



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	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K 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	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f081ash020ec">https://www.e-xfl.com/product-detail/zilog/z8f081ash020ec</a>

## HALT Mode

Executing the eZ8 CPU's HALT instruction places the device into HALT mode, which powers down the CPU but leaves all other peripherals active. In HALT mode, the operating characteristics are:

- Primary oscillator is enabled and continues to operate.
- System clock is enabled and continues to operate.
- eZ8 CPU is stopped.
- Program counter (PC) stops incrementing.
- Watchdog Timer's internal RC oscillator continues to operate.
- If enabled, the Watchdog Timer continues to operate.
- All other on-chip peripherals continue to operate, if enabled.

The eZ8 CPU can be brought out of HALT mode by any of the following operations:

- Interrupt
- Watchdog Timer time-out (interrupt or reset)
- Power-On Reset
- Voltage Brownout reset
- External  $\overline{\text{RESET}}$  pin assertion

To minimize current in HALT mode, all GPIO pins that are configured as inputs must be driven to one of the supply rails ( $V_{CC}$  or GND).

## Peripheral-Level Power Control

In addition to the STOP and HALT modes, it is possible to disable each peripheral on each of the Z8 Encore! XP F082A Series devices. Disabling a given peripheral minimizes its power consumption.

## Power Control Register Definitions

The following sections define the Power Control registers.

### Power Control Register 0

Each bit of the following registers disables a peripheral block, either by gating its system clock input or by removing power from the block. The default state of the low-power

PA0 and PA6 contain two different timer functions, a timer input and a complementary timer output. Both of these functions require the same GPIO configuration, the selection between the two is based on the timer mode. See [Timers](#) on page 69 for more details.



**Caution:** *For pin with multiple alternate functions, it is recommended to write to the AFS1 and AFS2 sub-registers before enabling the alternate function via the AF sub-register. This prevents spurious transitions through unwanted alternate function modes.*

## Direct LED Drive

The Port C pins provide a current sinked output capable of driving an LED without requiring an external resistor. The output sinks current at programmable levels of 3 mA, 7 mA, 13 mA and 20 mA. This mode is enabled through the Alternate Function sub-register AFS1 and is programmable through the LED control registers. The LED Drive Enable (LEDEN) register turns on the drivers. The LED Drive Level (LEDLVLH and LEDLVLL) registers select the sink current.

For correct function, the LED anode must be connected to  $V_{DD}$  and the cathode to the GPIO pin. Using all Port C pins in LED drive mode with maximum current may result in excessive total current. See [Electrical Characteristics](#) on page 221 for the maximum total current for the applicable package.

## Shared Reset Pin

On the 20- and 28-pin devices, the PD0 pin shares function with a bi-directional reset pin. Unlike all other I/O pins, this pin does not default to GPIO function on power-up. This pin acts as a bi-directional reset until the software re-configures it. The PD0 pin is output-only when in GPIO mode.

On the 8-pin product versions, the reset pin is shared with PA2, but the pin is not limited to output-only when in GPIO mode.



**Caution:** *If PA2 on the 8-pin product is reconfigured as an input, ensure that no external stimulus drives the pin low during any reset sequence. Since PA2 returns to its RESET alternate function during system resets, driving it Low holds the chip in a reset state until the pin is released.*

## Shared Debug Pin

On the 8-pin version of this device only, the Debug pin shares function with the PA0 GPIO pin. This pin performs as a general purpose input pin on power-up, but the debug logic monitors this pin during the reset sequence to determine if the unlock sequence occurs. If the unlock sequence is present, the debug function is unlocked and the pin no longer func-

**Table 14. Port Alternate Function Mapping (Non 8-Pin Parts)**

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
<b>Port A</b>	PA0	T0IN/T0OUT*	Timer 0 Input/Timer 0 Output Complement	N/A
		Reserved		
	PA1	T0OUT	Timer 0 Output	
		Reserved		
	PA2	DE0	UART 0 Driver Enable	
		Reserved		
	PA3	CTS0	UART 0 Clear to Send	
		Reserved		
	PA4	RXD0/IRRX0	UART 0/IrDA 0 Receive Data	
		Reserved		
	PA5	TXD0/IRTX0	UART 0/IrDA 0 Transmit Data	
		Reserved		
	PA6	T1IN/T1OUT*	Timer 1 Input/Timer 1 Output Complement	
		Reserved		
	PA7	T1OUT	Timer 1 Output	
		Reserved		

**Note:** Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in [Port A–D Alternate Function Sub-Registers](#) on page 47 automatically enables the associated alternate function.

\* Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in [Timer Pin Signal Operation](#) on page 82.



**Caution:** *The following coding style that clears bits in the Interrupt Request registers is not recommended. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost.*

**Poor coding style that can result in lost interrupt requests:**

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



**Caution:** *To avoid missing interrupts, use the following coding style to clear bits in the Interrupt Request 0 register:*

**Good coding style that avoids lost interrupt requests:**

```
ANDX IRQ0, MASK
```

## Software Interrupt Assertion

Program code can generate interrupts directly. Writing a 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:** *The following coding style used to generate software interrupts by setting bits in the Interrupt Request registers is not recommended. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost.*

**Poor coding style that can result in lost interrupt requests:**

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



**Caution:** *To avoid missing interrupts, use the following coding style to set bits in the Interrupt Request registers:*

**Good coding style that avoids lost interrupt requests:**

```
ORX IRQ0, MASK
```

## Watchdog Timer Interrupt Assertion

The Watchdog Timer interrupt behavior is different from interrupts generated by other sources. The Watchdog Timer continues to assert an interrupt as long as the timeout condition continues. As it operates on a different (and usually slower) clock domain than the rest of the device, the Watchdog Timer continues to assert this interrupt for many system clocks until the counter rolls over.

U0RXI—UART 0 Receiver Interrupt Request

0 = No interrupt request is pending for the UART 0 receiver.

1 = An interrupt request from the UART 0 receiver is awaiting service.

U0TXI—UART 0 Transmitter Interrupt Request

0 = No interrupt request is pending for the UART 0 transmitter.

1 = An interrupt request from the UART 0 transmitter is awaiting service.

ADCI—ADC Interrupt Request

0 = No interrupt request is pending for the analog-to-digital Converter.

1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

## Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register ([Table 34](#)) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 1 register to determine if any interrupt requests are pending.

**Table 34. Interrupt Request 1 Register (IRQ1)**

BITS	7	6	5	4	3	2	1	0
FIELD	PA7VI	PA6CI	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
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
ADDR	FC3H							

PA7VI—Port A Pin 7 or LVD Interrupt Request

0 = No interrupt request is pending for GPIO Port A or LVD.

1 = An interrupt request from GPIO Port A or LVD.

PA6CI—Port A Pin 6 or Comparator Interrupt Request

0 = No interrupt request is pending for GPIO Port A or Comparator.

1 = An interrupt request from GPIO Port A or Comparator.

PAxI—Port A Pin x Interrupt Request

0 = No interrupt request is pending for GPIO Port A pin x.

1 = An interrupt request from GPIO Port A pin x is awaiting service.

where x indicates the specific GPIO Port pin number (0–5).

Reserved—Must be 0.

C3ENL—Port C3 Interrupt Request Enable Low Bit

C2ENL—Port C2 Interrupt Request Enable Low Bit

C1ENL—Port C1 Interrupt Request Enable Low Bit

C0ENL—Port C0 Interrupt Request Enable Low Bit

## Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register ([Table 45](#)) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A input pin.

**Table 45. Interrupt Edge Select Register (IRQES)**

BITS	7	6	5	4	3	2	1	0
FIELD	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
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
ADDR	FCDH							

IES<sub>x</sub>—Interrupt Edge Select *x*

0 = An interrupt request is generated on the falling edge of the PA<sub>x</sub> input.

1 = An interrupt request is generated on the rising edge of the PA<sub>x</sub> input.

where *x* indicates the specific GPIO Port pin number (0 through 7).

## Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) register ([Table 46](#)) determines the source of the PAD<sub>x</sub>S interrupts. The Shared Interrupt Select register selects between Port A and alternate sources for the individual interrupts.

