



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, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	16 × 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f012aph020sg

Email: info@E-XFL.COM

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

## Z8 Encore! XP<sup>®</sup> F082A Series Product Specification

Table 29.	Port A–C Input Data Registers (PxIN)
Table 30.	Port A–D Output Data Register (PxOUT)
Table 31.	LED Drive Enable (LEDEN)
Table 32.	LED Drive Level High Register (LEDLVLH)
Table 33.	LED Drive Level Low Register (LEDLVLL)
Table 34.	Trap and Interrupt Vectors in Order of Priority
Table 35.	Interrupt Request 0 Register (IRQ0)
Table 36.	Interrupt Request 1 Register (IRQ1)
Table 37.	Interrupt Request 2 Register (IRQ2)
Table 38.	IRQ0 Enable and Priority Encoding
Table 39.	IRQ0 Enable High Bit Register (IRQ0ENH)
Table 40.	IRQ0 Enable Low Bit Register (IRQ0ENL)
Table 41.	IRQ1 Enable and Priority Encoding
Table 42.	IRQ1 Enable Low Bit Register (IRQ1ENL)
Table 43.	IRQ1 Enable High Bit Register (IRQ1ENH)
Table 44.	IRQ2 Enable and Priority Encoding
Table 45.	IRQ2 Enable High Bit Register (IRQ2ENH)
Table 46.	Interrupt Edge Select Register (IRQES)
Table 47.	IRQ2 Enable Low Bit Register (IRQ2ENL)
Table 48.	Shared Interrupt Select Register (IRQSS)
Table 49.	Interrupt Control Register (IRQCTL)
Table 50.	Timer 0–1 Control Register 0 (TxCTL0)85
Table 51.	Timer 0–1 Control Register 1 (TxCTL1) 86
Table 52.	Timer 0–1 High Byte Register (TxH) 90
Table 53.	Timer 0–1 Low Byte Register (TxL)90
Table 54.	Timer 0–1 Reload High Byte Register (TxRH)
Table 55.	Timer 0–1 Reload Low Byte Register (TxRL)
Table 56.	Timer 0–1 PWM High Byte Register (TxPWMH) 92
Table 57.	Timer 0–1 PWM Low Byte Register (TxPWML)
Table 58.	Watchdog Timer Approximate Time-Out Delays

operate in ONE-SHOT, CONTINUOUS, GATED, CAPTURE, CAPTURE RESTART, COMPARE, CAPTURE and COMPARE, PWM SINGLE OUTPUT and PWM DUAL OUTPUT modes.

#### **General-Purpose Input/Output**

The Product Line MCUs feature 6 to 25 port pins (Ports A–D) for general- purpose input/ output (GPIO). The number of GPIO pins available is a function of package and each pin is individually programmable. 5 V tolerant input pins are available on all I/Os on 8-pin devices and most I/Os on other package types.

#### **Direct LED Drive**

The 20- and 28-pin devices support controlled current sinking output pins capable of driving LEDs without the need for a current limiting resistor. These LED drivers are independently programmable to four different intensity levels.

#### **Flash Controller**

The Flash Controller programs and erases Flash memory. The Flash Controller supports several protection mechanisms against accidental program and erasure, plus factory serialization and read protection.

#### Non-Volatile Data Storage

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

**Note:** Devices with 8KB of Flash memory do not include the NVDS feature.

#### Interrupt Controller

The Z8 Encore! XP F082A Series products support up to 20 interrupts. These interrupts consist of 8 internal peripheral interrupts and 12 general-purpose I/O pin interrupt sources. The interrupts have three levels of programmable interrupt priority.

#### **Reset Controller**

The Z8 Encore! XP F082A Series products can be reset using the RESET pin, Power-On Reset, Watchdog Timer (WDT) time-out, STOP Mode exit, or Voltage Brown-Out (VBO)



Figure 2. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 8-Pin SOIC, QFN/MLF-S, or PDIP Package



Figure 3. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 20-Pin SOIC, SSOP or PDIP Package



Figure 4. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 28-Pin SOIC, SSOP or PDIP Package



Figure 6. Voltage Brown-Out Reset Operation

The POR level is greater than the VBO level by the specified hysteresis value. This ensures that the device undergoes a Power-On Reset after recovering from a VBO condition.

## Watchdog Timer Reset

If the device is operating in NORMAL or HALT Mode, the Watchdog Timer can initiate a System Reset at time-out if the WDT\_RES Flash option bit is programmed to 1, i.e., the unprogrammed state of the WDT\_RES Flash option bit. If the bit is programmed to 0, it configures the Watchdog Timer to cause an interrupt, not a System Reset, at time-out.

