OR IR2,R1	3.3 OR R1,IM	3.4 OR IR1,IM	4.3 ORX ER2,ER1	4.3 ORX IM,ER1						
	5	2.2 POP R1	2.3 POP IR1	2.3 AND r1,r2	2.4 AND r1,lr2	3.3 AND R2,R1	3.4 AND IR2,R1	3.3 AND R1,IM	3.4 AND IR1,IM	4.3 ANDX ER2,ER1	4.3 ANDX IM,ER1						1.2 WDT
	6	2.2 COM R1	2.3 COM IR1	2.3 TCM r1,r2	2.4 TCM r1,lr2	3.3 TCM R2,R1	3.4 TCM IR2,R1	3.3 TCM R1,IM	3.4 TCM IR1,IM	4.3 TCMX ER2,ER1	4.3 TCMX IM,ER1						1.2 STOP
	7	2.2 PUSH R2	2.3 PUSH IR2	2.3 TM r1,r2	2.4 TM r1,lr2	3.3 TM R2,R1	3.4 TM IR2,R1	3.3 TM R1,IM	3.4 TM IR1,IM	4.3 TMX ER2,ER1	4.3 TMX IM,ER1						1.2 HALT
	8	2.5 DECW RR1	2.6 DECW IRR1	2.5 LDE r1,lr2	2.9 LDEI lr1,lr2	3.2 LDX r1,ER2	3.3 LDX lr1,ER2	3.4 LDX IRR2,R1	3.5 LDX IRR2,IR1	3.4 LDX r1,rr2,X	3.4 LDX rr1,r2,X						1.2 DI
	9	2.2 RL R1	2.3 RL IR1	2.5 LDE r2,lr1	2.9 LDEI lr2,lr1	3.2 LDX r2,ER1	3.3 LDX lr2,ER1	3.4 LDX R2,IRR1	3.5 LDX IRR2,IRR1	3.3 LEA r1,r2,X	3.5 LEA rr1,rr2,X						1.2 EI
	A	2.5 INCW RR1	2.6 INCW IRR1	2.3 CP r1,r2	2.4 CP r1,lr2	3.3 CP R2,R1	3.4 CP IR2,R1	3.3 CP R1,IM	3.4 CP IR1,IM	4.3 CPX ER2,ER1	4.3 CPX IM,ER1						1.4 RET
	B	2.2 CLR R1	2.3 CLR IR1	2.3 XOR r1,r2	2.4 XOR r1,lr2	3.3 XOR R2,R1	3.4 XOR IR2,R1	3.3 XOR R1,IM	3.4 XOR IR1,IM	4.3 XORX ER2,ER1	4.3 XORX IM,ER1						1.5 IRET
	C	2.2 RRC R1	2.3 RRC IR1	2.5 LDC r1,lr2	2.9 LDCI lr1,lr2	2.3 JP IRR1	2.9 LDC lr1,lr2		3.4 LD r1,r2,X	3.2 PUSHX ER2							1.2 RCF
	D	2.2 SRA R1	2.3 SRA IR1	2.5 LDC r2,lr1	2.9 LDCI lr2,lr1	2.6 CALL IRR1	2.2 BSWAP R1	3.3 CALL DA	3.4 LD r2,r1,X	3.2 POPX ER1							1.2 SCF
	E	2.2 RR R1	2.3 RR IR1	2.2 BIT p,b,r1	2.3 LD r1,lr2	3.2 LD R2,R1	3.3 LD IR2,R1	3.2 LD R1,IM	3.3 LD IR1,IM	4.2 LDX ER2,ER1	4.2 LDX IM,ER1						1.2 CCF
	F	2.2 SWAP R1	2.3 SWAP IR1	2.6 TRAP Vector	2.3 LD lr1,r2	2.8 MULT RR1	3.3 LD R2,IR1	3.3 BTJ p,b,r1,X	3.4 BTJ p,b,lr1,X								

Figure 29. First Op Code Map

General Purpose I/O Port Output Timing

Figure 34 and Table 125 provide timing information for the GPIO port pins.