Because these shared interrupts are edge-triggered, it is possible to generate an interrupt just by switching from one shared source to another. For this reason, an interrupt must be disabled before switching between sources.

# Timers

These Z8 Encore! XP® F082A Series products contain two 16-bit reloadable timers that can be used for timing, event counting, or generation of pulse-width modulated (PWM) signals. The timers' feature include:

- 16-bit reload counter.
- Programmable prescaler with prescale values from 1 to 128.
- PWM output generation.
- Capture and compare capability.
- External input pin for timer input, clock gating, or capture signal. External input pin signal frequency is limited to a maximum of one-fourth the system clock frequency.
- Timer output pin.
- Timer interrupt.

In addition to the timers described in this chapter, the Baud Rate Generator of the UART (if unused) may also provide basic timing functionality. For information on using the Baud Rate Generator as an additional timer, see [Universal Asynchronous Receiver/Transmitter](#) on page 97.

## Architecture

[Figure 9](#) on page 70 displays the architecture of the timers.





**Caution:** *The 24-bit WDT Reload Value must not be set to a value less than 000004H.*

**Table 58. Watchdog Timer Reload Upper Byte Register (WDTU)**

BITS	7	6	5	4	3	2	1	0
FIELD	WDTU							
RESET	00H							
R/W	R/W*							
ADDR	FF1H							
R/W* - Read returns the current WDT count value. Write sets the appropriate Reload Value.								

WDTU—WDT Reload Upper Byte  
Most-significant byte (MSB), Bits[23:16], of the 24-bit WDT reload value.

**Table 59. Watchdog Timer Reload High Byte Register (WDTH)**

BITS	7	6	5	4	3	2	1	0
FIELD	WDTH							
RESET	04H							
R/W	R/W*							
ADDR	FF2H							
R/W* - Read returns the current WDT count value. Write sets the appropriate Reload Value.								

WDTH—WDT Reload High Byte  
Middle byte, Bits[15:8], of the 24-bit WDT reload value.

**Table 60. Watchdog Timer Reload Low Byte Register (WDTL)**

BITS	7	6	5	4	3	2	1	0
FIELD	WDTL							
RESET	00H							
R/W	R/W*							
ADDR	FF3H							
R/W* - Read returns the current WDT count value. Write sets the appropriate Reload Value.								

WDTL—WDT Reload Low  
Least significant byte (LSB), Bits[7:0], of the 24-bit WDT reload value.

value 63H to the Flash Control register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. Using the On-Chip Debugger, poll the Flash Status register to determine when the Mass Erase operation is complete. When the Mass Erase is complete, the Flash Controller returns to its locked state.

## Flash Controller Bypass

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Row Programming algorithms by controlling the Flash programming signals directly.

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

For more information on bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore!<sup>®</sup> MCU Application Note (AN0117)* available for download at [www.zilog.com](http://www.zilog.com).

## Flash Controller Behavior in DEBUG Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored.
- The Flash Sector Protect register is ignored for programming and erase operations.
- Programming operations are not limited to the page selected in the Page Select register.
- Bits in the Flash Sector Protect register can be written to one or zero.
- The second write of the Page Select register to unlock the Flash Controller is not necessary.
- The Page Select register can be written when the Flash Controller is unlocked.
- The Mass Erase command is enabled through the Flash Control register.



**Caution:** *For security reasons, the Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the flash controller must go through the unlock sequence again to select another page.*

## Flash Control Register Definitions

### Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FCTL) register before programming or erasing the Flash memory. Writing the sequence 73H 8CH, sequentially, to the Flash Control register unlocks the Flash Controller. When the Flash Controller is unlocked, the 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 78. Flash Control Register (FCTL)**

BITS	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
ADDR	FF8H							

FCMD—Flash Command

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

**Table 80. Flash Page Select Register (FPS)**

BITS	7	6	5	4	3	2	1	0
FIELD	INFO_EN	PAGE						
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
ADDR	FF9H							

INFO\_EN—Information Area Enable

0 = Information Area us not selected.

1 = Information Area is selected. The Information Area is mapped into the Program Memory address space at addresses FE00H through FFFFH.