The WDT bit in the Reset Status (RSTSTAT) Register is set to signify that the reset was initiated by the Watchdog Timer.

## **External Reset Input**

The  $\overline{\text{RESET}}$  pin has a Schmitt-Triggered input and an internal pull-up resistor. Once the  $\overline{\text{RESET}}$  pin is asserted for a minimum of four system clock cycles, the device progresses through the System Reset sequence. Because of the possible asynchronicity of the system clock and reset signals, the required reset duration may be as short as three clock periods

## Port A–C Input Data Registers

Reading from the Port A–C Input Data registers, shown in Table 29, return the sampled values from the corresponding port pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8-and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	Х	Х	Х	Х	Х	Х	Х	Х
R/W	R	R	R	R	R	R	R	R
Address	FD2H, FD6H, FDAH							
X = Undef	ined.							

Table 29. Port A–C Input Data Registers (PxIN)

Bit	Description
[7:0]	Port Input Data
PxIN	Sampled data from the corresponding port pin input.
	0 = Input data is logical 0 (Low).
	1 = Input data is logical 1 (High).

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.

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			F	D3H, FD7H,	FDBH, FDF	Ή		

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

#### Bit Description

[7:0] **Port Output Data** PxOUT 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. 0 = Drive a logical 0 (Low).

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.

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

## **LED Drive Enable Register**

The LED Drive Enable Register, shown in Table 31, activates the controlled current drive. The Alternate Function Register has no control over the LED function; therefore, setting the Alternate Function Register to select the LED function is not required. LEDEN bits [7:0] correspond to Port C bits [7:0], respectively.

Bit	7	6	5	4	3	2	1	0
Field		LEDEN[7:0]						
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				F8	2H			

Table 31		Drive	Enable	
Table ST.	LEV	Dire	Ellable	(LEDEN)

#### Bit Description

[7:0] LED Drive Enable

LEDENx These bits determine which Port C pins are connected to an internal current sink.

0 = Tristate the Port C pin.

1 = Enable controlled current sink on the Port C pin.

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

## LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin, as shown in Table 32. These two bits select between four programmable drive levels. Each pin is individually programmable.

Bit	7	6	5	4	3	2	1	0		
Field		LEDLVLH[7:0]								
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	F83H									
Bit	Descrip	tion								
[7:0] LEDLVLHx	<b>LED Lev</b> {LEDLVL 00 = 3m	<b>/el High Bit</b> .H, LEDLVLL A	_} select one	of four prog	ammable cu	urrent drive le	evels for eac	n Port C pin.		

Tabla '	22 I EI	סעויים ר		Linh	Dogistor	٨
i abie .	32. LEI	JUIIVe	Levei	піціі	Redister	

01 = 7mA 10 = 13mA

11 = 20 mA

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

## **Interrupt Request 2 Register**

The Interrupt Request 2 (IRQ2) Register, shown in Table 37, stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 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 2 Register to determine if any interrupt requests are pending.

Bit	7	6	5	4	3	2	1	0
Field		Rese	erved		PC3I	PC2I	PC1I	PC0I
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		FC6H						

Table 37.	Interrupt	Request 2	Register	(IRQ2)
-----------	-----------	-----------	----------	--------

Bit	Description
[7:4]	Reserved
	These bits are reserved and must be programmed to 0000.
[3:0]	Port C Pin x Interrupt Request
PCxI	0 = No interrupt request is pending for GPIO Port C pin x.
	1 = An interrupt request from GPIO Port C pin $x$ is awaiting service.
Note:	x indicates the specific GPIO Port C pin number (0–3).

## **IRQ0 Enable High and Low Bit Registers**

Table 38 describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers, shown in Tables 39 and 40, form a priority-encoded enabling for interrupts in the Interrupt Request 0 Register.

IRQ	0ENH[ <i>x</i> ]	IRQ0ENL[x]	Priority	Description
	0	0	Disabled	Disabled
	0	1	Level 1	Low
	1	0	Level 2	Medium
	1	1	Level 3	High
Note:	x indicates	register bits 0-7		

Table 38. IRQ0 Enable and Priority Encoding

- 5. Enable the timer interrupt, if appropriate and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL0 Register.
- 6. Configure the associated GPIO port pin for the Timer Input alternate function.
- 7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

Capture Elapsed Time (s) =  $\frac{(Capture Value - Start Value) \times Prescale}{System Clock Frequency (Hz)}$ 

#### **CAPTURE RESTART Mode**

In CAPTURE RESTART Mode, the current timer count value is recorded when the acceptable external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in TxCTL0 Register is set to indicate the timer interrupt is because of 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 reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in TxCTL0 Register is cleared to indicate the timer interrupt is not caused by an input capture event.

