



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### 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	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f041aph020sc

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



## **Table of Contents**

Overview
Features
Part Selection Guide
Block Diagram
CPU and Peripheral Overview
eZ8 CPU Features 5
10-Bit Analog-to-Digital Converter 5
Low-Power Operational Amplifier
Internal Precision Oscillator
Apalog Comparator
External Crystal Oscillator
Low Voltage Detector
On-Chip Debugger
Universal Asynchronous Receiver/Transmitter
Timers
General-Purpose Input/Output
Elash Controller
Non-Volatile Data Storage 7
Interrupt Controller
Reset Controller
Pin Description
Available Packages 9
Pin Configurations 9
Signal Descriptions
Pin Characteristics
Addross Space
Register File
Program Memory
Data Memory
Flash Information Area 17
Register Map



## **Signal Descriptions**

Table 2 describes the Z8 Encore! XP F082A Series signals. See Pin Configurations on page 9 to determine the signals available for the specific package styles.

Signal Mnemonic	I/O	Description
General-Purpose I/C	) Ports	A–D
PA[7:0]	I/O	Port A. These pins are used for general-purpose I/O.
PB[7:0]	I/O	Port B. These pins are used for general-purpose I/O. PB6 and PB7 are available only in those devices without an ADC.
PC[7:0]	I/O	Port C. These pins are used for general-purpose I/O.
PD[0]	I/O	Port D. This pin is used for general-purpose output only.
<b>Note:</b> PB6 and PB7 are replaced by AV <sub>DI</sub>	e only av <sub>D</sub> and A <sup>v</sup>	vailable in 28-pin packages without ADC. In 28-pin packages with ADC, they are $V_{\rm SS}$ .
UART Controllers		
TXD0	0	Transmit Data. This signal is the transmit output from the UART and IrDA.
RXD0	I	Receive Data. This signal is the receive input for the UART and IrDA.
CTS0	I	Clear To Send. This signal is the flow control input for the UART.
DE	0	Driver Enable. This signal allows automatic control of external RS-485 drivers. This signal is approximately the inverse of the TXE (Transmit Empty) bit in the UART Status 0 register. The DE signal may be used to ensure the external RS-485 driver is enabled when data is transmitted by the UART.
Timers		
T0OUT/T1OUT	0	Timer Output 0–1. These signals are outputs from the timers.
T0OUT/T1OUT	0	Timer Complement Output 0–1. These signals are output from the timers in PWM Dual Output mode.
T0IN/T1IN	Ι	Timer Input 0–1. These signals are used as the capture, gating and counter inputs.
Comparator		
CINP/CINN	Ι	Comparator Inputs. These signals are the positive and negative inputs to the comparator.
COUT	0	Comparator Output.

#### Table 2. Signal Descriptions

**z**ilog<sup>°</sup>

addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 5 describes the Program Memory Maps for the Z8 Encore! XP F082A Series products.

Program Memory Address (Hex)	Function				
Z8F082A and Z8F081A Products					
0000–0001	Flash Option Bits				
0002–0003	Reset Vector				
0004–0005	WDT Interrupt Vector				
0006–0007	Illegal Instruction Trap				
0008–0037	Interrupt Vectors*				
0038–0039	Reserved				
003A–003D	Oscillator Fail Trap Vectors				
003E–1FFF	Program Memory				
Z8F042A and Z8F041A Products					
0000–0001	Flash Option Bits				
0002–0003	Reset Vector				
0004–0005	WDT Interrupt Vector				
0006–0007	Illegal Instruction Trap				
0008–0037	Interrupt Vectors*				
0038–0039	Reserved				
003A-003D	Oscillator Fail Trap Vectors				
003E-0FFF	Program Memory				

 Table 5. Z8 Encore! XP F082A Series Program Memory Maps

16

zilog /

tions as a GPIO pin. If it is not present, the debug feature is disabled until/unless another reset event occurs. For more details, see On-Chip Debugger on page 173.

