



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	25
Program Memory Size	12KB (12K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1233sj020eg

Calibration and Compensation	101
ADC Control Register Definitions	101
ADC Control Register 0	102
ADC Data High Byte Register	103
ADC Data Low Bits Register	103
Sample Settling Time Register	104
Sample Time Register	105
Comparator	106
Operation	106
Comparator Control Register Definitions	107
Flash Memory	108
Data Memory Address Space	111
Flash Information Area	111
Operation	112
Flash Operation Timing Using the Flash Frequency Registers	114
Flash Code Protection Against External Access	114
Flash Code Protection Against Accidental Program and Erasure	114
Byte Programming	116
Page Erase	117
Mass Erase	117
Flash Controller Bypass	117
Flash Controller Behavior in Debug Mode	117
NVDS Operational Requirements	118
Flash Control Register Definitions	118
Flash Control Register	119
Flash Status Register	120
Flash Page Select Register	121
Flash Sector Protect Register	122
Flash Frequency High and Low Byte Registers	123
Flash Option Bits	124
Operation	124
Option Bit Configuration by Reset	124
Option Bit Types	125
Flash Option Bit Control Register Definitions	126
Trim Bit Address Register	126
Trim Bit Data Register	126
Flash Option Bit Address Space	127
Trim Bit Address Space	129
Nonvolatile Data Storage	134

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	

Nonvolatile Data Storage

The Nonvolatile Data Storage (NVDS) function 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) function, with an accuracy of $\pm 4\%$ full voltage/temperature range, is a trimmable clock source that requires no external components.

External Crystal Oscillator

The crystal oscillator circuit provides highly accurate clock frequencies using an external crystal, ceramic resonator or RC network.

10-Bit Analog-to-Digital Converter

The optional 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.

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 can be 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! F0830 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.

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No.
Trim Bit Control				
FF6	Trim bit address	TRMADR	00	126
FF7	Trim data	TRMDR	XX	127
Flash Memory Controller				
FF8	Flash control	FCTL	00	119
FF8	Flash status	FSTAT	00	120
FF9	Flash page select	FPS	00	121
	Flash sector protect	FPROT	00	122
FFA	Flash programming frequency high byte	FFREQH	00	123
FFB	Flash programming frequency low byte	FFREQL	00	123
eZ8 CPU				
FFC	Flags	—	XX	Refer to the eZ8 CPU Core User Manual (UM0128)
FFD	Register pointer	RP	XX	
FFE	Stack pointer high byte	SPH	XX	
FFF	Stack pointer low byte	SPL	XX	
Note: XX = Undefined.				

Architecture

Figure 8 displays a simplified block diagram of a GPIO port pin. In this figure, the ability to accommodate alternate functions and variable port current drive strength is not displayed.