Observe the following steps for configuring a timer for CAPTURE RESTART Mode and initiating the count:

- 1. Write to the Timer Control Register to:
  - Disable the timer
  - Configure the timer for CAPTURE RESTART Mode by writing the TMODE bits in the TxCTL1 Register and the TMODEHI bit in TxCTL0 Register
  - Set the prescale value
  - Set the Capture edge (rising or falling) for the Timer Input
- 2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).

- 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, if appropriate 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. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL0 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. Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge.

In CAPTURE/COMPARE Mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

Capture Elapsed Time (s) =  $\frac{(Capture Value - Start Value) \times Prescale}{System Clock Frequency (Hz)}$ 

## **Reading the Timer Count Values**

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte Register is read, the contents of the Timer Low Byte Register are placed in a holding register. A subsequent read from the Timer Low Byte Register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte Register returns the actual value in the counter.

#### **Timer Pin Signal Operation**

The timer output function is a GPIO port pin alternate function. The Timer Output is toggled every time the counter is reloaded.

Bit	7	6	5	4	3	2	1	0			
Field	TH										
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										
Address	F00H, F08H										

#### Table 52. Timer 0–1 High Byte Register (TxH)

#### Table 53. Timer 0–1 Low Byte Register (TxL)

Bit	7	6	5	4	3	2	1	0		
Field	TL									
RESET	0	0	0	0	0	0	1			
R/W	R/W	R/W R/W R/W R/W R/W R/W					R/W			
Address	F01H, F09H									

Bit	Description
[7:0]	Timer High and Low Bytes
TH, TL	These 2 bytes, {TH[7:0], TL[7:0]}, contain the current 16-bit timer count value.

# Universal Asynchronous Receiver/ Transmitter

The universal asynchronous receiver/transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The UART uses a single 8-bit data mode with selectable parity. Features of the UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- Option of one or two STOP bits
- Separate transmit and receive interrupts
- Framing, parity, overrun and break detection
- Separate transmit and receive enables
- 16-bit baud rate generator (BRG)
- Selectable MULTIPROCESSOR (9-bit) Mode with three configurable interrupt schemes
- Baud rate generator (BRG) can be configured and used as a basic 16-bit timer
- Driver enable (DE) output for external bus transceivers

## Architecture

The UART consists of three primary functional blocks: transmitter, receiver and baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. Figure 10 displays the UART architecture.

# Low Power Operational Amplifier

The LPO is a general-purpose low power operational amplifier. Each of the three ports of the amplifier is accessible from the package pins. The LPO contains only one pin configuration: ANA0 is the output/feedback node, ANA1 is the inverting input and ANA2 is the noninverting input.

## Operation

To use the LPO, it must be enabled in the Power Control Register 0 (PWRCTL0). The default state of the LPO is OFF. To use the LPO, the LPO bit must be cleared by turning it ON (for details, see the <u>Power Control Register 0</u> section on page 33). When making normal ADC measurements on ANA0 (i.e., measurements not involving the LPO output), the LPO bit must be turned OFF. Turning the LPO bit ON interferes with normal ADC measurements.

**Caution:** The LPO bit enables the amplifier even in STOP Mode. If the amplifier is not required in STOP Mode, disable it. Failing to perform this results in STOP Mode currents higher than necessary.

As with other ADC measurements, any pins used for analog purposes must be configured as such in the GPIO registers. See the <u>Port A–D Alternate Function Subregisters</u> section on page 47 for details.

LPO output measurements are made on ANA0, as selected by the ANAIN[3:0] bits of ADC Control Register 0. It is also possible to make single-ended measurements on ANA1 and ANA2 while the amplifier is enabled, which is often useful for determining offset conditions. Differential measurements between ANA0 and ANA2 may be useful for noise cancellation purposes.

If the LPO output is routed to the ADC, then the BUFFMODE[2:0] bits of ADC Control/Status Register 1 must also be configured for unity-gain buffered operation. Sampling the LPO in an unbuffered mode is not recommended.

When either input is overdriven, the amplifier output saturates at the positive or negative supply voltage. No instability results.