## **Crystal Oscillator Override**

For systems using a crystal oscillator, PA0 and PA1 are used to connect the crystal. When the crystal oscillator is enabled (see Oscillator Control Register Definitions on page 190), the GPIO settings are overridden and PA0 and PA1 are disabled.

## **5 V Tolerance**

All six I/O pins on the 8-pin devices are 5 V-tolerant, unless the programmable pull-ups are enabled. If the pull-ups are enabled and inputs higher than  $V_{DD}$  are applied to these parts, excessive current flows through those pull-up devices and can damage the chip.

**Note:** In the 20- and 28-pin versions of this device, any pin which shares functionality with an ADC, crystal or comparator port is not 5 V-tolerant, including PA[1:0], PB[5:0] and PC[2:0]. All other signal pins are 5 V-tolerant, and can safely handle inputs higher than  $V_{DD}$  except when the programmable pull-ups are enabled.

## **External Clock Setup**

For systems using an external TTL drive, PB3 is the clock source for 20- and 28-pin devices. In this case, configure PB3 for alternate function CLKIN. Write the Oscillator Control (OSCCTL) register (see Oscillator Control Register Definitions on page 190) such that the external oscillator is selected as the system clock. For 8-pin devices use PA1 instead of PB3.



#### Table 18. Port A–D Control Registers (PxCTL)

BITS	7	6	5	4	3	2	1	0	
FIELD		PCTL							
RESET		00H							
R/W	R/W	R/W R/W R/W R/W R/W R/W R/W							
ADDR			FI	D1H, FD5H,	FD9H, FDD	H			

PCTL[7:0]—Port Control

The Port Control register provides access to all sub-registers that configure the GPIO Port operation.

## Port A–D Data Direction Sub-Registers

The Port A–D Data Direction sub-register is accessed through the Port A–D Control register by writing 01H to the Port A–D Address register (Table 19).

BITS	7	6	5	4	3	2	1	0
FIELD	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	lf 01H i	n Port A–D /	Address Reg	gister, acces	sible throug	n the Port A-	-D Control F	Register

#### Table 19. Port A–D Data Direction Sub-Registers (PxDD)

#### DD[7:0]—Data Direction

These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction register setting.

0 = Output. Data in the Port A–D Output Data register is driven onto the port pin.

1 = Input. The port pin is sampled and the value written into the Port A–D Input Data Register. The output driver is tristated.

## Port A–D Alternate Function Sub-Registers

The Port A–D Alternate Function sub-register (Table 20) is accessed through the Port A–D Control register by writing 02H to the Port A–D Address register. The Port A–D Alternate Function sub-registers enable the alternate function selection on pins. If disabled, pins functions as GPIO. If enabled, select one of four alternate functions using alternate function set subregisters 1 and 2 as described in the Port A–D Alternate Function

zilog<sup>\*</sup>

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.

Follow the steps below for configuring a timer for PWM SINGLE OUTPUT mode and initiating the PWM operation:

- 1. Write to the Timer Control register to:
  - Disable the timer.
  - Configure the timer for PWM SINGLE OUTPUT mode.
  - Set the prescale value.
  - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function.
- 2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0001H.
- 3. Write to the PWM High and Low Byte registers to set the PWM value.
- 4. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
- 5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 6. Configure the associated GPIO port pin for the Timer Output alternate function.
- 7. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

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

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 is set to 0, the ratio of the PWM output High time to the total period is represented by:

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

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

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







### Watchdog Timer Refresh

When first enabled, 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! XP<sup>®</sup> F082A Series devices are operating in DEBUG mode (using the on-chip debugger), the Watchdog Timer is continuously refreshed to prevent any Watchdog Timer 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. For information on programming the WDT\_RES Flash Option Bit, see Flash Option Bits on page 153.

#### 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 (RSTSTAT) register (see Reset Status Register on page 30). 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 rolls over to its maximum value of FFFFFH and continues counting. The Watchdog Timer counter is not automatically returned to its Reload Value.

The Reset Status (RSTSTAT) register must be read before clearing the WDT interrupt. This read clears the WDT timeout Flag and prevents further WDT interrupts from immediately occurring.