PAGE—Page Select

This 7-bit field identifies the Flash memory page for Page Erase and page unlocking. Program Memory Address[15:9] = PAGE[6:0]. For the Z8F08xx devices, the upper 3 bits must be zero. For the Z8F04xx devices, the upper 4 bits must be zero. For Z8F02xx devices, the upper 5 bits must always be 0. For the Z8F01xx devices, the upper 6 bits must always be 0.

## Flash Sector Protect Register

The Flash Sector Protect (FPROT) register is shared with the Flash Page Select Register. When the [Flash Control Register](#) is written with 73H followed by 5EH, the next write to this address targets the Flash Sector Protect Register. In all other cases, it targets the Flash Page Select Register.

This register selects one of the 8 available Flash memory sectors to be protected. The reset state of each Sector Protect bit is an unprotected state. After a sector is protected by setting its corresponding register bit, it cannot be unprotected (the register bit cannot be cleared) without powering down the device.

**Table 81. Flash Sector Protect Register (FPROT)**

BITS	7	6	5	4	3	2	1	0
FIELD	SPROT7	SPROT6	SPROT5	SPROT4	SPROT3	SPROT2	SPROT1	SPROT0
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
ADDR	FF9H							

**Trim Bit Address 0004H****Table 92. Trim Option Bits at 0004H**

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved							
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
ADDR	Information Page Memory 0024H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Reserved—Altering this register may result in incorrect device operation.

**Zilog Calibration Data****ADC Calibration Data****Table 93. ADC Calibration Bits**

BITS	7	6	5	4	3	2	1	0
FIELD	ADC_CAL							
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
ADDR	Information Page Memory 0060H–007DH							
Note: U = Unchanged by Reset. R/W = Read/Write.								

ADC\_CAL—Analog-to-Digital Converter Calibration Values

Contains factory calibrated values for ADC gain and offset compensation. Each of the ten supported modes has one byte of offset calibration and two bytes of gain calibration.

These values are read by the software to compensate ADC measurements as described in [Software Compensation Procedure Using Factory Calibration Data](#) on page 126. The location of each calibration byte is provided in [Table 94](#) on page 162.

**Table 94. ADC Calibration Data Location (Continued)**

Info Page Address	Memory Address	Compensation Usage	ADC Mode	Reference Type
34	FE34	Negative Gain High Byte	Differential Unbuffered	External 2.0 V
35	FE35	Negative Gain Low Byte	Differential Unbuffered	External 2.0 V
78	FE78	Offset	Differential 1x Buffered	Internal 2.0 V
18	FE18	Positive Gain High Byte	Differential 1x Buffered	Internal 2.0 V
19	FE19	Positive Gain Low Byte	Differential 1x Buffered	Internal 2.0 V
36	FE36	Negative Gain High Byte	Differential 1x Buffered	Internal 2.0 V
37	FE37	Negative Gain Low Byte	Differential 1x Buffered	Internal 2.0 V
7B	FE7B	Offset	Differential 1x Buffered	External 2.0 V
1A	FE1A	Positive Gain High Byte	Differential 1x Buffered	External 2.0 V
1B	FE1B	Positive Gain Low Byte	Differential 1x Buffered	External 2.0 V
38	FE38	Negative Gain High Byte	Differential 1x Buffered	External 2.0 V
39	FE39	Negative Gain Low Byte	Differential 1x Buffered	External 2.0 V

**Table 102. Randomized Lot ID Locations (Continued)**

Info Page Address	Memory Address	Usage
5C	FE5C	Randomized Lot ID Byte 23
5D	FE5D	Randomized Lot ID Byte 22
5E	FE5E	Randomized Lot ID Byte 21
5F	FE5F	Randomized Lot ID Byte 20
61	FE61	Randomized Lot ID Byte 19
62	FE62	Randomized Lot ID Byte 18
64	FE64	Randomized Lot ID Byte 17
65	FE65	Randomized Lot ID Byte 16
67	FE67	Randomized Lot ID Byte 15
68	FE68	Randomized Lot ID Byte 14
6A	FE6A	Randomized Lot ID Byte 13
6B	FE6B	Randomized Lot ID Byte 12
6D	FE6D	Randomized Lot ID Byte 11
6E	FE6E	Randomized Lot ID Byte 10
70	FE70	Randomized Lot ID Byte 9
71	FE71	Randomized Lot ID Byte 8
73	FE73	Randomized Lot ID Byte 7
74	FE74	Randomized Lot ID Byte 6
76	FE76	Randomized Lot ID Byte 5
77	FE77	Randomized Lot ID Byte 4
79	FE79	Randomized Lot ID Byte 3
7A	FE7A	Randomized Lot ID Byte 2
7C	FE7C	Randomized Lot ID Byte 1
7D	FE7D	Randomized Lot ID Byte 0 (least significant)