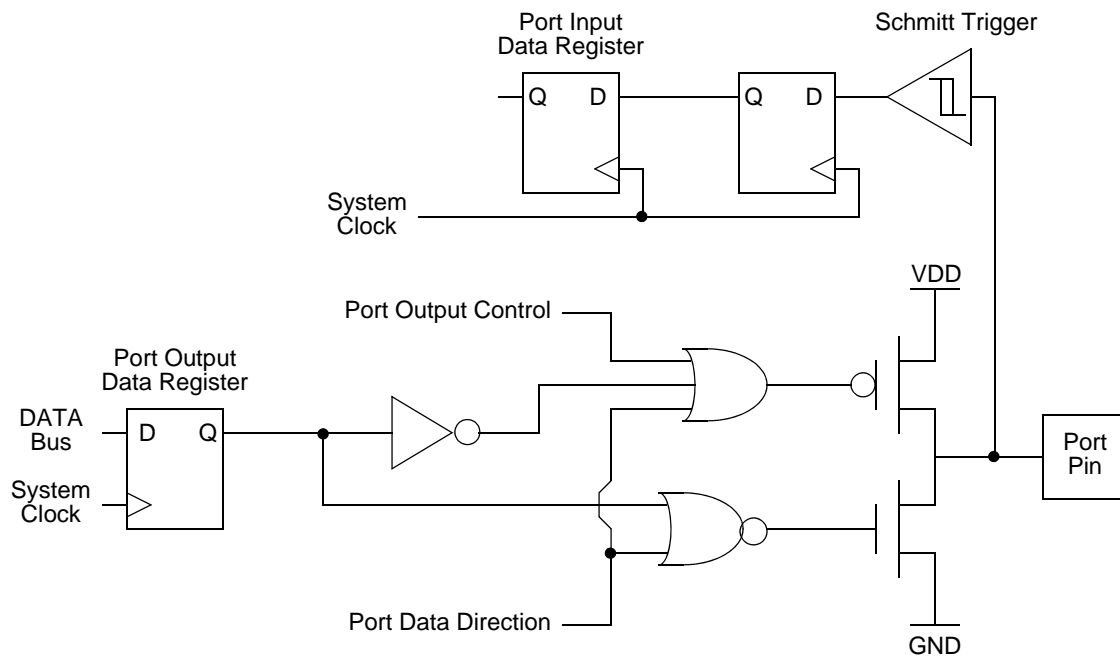


Figure 8. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many of the GPIO port pins can be used for general purpose input/output and access to on-chip peripheral functions such as the timers and serial communication devices. The Port A–D Alternate Function subregisters configure these pins for either GPIO or Alternate function operation. When a pin is configured for Alternate function, control of the port pin direction (input/output) is passed from the Port A–D data direction registers to the Alternate function assigned to this pin. [Table 16](#) on page 36 lists the alternate functions possible with each port pin. The alternate function associated at a pin is defined through Alternate Function subregisters AFS1 and AFS2.

The crystal oscillator functionality is not controlled by the GPIO block. When the crystal oscillator is enabled in the oscillator control block, the GPIO functionality of PA0 and PA1 is overridden. In that case, pins PA0 and PA1 functions as input and output for the crystal oscillator.

Table 16. Port Alternate Function Mapping (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port B ²	PB0	Reserved		AFS1[0]: 0
		ANA0	ADC analog input	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1	ADC analog input	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2	ADC analog input	AFS1[2]: 1
	PB3	CLKIN	External input clock	AFS1[3]: 0
		ANA3	ADC analog input	AFS1[3]: 1
	PB4	Reserved		AFS1[4]: 0
		ANA7	ADC analog input	AFS1[4]: 1
	PB5	Reserved		AFS1[5]: 0
		V _{REF}	ADC reference voltage	AFS1[5]: 1
	PB6	Reserved		AFS1[6]: 0
		Reserved		AFS1[6]: 1
	PB7	Reserved		AFS1[7]: 0
		Reserved		AFS1[7]: 1

Notes:

1. Because there is only a single alternate function for each Port A and Port D (PD0) pin, the Alternate Function Set registers are not implemented for Port A and Port D (PD0). Enabling alternate function selections (as described in the [Port A–D Alternate Function Subregisters](#) section on page 42) automatically enables the associated alternate function.
2. Because there are at most two choices of alternate functions for any Port B pin, the AFS2 Alternate Function Set Register is implemented but is not used to select the function. Additionally, alternate function selection (as described in the [Port A–D Alternate Function Subregisters](#) section on page 42) must also be enabled.
3. Because there are at most two choices of alternate functions for any Port C pin, the AFS2 Alternate Function Set Register is implemented but is not used to select the function. Additionally, alternate function selection (as described in the [Port A–D Alternate Function Subregisters](#) section on page 42) must also be enabled.

Port A–D Alternate Function Set 2 Subregisters

The Port A–D Alternate Function Set 2 Subregister, shown in Table 28, is accessed through the Port A–D Control Register by writing 08H to the Port A–D Address Register. The Alternate Function Set 2 subregisters select the alternate function available at a port pin. Alternate functions selected by setting or clearing bits in this register are defined in Table 16 in the GPIO Alternate Functions section on page 34.

► **Note:** Alternate function selection on the port pins must also be enabled, as described in the Port A–D Alternate Function Subregisters section on page 42.