#### WDT Interrupt in STOP Mode

If configured to generate an interrupt when a time-out occurs and the Z8 Encore! XP F082A 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 Reset Status (RSTSTAT) register are set to 1 following a WDT time-out in STOP mode. For more information on Stop Mode Recovery, see Reset, Stop Mode Recovery, and Low Voltage Detection on page 23.

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 executing code from the vector address.

92



**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/* - Rea	PAW* - Read returns the current WDT count value, Write sets the appropriate Reload Value								

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* - Rea	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* - Rea	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.



MPRX—Multiprocessor Receive

Returns the value of the most recent multiprocessor bit received. Reading from the UART Receive Data register resets this bit to 0.

### **UART Transmit Data Register**

Data bytes written to the UART Transmit Data (UxTXD) register (Table 65) are shifted out on the TXDx pin. The Write-only UART Transmit Data register shares a Register File address with the read-only UART Receive Data register.

BITS	7	6	5	4	3	2	1	0
FIELD				Tک	(D			
RESET	Х	Х	Х	Х	Х	Х	Х	Х
R/W	W	W	W	W	W	W	W	W
ADDR				F4	0H			

#### Table 65. UART Transmit Data Register (U0TXD)

TXD-Transmit Data

UART transmitter data byte to be shifted out through the TXDx pin.

#### **UART Receive Data Register**

Data bytes received through the RXDx pin are stored in the UART Receive Data (UxRXD) register (Table 66). The read-only UART Receive Data register shares a Register File address with the Write-only UART Transmit Data register.

#### Table 66. UART Receive Data Register (U0RXD)

BITS	7	6	5	4	3	2	1	0	
FIELD	RXD								
RESET	Х	Х	Х	Х	Х	Х	Х	Х	
R/W	R	R	R	R	R	R	R	R	
ADDR	F40H								
X = Undef	K = Undefined.								

RXD—Receive Data

UART receiver data byte from the RXDx pin

Zilog

#### **Receiving IrDA Data**

Data received from the infrared transceiver using the IR\_RXD signal through the RXD pin is decoded by the Infrared Endec and passed to the UART. The UART's baud rate clock is used by the Infrared Endec to generate the demodulated signal (RXD) that drives the UART. Each UART/Infrared data bit is 16-clocks wide. Figure 18 displays data reception. When the Infrared Endec is enabled, the UART's RXD signal is internal to the Z8 Encore! XP<sup>®</sup> F082A Series products while the IR\_RXD signal is received through the RXD pin.



Figure 18. IrDA Data Reception

#### **Infrared Data Reception**

**Caution:** The system clock frequency must be at least 1.0 MHz to ensure proper reception of the  $1.4 \,\mu s$  minimum width pulses allowed by the IrDA standard.

#### **Endec Receiver Synchronization**

The IrDA receiver uses a local baud rate clock counter (0 to 15 clock periods) to generate an input stream for the UART and to create a sampling window for detection of incoming pulses. The generated UART input (UART RXD) is delayed by 8 baud rate clock periods with respect to the incoming IrDA data stream. When a falling edge in the input data stream is detected, the Endec counter is reset. When the count reaches a value of 8, the UART RXD value is updated to reflect the value of the decoded data. When the count reaches 12 baud clock periods, the sampling window for the next incoming pulse opens. The window remains open until the count again reaches 8 (that is, 24 baud clock periods since the previous pulse was detected), giving the Endec a sampling window of minus four

zilog <sub>123</sub>

can output values across the entire 11-bit range, from -1024 to +1023. In SINGLE-ENDED mode, the output generally ranges from 0 to +1023, but offset errors can cause small negative values.

The ADC registers actually return 13 bits of data, but the two LSBs are intended for compensation use only. When the software compensation routine is performed on the 13 bit raw ADC value, two bits of resolution are lost because of a rounding error. As a result, the final value is an 11-bit number.