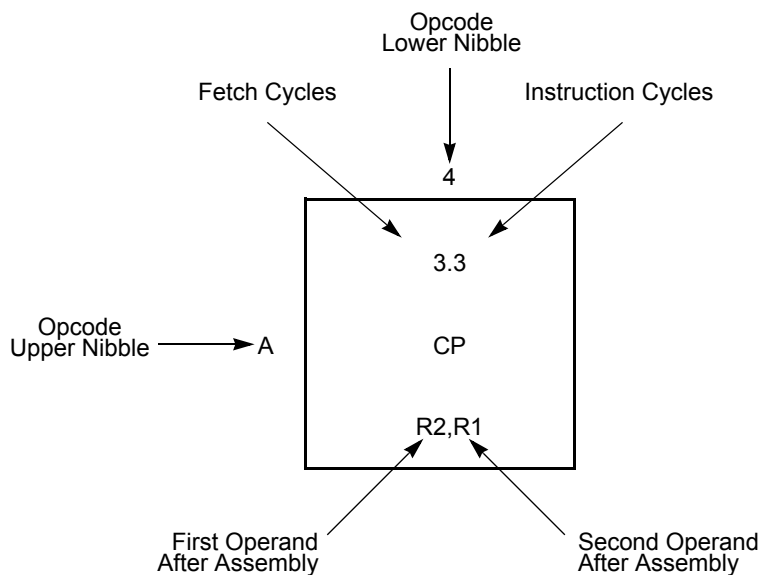
Table 124. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
COM dst	dst ← ~dst	R		60	–	*	*	0	–	–	2	2
		IR		61							2	3
CP dst, src	dst - src	r	r	A2	*	*	*	*	–	–	2	3
		r	lr	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst - src - C	r	r	1F A2	*	*	*	*	–	–	3	3
		r	lr	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst - src - C	ER	ER	1F A8	*	*	*	*	–	–	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst - src	ER	ER	A8	*	*	*	*	–	–	4	3
		ER	IM	A9							4	3
DA dst	dst ← DA(dst)	R		40	*	*	*	X	–	–	2	2
		IR		41							2	3
DEC dst	dst ← dst - 1	R		30	–	*	*	*	–	–	2	2
		IR		31							2	3
DECW dst	dst ← dst - 1	RR		80	–	*	*	*	–	–	2	5
		IRR		81							2	6
DI	IRQCTL[7] ← 0			8F	–	–	–	–	–	–	1	2
DJNZ dst, RA	dst ← dst - 1 if dst ≠ 0 PC ← PC + X	r		0A-FA	–	–	–	–	–	–	2	3
EI	IRQCTL[7] ← 1			9F	–	–	–	–	–	–	1	2
Flags Notation:	* = Value is a function of the result of the operation. – = Unaffected X = Undefined				0 = Reset to 0 1 = Set to 1							



## Opcode Maps

A description of the opcode map data and the abbreviations are provided in [Figure 30](#). [Figure 31](#) and [Figure 32](#) displays the eZ8 CPU instructions. [Table 125](#) lists Opcode Map abbreviations.



**Figure 30. Opcode Map Cell Description**

Figure 46 displays the 28-pin Small Outline Integrated Circuit package (SOIC) available in the Z8 Encore! XP F082A Series devices.