In the above equation, T is the temperature in °C; V is the sensor output in volts.

Assuming a compensated ADC measurement, the following equation defines the relationship between the ADC reading and the die temperature:

 $T = (25/128) \times (ADC - TSCAL[11:2]) + 30$ 

In the above equation, T is the temperature in C; ADC is the 10-bit compensated ADC value; and TSCAL is the temperature sensor calibration value, ignoring the two least significant bits of the 12-bit value.

See the <u>Temperature Sensor Calibration Data</u> section on page 171 for the location of TSCAL.

#### Calibration

The temperature sensor undergoes calibration during the manufacturing process and is maximally accurate at 30°C. Accuracy decreases as measured temperatures move further from the calibration point.

## Flash Operation Timing Using the Flash Frequency Registers

Before performing either a program or erase operation on Flash memory, you must first configure the Flash Frequency High and Low Byte registers. The Flash Frequency registers allow programming and erasing of the Flash with system clock frequencies ranging from 32kHz (32768Hz) through 20MHz.

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

 $FFREQ[15:0] = \frac{System Clock Frequency (Hz)}{1000}$ 

**Caution:** Flash programming and erasure are not supported for system clock frequencies below 32kHz (32768Hz) or above 20MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure operation of the Z8 Encore! XP F082A Series devices.

## Flash Code Protection Against External Access

The user code contained within the Flash memory can be protected against external access by the on-chip debugger. Programming the FRP Flash option bit prevents reading of the user code with the On-Chip Debugger. See the <u>Flash Option Bits</u> chapter on page 159 and the <u>On-Chip Debugger</u> chapter on page 180 for more information.

# Flash Code Protection Against Accidental Program and Erasure

The Z8 Encore! XP F082A Series provides several levels of protection against accidental program and erasure of the Flash memory contents. This protection is provided by a combination of the Flash option bits, the register locking mechanism, the page select redundancy and the sector level protection control of the Flash Controller.

#### Flash Code Protection Using the Flash Option Bits

The FRP and FWP Flash option bits combine to provide three levels of Flash Program Memory protection, as shown in Table 79. See the <u>Flash Option Bits</u> chapter on page 159 for more information.

## **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 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.

Bit	7 6		5	4	3	2	1	0			
Field	SPROT7	SPROT6 SPROT5 SPRO		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			
Address		FF9H									

Table 83. Flash Sector Protect Register (FPROT)

#### Bit Description

#### [7:0] Sector Protection

- SPROT*n* Each bit corresponds to a 1024-byte Flash sector on devices in the 8K range, while the remaining devices correspond to a 512-byte Flash sector. To determine the appropriate Flash memory sector address range and sector number for your Z8F082A Series product, please refer to <u>Table 78</u> on page 146 and to Figure 21, which follows the table.
  - For Z8F08xA and Z8F04xA devices, all bits are used.
  - For Z8F02xA devices, the upper 4 bits are unused.
  - For Z8F01xA devices, the upper 6 bits are unused.

## Flash Frequency High and Low Byte Registers

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

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

## Trim Bit Address 0004H

#### Table 95. Trim Option Bits at 0004H

Bit	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				
Address	Information Page Memory 0024H											
Note: U =	lote: U = Unchanged by Reset. R/W = Read/Write.											

#### Bit Description

#### [7:0] Reserved

These bits are reserved; altering this register may result in incorrect device operation.

## **Zilog Calibration Data**

This section briefly describes the features of the following Flash option bit calibration registers.

ADC Calibration Data: see page 169

Temperature Sensor Calibration Data: see page 171

Watchdog Timer Calibration Data: see page 172

Serialization Data: see page 173

Randomized Lot Identifier: see page 174

 If the PA2/RESET pin is held Low while a 32-bit key sequence is issued to the PA0/ DBG pin, the DBG feature is unlocked. After releasing PA2/RESET, it is pulled High. At this point, the PA0/DBG pin may be used to autobaud and cause the device to enter DEBUG Mode. See the <u>OCD Unlock Sequence (8-Pin Devices Only) section on</u> page 185.

#### **Exiting DEBUG Mode**

The device exits DEBUG Mode following any of these operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset
- Watchdog Timer reset
- Asserting the RESET pin Low to initiate a Reset
- Driving the DBG pin Low while the device is in STOP Mode initiates a System Reset

### OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character transmitted and received by the OCD consists of 1 Start bit, 8 data bits (least-significant bit first) and 1 Stop bit as displayed in Figure 26.