#### Hardware Overflow

When the hardware overflow bit (OVF) is set in ADC Data Low Byte (ADCD\_L) register, all other data bits are invalid. The hardware overflow bit is set for values greater than  $V_{ref}$  and less than  $-V_{ref}$  (DIFFERENTIAL mode).

#### **Automatic Powerdown**

If the ADC is idle (no conversions in progress) for 160 consecutive system clock cycles, portions of the ADC are automatically powered down. From this powerdown state, the ADC requires 40 system clock cycles to power up. The ADC powers up when a conversion is requested by the ADC Control register.

#### Single-Shot Conversion

When configured for single-shot conversion, the ADC performs a single analog-to-digital conversion on the selected analog input channel. After completion of the conversion, the ADC shuts down. Follow the steps below for setting up the ADC and initiating a single-shot conversion:

- 1. Enable the desired analog inputs by configuring the general-purpose I/O pins for alternate analog function. This configuration disables the digital input and output drivers.
- 2. Write the ADC Control/Status Register 1 to configure the ADC.
  - Write to BUFMODE [2:0] to select SINGLE-ENDED or DIFFERENTIAL mode, as well as unbuffered or buffered mode.
  - Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is. contained in the ADC Control Register 0.
- 3. Write to the ADC Control Register 0 to configure the ADC and begin the conversion. The bit fields in the ADC Control register can be written simultaneously (the ADC can be configured and enabled with the same write instruction):
  - Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
  - Clear CONT to 0 to select a single-shot conversion.

zilog 1

- If the internal voltage reference must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
- Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in the ADC Control/Status Register 1.
- Set CEN to 1 to start the conversion.
- 4. CEN remains 1 while the conversion is in progress. A single-shot conversion requires 5129 system clock cycles to complete. If a single-shot conversion is requested from an ADC powered-down state, the ADC uses 40 additional clock cycles to power up before beginning the 5129 cycle conversion.
- 5. When the conversion is complete, the ADC control logic performs the following operations:
  - 13-bit two's-complement result written to {ADCD\_H[7:0], ADCD\_L[7:3]}.
  - Sends an interrupt request to the Interrupt Controller denoting conversion complete.
  - CEN resets to 0 to indicate the conversion is complete.
- 6. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

#### **Continuous Conversion**

When configured for continuous conversion, the ADC continuously performs an analog-to-digital conversion on the selected analog input. Each new data value over-writes the previous value stored in the ADC Data registers. An interrupt is generated after each conversion.

∕i∕ c

**Caution:** In CONTINUOUS mode, ADC updates are limited by the input signal bandwidth of the ADC and the latency of the ADC and its digital filter. Step changes at the input are not immediately detected at the next output from the ADC. The response of the ADC (in all modes) is limited by the input signal bandwidth and the latency.

Follow the steps below for setting up the ADC and initiating continuous conversion:

- 1. Enable the desired analog input by configuring the general-purpose I/O pins for alternate function. This action disables the digital input and output driver.
- 2. Write the ADC Control/Status Register 1 to configure the ADC.
  - Write to BUFMODE[2:0] to select SINGLE-ENDED or DIFFERENTIAL mode, as well as unbuffered or buffered mode.
  - Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in the ADC Control Register 0.

# zilog

## **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. I	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

zilog<sup>®</sup>

1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

OSC SEL[1:0]—Oscillator Mode Selection

00 = On-chip oscillator configured for use with external RC networks (<4 MHz).

01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).

10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 5.0 MHz).

11 = Maximum power for use with high frequency crystals (5.0 MHz to 20.0 MHz). This setting is the default for unprogrammed (erased) Flash.

VBO AO-Voltage Brownout Protection Always On

0 = Voltage Brownout Protection can be disabled in STOP mode to reduce total power consumption. For the block to be disabled, the power control register bit must also be written (see Power Control Register Definitions on page 34).

1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

FRP—Flash Read Protect

0 = User program code is inaccessible. Limited control features are available through the On-Chip Debugger.

1 = User program code is accessible. All On-Chip Debugger commands are enabled. This setting is the default for unprogrammed (erased) Flash.

Reserved—Must be 1.