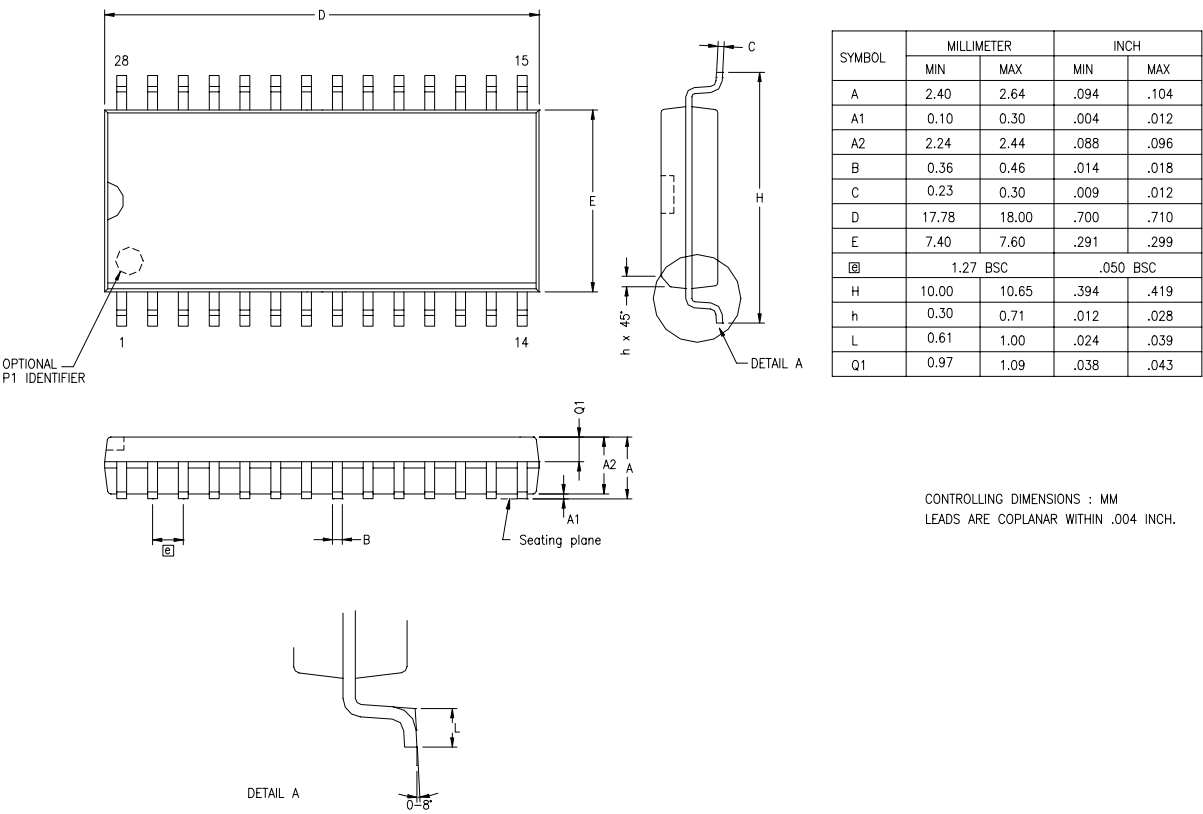


Figure 46. 28-Pin Small Outline Integrated Circuit Package (SOIC)

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
<b>Z8 Encore! XP<sup>®</sup> F082A Series with 8 KB Flash</b>											
<b>Standard Temperature: 0 °C to 70 °C</b>											
Z8F081APB020SC	8 KB	1 KB	0	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F081AQB020SC	8 KB	1 KB	0	6	13	2	0	1	1	0	QFN 8-pin package
Z8F081ASB020SC	8 KB	1 KB	0	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F081ASH020SC	8 KB	1 KB	0	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F081AHH020SC	8 KB	1 KB	0	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F081APH020SC	8 KB	1 KB	0	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F081ASJ020SC	8 KB	1 KB	0	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F081AHJ020SC	8 KB	1 KB	0	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F081APJ020SC	8 KB	1 KB	0	25	19	2	0	1	1	0	PDIP 28-pin package
<b>Extended Temperature: -40 °C to 105 °C</b>											
Z8F081APB020EC	8 KB	1 KB	0	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F081AQB020EC	8 KB	1 KB	0	6	13	2	0	1	1	0	QFN 8-pin package
Z8F081ASB020EC	8 KB	1 KB	0	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F081ASH020EC	8 KB	1 KB	0	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F081AHH020EC	8 KB	1 KB	0	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F081APH020EC	8 KB	1 KB	0	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F081ASJ020EC	8 KB	1 KB	0	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F081AHJ020EC	8 KB	1 KB	0	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F081APJ020EC	8 KB	1 KB	0	25	19	2	0	1	1	0	PDIP 28-pin package
Replace C with G for Lead-Free Packaging											