	START	D0	D1	D2	D3	D4	D5	D6	D7	STOP	
--	-------	----	----	----	----	----	----	----	----	------	--

Figure 26. OCD Data Format

**Note:** When responding to a request for data, the OCD may commence transmitting immediately after receiving the stop bit of an incoming frame. Therefore, when sending the stop bit, the host must not actively drive the DBG pin High for more than 0.5 bit times. Zilog recommends that, if possible, the host drives the DBG pin using an open drain output to avoid this issue.

## **OCD Auto-Baud Detector/Generator**

To run over a range of baud rates (data bits per second) with various system clock frequencies, the On-Chip Debugger contains an Auto-Baud Detector/Generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the

in DEBUG Mode or the Flash Read Protect Option bit is enabled, this command reads and discards one byte.

DBG  $\leftarrow$  12H DBG  $\leftarrow$  1-5 byte opcode

## **On-Chip Debugger Control Register Definitions**

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

## **OCD Control Register**

The OCD Control Register controls the state of the On-Chip Debugger. This register is used to enter or exit DEBUG Mode and to enable the BRK instruction. It can also reset the Z8 Encore! XP F082A Series device.

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

Bit	7	6	5	4	4 3 2 1							
Field	DBGMODE	BRKEN	DBGACK		Reserved							
RESET	0	0	0	0	0 0 0 0							
R/W	R/W	R/W	R/W	R	R	R	R	R/W				

Table 110. OCD Control Register (OCDCTL)

#### Bit Description

#### [7] **DEBUG Mode** DBGMODE The device enters DEBUG Mode when this bit is 1. When in DEBUG Mode, the eZ8 CPU stops fetching new instructions. Clearing this bit causes the eZ8 CPU to restart. This bit is automatically set when a BRK instruction is decoded and Breakpoints are enabled. If the Flash Read Protect Option Bit is enabled, this bit can only be cleared by resetting the device. It cannot be written to 0. 0 = The Z8 Encore! XP F082A Series device is operating in NORMAL Mode. 1 = The Z8 Encore! XP F082A Series device is in DEBUG Mode. [6] **Breakpoint Enable** BRKEN This bit controls the behavior of the BRK instruction (opcode 00H). By default, Breakpoints are disabled and the BRK instruction behaves similar to an NOP instruction. If this bit is 1, when a BRK instruction is decoded, the DBGMODE bit of the OCDCTL Register is automatically set to 1. 0 = Breakpoints are disabled.

1 = Breakpoints are enabled.

Assembly		Address Mode		Opcode(s)			Fla	ags			Fetch Cycle	Instr. Cvcle
Mnemonic	Symbolic Operation	dst	src	(Hex)	С	Ζ	S	۷	D	Н	S	S
ORX dst, src	$dst \gets dst \ OR \ src$	ER	ER	48	-	*	*	0	_	_	4	3
		ER	IM	49	-						4	3
POP dst	dst ← @SP	R		50	_	_	_	_	_	_	2	2
	$SP \leftarrow SP + 1$	IR		51	-						2	3
POPX dst	dst $\leftarrow @SP$ SP $\leftarrow$ SP + 1	ER		D8	-	-	-	-	-	-	3	2
PUSH src	$SP \leftarrow SP - 1$	R		70	-	_	_	-	_	_	2	2
	$@SP \leftarrow src$	IR		71	-						2	3
		IM		IF70	-						3	2
PUSHX src	$SP \leftarrow SP - 1$ @SP ← src	ER		C8	-	_	_	-	-	_	3	2
RCF	C ← 0			CF	0	-	-	-	-	-	1	2
RET	$\begin{array}{l} PC \leftarrow @SP \\ SP \leftarrow SP + 2 \end{array}$			AF	-	_	_	-	-	_	1	4
RL dst		R		90	*	*	*	*	-	-	2	2
	C	IR		91	-						2	3
RLC dst		R		10	*	*	*	*	_	_	2	2
	└─ <u> </u> C <b>  ←  </b> D7 D6 D5 D4 D3 D2 D1 D0  ← <sup>J</sup> dst	IR		11	-						2	3
RR dst	]	R		E0	*	*	*	*	_	_	2	2
	► D7 D6 D5 D4 D3 D2 D1 D0 C	IR		E1	-						2	3

#### Table 128. eZ8 CPU Instruction Summary (Continued)

Note: Flags Notation:

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

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

## Z8 Encore! XP<sup>®</sup> F082A Series Product Specification

## Ζ

Z8 Encore! block diagram 3 features 1 part selection guide 2