FWP—Flash Write Protect

This Option Bit provides Flash Program Memory protection:

0 = Programming and erasure disabled for all of Flash Program Memory. Programming,

Page Erase, and Mass Erase through User Code is disabled. Mass Erase is available using the On-Chip Debugger.

1 = Programming, Page Erase, and Mass Erase are enabled for all of Flash program memory.

# zilog

180

Debug Command	Command Byte	Enabled when NOT in DEBUG mode?	Disabled by Flash Read Protect Option Bit
Write Register	08H	_	Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control register.
Read Register	09H	_	Disabled
Write Program Memory	0AH	_	Disabled
Read Program Memory	0BH	_	Disabled
Write Data Memory	0CH	_	Yes
Read Data Memory	0DH	_	-
Read Program Memory CRC	0EH	_	-
Reserved	0FH	_	-
Step Instruction	10H	_	Disabled
Stuff Instruction	11H	_	Disabled
Execute Instruction	12H	_	Disabled
Reserved	13H–FFH	_	_

In the following bulleted list of OCD Commands, data and commands sent from the host to the On-Chip Debugger are identified by 'DBG  $\leftarrow$  Command/Data'. Data sent from the On-Chip Debugger back to the host is identified by 'DBG  $\rightarrow$  Data'

• **Read OCD Revision (00H)**—The Read OCD Revision command determines the version of the On-Chip Debugger. If OCD commands are added, removed, or changed, this revision number changes.

```
DBG \leftarrow 00H
DBG \rightarrow OCDRev[15:8] (Major revision number)
DBG \rightarrow OCDRev[7:0] (Minor revision number)
```

• **Read OCD Status Register (02H)**—The Read OCD Status Register command reads the OCDSTAT register.

```
DBG \leftarrow 02H
DBG \rightarrow OCDSTAT[7:0]
```

• **Read Runtime Counter (03H)**—The Runtime Counter counts system clock cycles in between Breakpoints. The 16-bit Runtime Counter counts up from 0000H and stops at the maximum count of FFFFH. The Runtime Counter is overwritten during the

zilog

189

When selecting a new clock source, the system clock oscillator failure detection circuitry and the Watchdog Timer oscillator failure circuitry must be disabled. If SOFEN and WOFEN are not disabled prior to a clock switch-over, it is possible to generate an interrupt for a failure of either oscillator. The Failure detection circuitry can be enabled anytime after a successful write of OSCSEL in the OSCCTL register.

The internal precision oscillator is enabled by default. If the user code changes to a different oscillator, it may be appropriate to disable the IPO for power savings. Disabling the IPO does not occur automatically.

#### **Clock Failure Detection and Recovery**

#### System Clock Oscillator Failure

The Z8F04xA family devices can generate non-maskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer oscillator to drive the system clock. The Watchdog Timer oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function (see Watchdog Timer on page 91).

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below 1 kHz  $\pm$ 50%. If an external signal is selected as the system oscillator, it is possible that a very slow but non-failing clock can generate a failure condition. Under these conditions, do not enable the clock failure circuitry (SOFEN must be deasserted in the OSCCTL register).

#### Watchdog Timer Failure

In the event of a Watchdog Timer oscillator failure, a similar non-maskable interrupt-like event is issued. This event does not trigger an attendant clock switch-over, but alerts the CPU of the failure. After a Watchdog Timer failure, it is no longer possible to detect a primary oscillator failure. The failure detection circuitry does not function if the Watchdog Timer is used as the system clock oscillator or if the Watchdog Timer oscillator has been disabled. For either of these cases, it is necessary to disable the detection circuitry by deasserting the WDFEN bit of the OSCCTL register.

The Watchdog Timer oscillator failure detection circuit counts system clocks while looking for a Watchdog Timer clock. The logic counts 8004 system clock cycles before determining that a failure has occurred. The system clock rate determines the speed at which the Watchdog Timer failure can be detected. A very slow system clock results in very slow detection times.