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
<b>Z8 Encore! XP<sup>®</sup> F082A Series with 1 KB Flash</b>											
<b>Standard Temperature: 0 °C to 70 °C</b>											
Z8F011APB020SC	1 KB	256 B	16 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F011AQB020SC	1 KB	256 B	16 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F011ASB020SC	1 KB	256 B	16 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F011ASH020SC	1 KB	256 B	16 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F011AHH020SC	1 KB	256 B	16 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F011APH020SC	1 KB	256 B	16 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F011ASJ020SC	1 KB	256 B	16 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F011AHJ020SC	1 KB	256 B	16 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F011APJ020SC	1 KB	256 B	16 B	25	19	2	0	1	1	0	PDIP 28-pin package
<b>Extended Temperature: -40 °C to 105 °C</b>											
Z8F011APB020EC	1 KB	256 B	16 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F011AQB020EC	1 KB	256 B	16 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F011ASB020EC	1 KB	256 B	16 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F011ASH020EC	1 KB	256 B	16 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F011AHH020EC	1 KB	256 B	16 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F011APH020EC	1 KB	256 B	16 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F011ASJ020EC	1 KB	256 B	16 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F011AHJ020EC	1 KB	256 B	16 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F011APJ020EC	1 KB	256 B	16 B	25	19	2	0	1	1	0	PDIP 28-pin package
Replace C with G for Lead-Free Packaging											

register 201

- ADC control (ADCCTL) 130, 132
- ADC data high byte (ADCDH) 132
- ADC data low bits (ADC DL) 133
- flash control (FCTL) 149, 155, 156
- flash high and low byte (FFREQH and FREEQL) 152
- flash page select (FPS) 150, 151
- flash status (FSTAT) 150
- GPIO port A-H address (PxADDR) 46
- GPIO port A-H alternate function sub-registers 48
- GPIO port A-H control address (PxCTL) 47
- GPIO port A-H data direction sub-registers 47
- OCD control 184
- OCD status 185
- UARTx baud rate high byte (UxBRH) 114
- UARTx baud rate low byte (UxBRL) 114
- UARTx Control 0 (UxCTL0) 108, 114
- UARTx control 1 (UxCTL1) 109
- UARTx receive data (UxRXD) 113
- UARTx status 0 (UxSTAT0) 111
- UARTx status 1 (UxSTAT1) 112
- UARTx transmit data (UxTXD) 113
- Watchdog Timer control (WDTCTL) 31, 94, 136, 190
- Watchdog Timer reload high byte (WDTH) 95
- Watchdog Timer reload low byte (WDTL) 95
- Watchdog Timer reload upper byte (WD-TU) 95

register file 15

register pair 201

register pointer 202

reset

- and stop mode characteristics 24
- and Stop Mode Recovery 23
- carry flag 204
- sources 25

RET 206

return 206

RL 206

RLC 206

rotate and shift instructions 206

rotate left 206

rotate left through carry 206

rotate right 206

rotate right through carry 206

RP 202

RR 201, 206

rr 201

RRC 206

## S

SBC 203

SCF 204, 205

second opcode map after 1FH 219

set carry flag 204, 205

set register pointer 205

shift right arithmetic 207

shift right logical 207

signal descriptions 11

single-shot conversion (ADC) 123

software trap 206

source operand 202

SP 202

SRA 207

src 202

SRL 207

SRP 205

stack pointer 202

STOP 205

STOP mode 33

stop mode 205

Stop Mode Recovery

- sources 28

- using a GPIO port pin transition 29

- using Watchdog Timer time-out 29

stop mode recovery

- sources 30

- using a GPIO port pin transition 30

SUB 203

subtract 203

subtract - extended addressing 203

subtract with carry 203