Table 28. Port A–D Alternate Function Set 2 Subregisters (PxAFS2)

Bit	7	6	5	4	3	2	1	0
Field	PAFS27	PAFS26	PAFS25	PAFS24	PAFS23	PAFS22	PAFS21	PAFS20
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	If 08H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0]	Port Alternate Function Set 2
PAFS2x	0 = The Port Alternate function is selected, as defined in Table 16 in the <u>GPIO Alternate Functions</u> section on page 34. 1 = The Port Alternate function is selected, as defined in Table 16 in the <u>GPIO Alternate Functions</u> section on page 34.

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

Port A–D Output Data Register

The Port A–D Output Data Register, shown in Table 30, controls the output data to the pins.

Table 30. Port A–D Output Data Register (PxOUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
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	FD3H, FD7H, FDBH, FDFH							

Bit	Description
[7:0]	Port Output Data
PxOUT	<p>These bits contain the data to be driven to the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for Alternate function operation.</p> <p>0 = Drive a logical 0 (Low).</p> <p>1 = Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding port output Control Register bit to 1.</p>

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

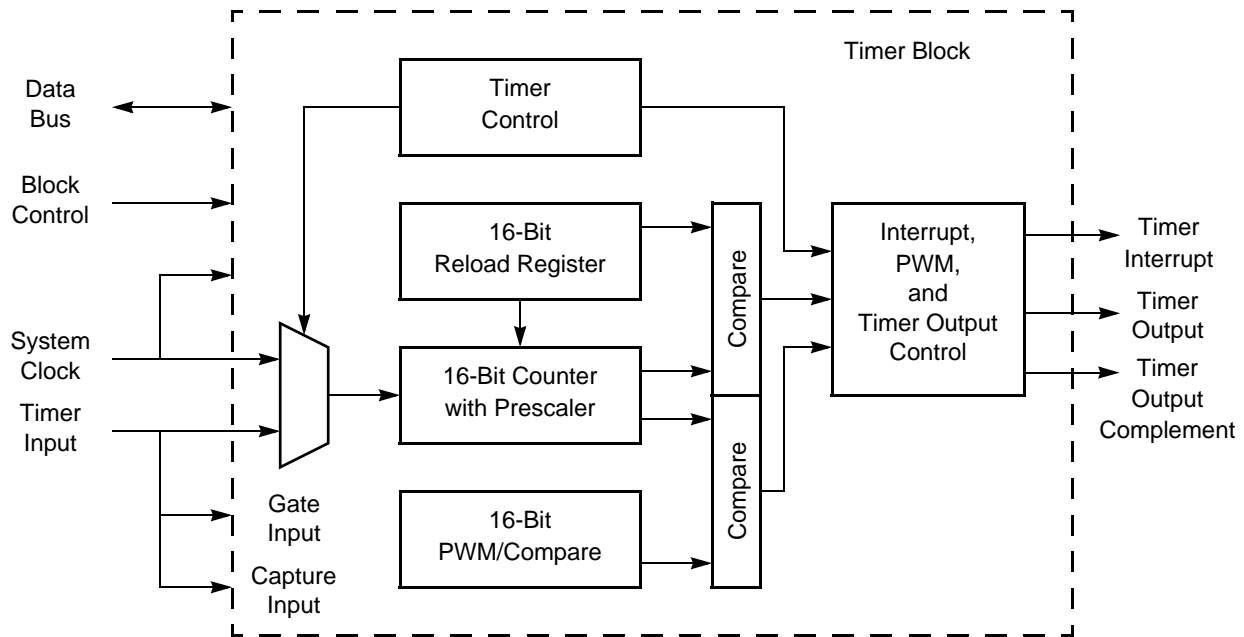


Figure 10. Timer Block Diagram

Operation

The timers are 16-bit up-counters. Minimum time-out delay is set by loading the value 0001H into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value 0000H into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the Timer reaches FFFFH, the timer resets back to 0000H and continues counting.

Timer Operating Modes

The timers can be configured to operate in the following modes:

ONE-SHOT Mode

In ONE-SHOT Mode, the timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the reload value, the timer generates an interrupt and the count value in the Timer High and Low Byte registers is reset to 0001H. The timer is automatically disabled and stops counting.