# zilog

#### 199

## eZ8 CPU Instruction Set

## **Assembly Language Programming Introduction**

The eZ8 CPU assembly language provides a means for writing an application program without concern for actual memory addresses or machine instruction formats. A program written in assembly language is called a source program. Assembly language allows the use of symbolic addresses to identify memory locations. It also allows mnemonic codes (opcodes and operands) to represent the instructions themselves. The opcodes identify the instruction while the operands represent memory locations, registers, or immediate data values.

Each assembly language program consists of a series of symbolic commands called statements. Each statement can contain labels, operations, operands and comments.

Labels can be assigned to a particular instruction step in a source program. The label identifies that step in the program as an entry point for use by other instructions.

The assembly language also includes assembler directives that supplement the machine instruction. The assembler directives, or pseudo-ops, are not translated into a machine instruction. Rather, the pseudo-ops are interpreted as directives that control or assist the assembly process.

The source program is processed (assembled) by the assembler to obtain a machine language program called the object code. The object code is executed by the eZ8 CPU. An example segment of an assembly language program is detailed in the following example.

#### Assembly Language Source Program Example

JP START	; Everything after the semicolon is a comment.	
START:	; A label called 'START'. The first instruction (JP START) in this ; example causes program execution to jump to the point within the ; program where the START label occurs.	
LD R4, R7	; A Load (LD) instruction with two operands. The first operand, ; Working Register R4, is the destination. The second operand, ; Working Register R7, is the source. The contents of R7 is ; written into R4.	
LD 234H, #%01	; Another Load (LD) instruction with two operands. ; The first operand, Extended Mode Register Address 234H, ; identifies the destination. The second operand, Immediate Data	
	; value 01H, is the source. The value 01H is written into the ; Register at address 234H.	

INCH

MAX

0.068

0.010

0.061

0.019

0.010

0.196

0.157

0.242

0.016

0.032

.050 BSC

MIN

0.061

0.004

0.055

0.014

0.007

0.189

0.150

0.230

0.010

0.018

MAX

1.73

0.25

1.55

0.48

0.25

4.98

3.99

6.15

0.40

0.81

zilog 242



## Figure 40 displays the 8-pin Small Outline Integrated Circuit package (SOIC) available for the Z8 Encore! XP<sup>®</sup> F082A Series devices.

CONTROLLING DIMENSIONS : MM LEADS ARE COPLANAR WITHIN .004 INCH.



Figure 40. 8-Pin Small Outline Integrated Circuit Package (SOIC)



264

**DJNZ 206** EI 204 **HALT 204 INC 203 INCW 203 IRET 206** JP 206 LD 205 LDC 205 LDCI 204, 205 LDE 205 LDEI 204 LDX 205 LEA 205 logical 205 **MULT 203** NOP 204 OR 205 **ORX 206** POP 205 **POPX 205** program control 206 **PUSH 205** PUSHX 205 **RCF 204 RET 206** RL 206 **RLC 206** rotate and shift 206 RR 206 **RRC 206** SBC 203 SCF 204, 205 SRA 207 SRL 207 **SRP 205 STOP 205** SUB 203 SUBX 203 **SWAP 207** TCM 204 **TCMX 204** TM 204 TMX 204

**TRAP 206** Watchdog Timer refresh 205 XOR 206 **XORX 206** instructions, eZ8 classes of 202 interrupt control register 67 interrupt controller 55 architecture 55 interrupt assertion types 58 interrupt vectors and priority 58 operation 57 register definitions 60 software interrupt assertion 59 interrupt edge select register 66 interrupt request 0 register 60 interrupt request 1 register 61 interrupt request 2 register 62 interrupt return 206 interrupt vector listing 55 interrupts **UART 105** IR 201 lr 201 IrDA architecture 117 block diagram 117 control register definitions 120 operation 117 receiving data 119 transmitting data 118 **IRET 206** IRQ0 enable high and low bit registers 62 IRQ1 enable high and low bit registers 63 IRQ2 enable high and low bit registers 65 **IRR 201** Irr 201

## J

JP 206 jump, conditional, relative, and relative conditional 206