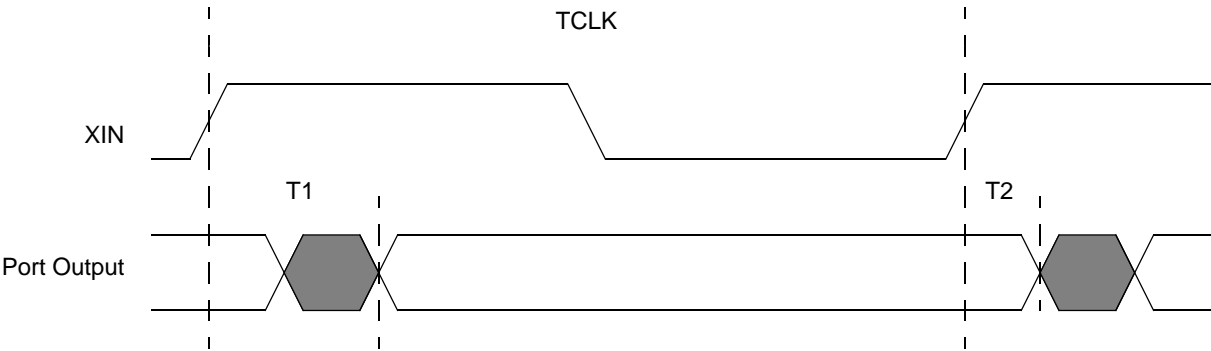


Figure 34. GPIO Port Output Timing

Table 125. GPIO Port Output Timing

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
GPIO Port Pins			
T <sub>1</sub>	XIN Rise to Port Output Valid Delay	–	15
T <sub>2</sub>	XIN Rise to Port Output Hold Time	2	–

Table 129 lists the pin count by package.

**Table 129. Package and Pin Count Description**

<b>Package</b>	<b>Pin Count</b>	
	<b>20</b>	<b>28</b>
PDIP	√	√
QFN	√	√
SOIC	√	√
SSOP	√	√

- compare - extended addressing 166
- compare mode 89
- compare with carry 166
- compare with carry - extended addressing 166
- complement 169
- complement carry flag 167, 168
- condition code 164
- continuous mode 89
- Control Registers 14, 17
- counter modes 89
- CP 166
- CPC 166
- CPCX 166
- CPU and peripheral overview 4
- CPU control instructions 168
- CPX 166
- current measurement
  - architecture 98
  - operation 99
- Customer Feedback Form 239
- Customer Information 239

## **D**

- DA 164, 166
- data memory 16
- DC characteristics 185
- debugger, on-chip 139
- DEC 166
- decimal adjust 166
- decrement 166
- decrement and jump non-zero 169
- decrement word 166
- DECW 166
- destination operand 165
- device, port availability 33
- DI 168
- direct address 164
- disable interrupts 168
- DJNZ 169
- dst 165

## **E**

- EI 168
- electrical characteristics 184
  - GPIO input data sample timing 195
  - watch-dog timer 194
- electrical noise 98
- enable interrupt 168
- ER 164
- extended addressing register 164
- external pin reset 25
- eZ8 CPU features 4
- eZ8 CPU instruction classes 166
- eZ8 CPU instruction notation 164
- eZ8 CPU instruction set 162
- eZ8 CPU instruction summary 171

## **F**

- FCTL register 119, 126, 127, 228
- features, Z8 Encore! 1
- first opcode map 182
- FLAGS 165
- flags register 165
- flash
  - controller 4
  - option bit address space 127
  - option bit configuration - reset 124
  - program memory address 0000H 127
  - program memory address 0001H 128
- flash memory 108
  - byte programming 116
  - code protection 114
  - configurations 108
  - control register definitions 118, 126
  - controller bypass 117
  - flash control register 119, 126, 127, 228
  - flash option bits 115
  - flash status register 120
  - flow chart 113
  - frequency high and low byte registers 123
  - mass erase 117
  - operation 112
  - operation timing 114
  - page erase 117