Additionally, if the timer output alternate function is enabled, the timer output pin changes state for one system clock cycle (from Low to High or from High to Low) upon timer

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT Mode equation to determine the first PWM time-out period.

If TPOL bit is set to 0, the ratio of the PWM output high time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL bit is set to 1, the ratio of the PWM output high time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

PWM DUAL OUTPUT Mode

In PWM DUAL OUTPUT Mode, the timer outputs a PWM output signal pair (basic PWM signal and its complement) through two GPIO port pins. The timer input is the system clock. The timer first counts up to 16-bit PWM match value stored in the timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the timer output toggles. The timer continues counting until it reaches the reload 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.

If the TPOL bit in the Timer Control Register is set to 1, the timer output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The timer output signal returns to a High (1) after the timer reaches the reload value and is reset to 0001H.

If the TPOL bit in the Timer Control Register is set to 0, the timer output signal begins as a Low (0) and transitions to a High (1) when the timer value matches the PWM value. The timer output signal returns to a Low (0) after the timer reaches the reload value and is reset to 0001H.

The timer also generates a second PWM output signal: the timer output complement. The timer output complement is the complement of the timer output PWM signal. A programmable deadband delay can be configured to time delay (0 to 128 system clock cycles) PWM output transitions on these two pins from a Low to a High (inactive to active) to ensure a time gap between the deassertion of one PWM output to the assertion of its complement.

tion and reload events. The user can configure the timer interrupt to be generated only at the input deassertion event or the reload event by setting the TICONFIG field of the TxCTL1 Register.

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. Assert the timer input signal to initiate the counting.

CAPTURE/COMPARE Mode

In CAPTURE/COMPARE Mode, the timer begins counting on the first external timer input transition. The acceptable transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the system clock.

Every subsequent acceptable transition (after the first) of the timer input signal, captures the current count value. The capture value is written to the timer PWM High and Low Byte registers. When the capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to 0001H and the counting resumes. The INPCAP bit in the TxCTL1 Register is set to indicate that the timer interrupt is caused by an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the compare 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/COMPARE Mode and for initiating the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for CAPTURE/COMPARE Mode.
 - Set the prescale value.
 - Set the capture edge (rising or falling) for the timer input.
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the compare value.
4. Enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt are generated for both input capture and Reload events. The user can 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.
5. Configure the associated GPIO port pin for the timer input alternate function.

Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers, shown in Tables 54 and 55, control PWM operations. These registers also store the capture values for the CAPTURE and CAPTURE/COMPARE modes.

Table 54. Timer 0–1 PWM High Byte Register (TxPWMH)

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
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	F04H, F0CH							

Table 55. Timer 0–1 PWM Low Byte Register (TxPWML)

Bit	7	6	5	4	3	2	1	0
Field	PWML							
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	F05H, F0DH							

Bit	Description
[7:0]	Pulse Width Modulator High and Low Bytes
PWMH, PWML	These two bytes, {PWMH[7:0], PWML[7:0]}, form a 16-bit value that is compared to the current 16-bit timer count. When a match occurs, the PWM output changes state. The PWM output value is set by the TPOL bit in the Timer Control Register (TxCTL1). The TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in capture or CAPTURE/COMPARE modes.

Bit	Description (Continued)
[6] TPOL (cont'd)	<p>PWM DUAL OUTPUT Mode</p> <p>0 = Timer output is forced Low (0) and timer output complement is forced High (1), when the timer is disabled. When enabled and the PWM count matches, the timer output is forced High (1) and forced Low (0) when enabled and reloaded. When enabled and the PWM count matches, the timer output complement is forced Low (0) and forced High (1) when enabled and reloaded.</p> <p>1 = Timer output is forced High (1) and timer output complement is forced Low (0) when the timer is disabled. When enabled and the PWM count matches, the timer output is forced Low (0) and forced High (1) when enabled and reloaded. When enabled and the PWM count matches, the timer output complement is forced High (1) and forced Low (0) when enabled and reloaded. The PWMD field in the TxCTL0 register determines an optional added delay on the assertion (Low to High) transition of both timer output and timer output complement for deadband generation.</p> <p>CAPTURE RESTART Mode</p> <p>0 = Count is captured on the rising edge of the timer input signal.</p> <p>1 = Count is captured on the falling edge of the timer input signal.</p> <p>COMPARATOR COUNTER Mode</p> <p>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>Caution: When the timer output alternate function TxOUT on a GPIO port pin is enabled, TxOUT will change to whatever state the TPOL bit is in. The timer does not need to be enabled for that to happen. Additionally, the port data direction sub register is not needed to be set to output on TxOUT. Changing the TPOL bit when the timer is enabled and running does not immediately change the polarity TxOUT.</p>
[5:3] PRES	<p>Prescale Value</p> <p>The timer input clock is divided by 2^{PRES}, where PRES can be set from 0 to 7. The prescaler is reset each time the timer is disabled. This reset ensures proper clock division each time the timer is restarted.</p> <p>000 = Divide by 1.</p> <p>001 = Divide by 2.</p> <p>010 = Divide by 4.</p> <p>011 = Divide by 8.</p> <p>100 = Divide by 16.</p> <p>101 = Divide by 32.</p> <p>110 = Divide by 64.</p> <p>111 = Divide by 128.</p>

Watchdog Timer

The Watchdog Timer (WDT) protects from corrupted or unreliable software, power faults and other system-level problems which can place the Z8 Encore! F0830 Series devices into unsuitable operating states. The features of the Watchdog Timer include:

- On-chip RC oscillator
- A selectable time-out response: reset or interrupt
- 24-bit programmable time-out value

Operation

The Watchdog Timer is a retriggerable one-shot timer that resets or interrupts the Z8 Encore! F0830 Series devices when the WDT reaches its terminal count. The WDT uses a dedicated on-chip RC oscillator as its clock source. The WDT operates only in two modes: ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. Perform an enable by executing the WDT instruction or by setting the WDT_AO Flash option bit. The WDT_AO bit forces the WDT to operate immediately on reset, even if a WDT instruction has not been executed.

The Watchdog Timer is a 24-bit reloadable downcounter that uses three 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is calculated using the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{10}$$

where the WDT reload value is the 24-bit decimal value provided by {WDTU[7:0], WDTH[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10 KHz. The Watchdog Timer cannot be refreshed after it reaches 000002H. The WDT reload value must not be set to values below 000004H. Table 58 provides information about approximate time-out delays for the minimum and maximum WDT reload values.

Table 58. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value (Hex)	WDT Reload Value (Decimal)	Approximate Time-Out Delay (with 10 KHz Typical WDT Oscillator Frequency)	
		Typical	Description
000004	4	400µs	Minimum time-out delay
000400	1024	102ms	Default time-out delay
FFFFFF	16,777,215	28 minutes	Maximum time-out delay

Flash Control Register

The Flash Controller must be unlocked using the Flash Control Register before programming or erasing Flash memory. Writing the sequence 73H 8CH, sequentially, to the Flash Control Register unlocks the Flash Controller. When the Flash Controller is unlocked, Flash memory can be enabled for mass erase or page erase by writing the appropriate enable command to the FCTL. Page erase applies only to the active page selected in Flash Page Select Register. Mass erase is enabled only through the On-Chip Debugger. Writing an invalid value or an invalid sequence returns the Flash Controller to its Locked state. The write-only Flash Control Register shares its register file address with the read-only Flash Status Register.

Table 72. Flash Control Register (FCTL)

Bit	7	6	5	4	3	2	1	0
Field	FCMD							
RESET	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	FF8H							

Bit	Description
[7:0]	Flash Command
FCMD	73H = First unlock command. 8CH = Second unlock command. 95H = Page erase command (must be third command in sequence to initiate page erase). 63H = Mass erase command (must be third command in sequence to initiate mass erase). 5EH = Enable Flash Sector Protect Register access.