- read program memory CRC (0EH) 147
- read register (09H) 146
- read runtime counter (03H) 145
- 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 180
  - first 182
  - second after 1FH 183
- operation 100
  - current measurement 99
  - voltage measurement timing diagram 100
- Operational Description 21, 30, 33, 53, 68, 92, 98, 106, 108, 124, 134, 139, 151, 157, 161
- OR 169
- ordering information 200
- ORX 169
- oscillator signals 12

## **P**

- p 164
- Packaging 199
- part selection guide 2
- PC 165
- peripheral AC and DC electrical characteristics 190
- 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) 195

- port output timing, GPIO 196
- 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, 222, 223, 224, 225
- PxCTL register 41, 222, 223, 224, 225

## **R**

- R 165
- r 164
- RA
  - register address 165
- RCF 167, 168
- register 165
  - flash control (FCTL) 119, 126, 127, 228
  - flash high and low byte (FFREQH and FRE-EQL) 123
  - flash page select (FPS) 121, 122
  - flash status (FSTAT) 120
  - GPIO port A-H address (PxADDR) 40, 222, 223, 224, 225
  - GPIO port A-H alternate function sub-registers 42
  - GPIO port A-H control address (PxCTL) 41, 222, 223, 224, 225
  - GPIO port A-H data direction sub-registers 41
  - OCD control 148
  - OCD status 150
  - watch-dog timer control (WDTCTL) 95, 107, 154, 217, 218, 226
  - watchdog timer control (WDTCTL) 29
  - watch-dog timer reload high byte (WDTH) 227
  - watchdog timer reload high byte (WDTH) 96
  - watch-dog timer reload low byte (WDTL) 227
  - watchdog timer reload low byte (WDTL) 97
  - watch-dog timer reload upper byte (WDTU) 227

- watchdog timer reload upper byte (WDTU) 96
- register file 14
- register pair 165
- register pointer 165
- registers
  - ADC channel 1 102
  - ADC data high byte 103
  - ADC data low bit 103, 104, 105
- reset
  - and stop mode characteristics 22
  - and stop mode recovery 21
  - carry flag 167
  - sources 23
- RET 169
- return 169
- RL 169
- RLC 169
- rotate and shift instructions 169
- rotate left 169
- rotate left through carry 169
- rotate right 170
- rotate right through carry 170
- RP 165
- RR 165, 170
- rr 165
- RRC 170

## **S**

- SBC 167
- SCF 167, 168
- second opcode map after 1FH 183
- set carry flag 167, 168
- set register pointer 168
- shift right arithmetic 170
- shift right logical 170
- signal descriptions 11
- software trap 169
- source operand 165
- SP 165
- SRA 170
- src 165
- SRL 170
- SRP 168

- stack pointer 165
- STOP 168
- stop mode 30, 168
- stop mode recovery
  - sources 26
  - using a GPIO port pin transition 27, 28
  - using watch-dog timer time-out 27
- SUB 167
- subtract 167
- subtract - extended addressing 167
- subtract with carry 167
- subtract with carry - extended addressing 167
- SUBX 167
- SWAP 170
- swap nibbles 170
- symbols, additional 165

## **T**

- Table 134. Power Consumption Reference Table 197
- TCM 167
- TCMX 167
- test complement under mask 167
- test complement under mask - extended addressing 167
- test under mask 167
- test under mask - extended addressing 167
- timing diagram, voltage measurement 100
- timer signals 11
- timers 68
  - architecture 68
  - block diagram 69
  - capture mode 77, 78, 89, 90
  - capture/compare mode 81, 89
  - compare mode 79, 89
  - continuous mode 70, 89
  - counter mode 71, 72
  - counter modes 89
  - gated mode 80, 89
  - one-shot mode 69, 89
  - operating mode 69
  - PWM mode 74, 75, 89, 90
  - reading the timer count values 82