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		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
ADC dst, src	$\text{dst} \leftarrow \text{dst} + \text{src} + \text{C}$	r	r	12	*	*	*	*	0	*	2	3
		r	lr	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	$\text{dst} \leftarrow \text{dst} + \text{src} + \text{C}$	ER	ER	18	*	*	*	*	0	*	4	3
		ER	IM	19							4	3
ADD dst, src	$\text{dst} \leftarrow \text{dst} + \text{src}$	r	r	02	*	*	*	*	0	*	2	3
		r	lr	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	$\text{dst} \leftarrow \text{dst} + \text{src}$	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 113. 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	$\text{dst}[\text{bit}] \leftarrow \text{p}$	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 $\text{src}[\text{bit}] = \text{p}$ $\text{PC} \leftarrow \text{PC} + \text{X}$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJNZ bit, src, dst	if $\text{src}[\text{bit}] = 1$ $\text{PC} \leftarrow \text{PC} + \text{X}$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
BTJZ bit, src, dst	if $\text{src}[\text{bit}] = 0$ $\text{PC} \leftarrow \text{PC} + \text{X}$		r	F6	–	–	–	–	–	–	3	3
			lr	F7							3	4
CALL dst	$\text{SP} \leftarrow \text{SP} - 2$	IRR		D4	–	–	–	–	–	–	2	6
	@SP \leftarrow PC PC \leftarrow dst	DA		D6							3	3
CCF	$\text{C} \leftarrow \sim \text{C}$			EF	*	–	–	–	–	–	1	2

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 113. 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		
DJNZ dst, RA	$\text{dst} \leftarrow \text{dst} - 1$ if $\text{dst} \neq 0$ $\text{PC} \leftarrow \text{PC} + X$	r		0A–FA	–	–	–	–	–	–	2	3
EI	$\text{IRQCTL}[7] \leftarrow 1$			9F	–	–	–	–	–	–	1	2
HALT	HALT Mode			7F	–	–	–	–	–	–	1	2
INC dst	$\text{dst} \leftarrow \text{dst} + 1$	R		20	–	*	*	–	–	–	2	2
		IR		21							2	3
		r		0E–FE							1	2
INCW dst	$\text{dst} \leftarrow \text{dst} + 1$	RR		A0	–	*	*	*	–	–	2	5
		IRR		A1							2	6
IRET	$\text{FLAGS} \leftarrow @\text{SP}$ $\text{SP} \leftarrow \text{SP} + 1$ $\text{PC} \leftarrow @\text{SP}$ $\text{SP} \leftarrow \text{SP} + 2$ $\text{IRQCTL}[7] \leftarrow 1$			BF	*	*	*	*	*	*	1	5
JP dst	$\text{PC} \leftarrow \text{dst}$	DA		8D	–	–	–	–	–	–	3	2
		IRR		C4							2	3
JP cc, dst	if cc is true $\text{PC} \leftarrow \text{dst}$	DA		0D–FD	–	–	–	–	–	–	3	2
JR dst	$\text{PC} \leftarrow \text{PC} + X$	DA		8B	–	–	–	–	–	–	2	2
JR cc, dst	if cc is true $\text{PC} \leftarrow \text{PC} + X$	DA		0B–FB	–	–	–	–	–	–	2	2

Note: Flags Notation:

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

– = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Hex Address: FDF**Table 183. Port D Output Data Register (PDOUT)**

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
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	FDFH							

Hex Addresses: FE0–FEF

This address range is reserved.

Watchdog Timer

For more information about the Watchdog Timer registers, see the [Watchdog Timer Control Register Definitions](#) section on page 95.

Hex Address: FF0

The Watchdog Timer Control Register address is shared with the read-only Reset Status Register.

Table 184. Watchdog Timer Control Register (WDTCTL)

Bit	7	6	5	4	3	2	1	0
Field	WDTUNLK							
RESET	X	X	X	X	X	X	X	X
R/W	W	W	W	W	W	W	W	W
Address	FF0H							

Table 185. Reset Status Register (RSTSTAT)

Bit	7	6	5	4	3	2	1	0
Field	POR	STOP	WDT	EXT	Reserved			
RESET	See Table 12 on page 29			0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	FF0H							

- 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