E·XFL



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 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f012ahh020eg

Email: info@E-XFL.COM

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

Overview

Zilog's Z8 Encore! MCU family of products are the first in a line of Zilog microcontroller products based upon the 8-bit eZ8 CPU. Zilog's Z8 Encore! XP F082A Series products expand upon Zilog's extensive line of 8-bit microcontrollers. The Flash in-circuit programming capability allows for faster development time and program changes in the field. The new eZ8 CPU is upward compatible with existing Z8 instructions. The rich peripheral set of the Z8 Encore! XP F082A Series makes it suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices and sensors.

Features

The key features of Z8 Encore! XP F082A Series products include:

- 20 MHz eZ8 CPU
- 1KB, 2KB, 4KB, or 8KB Flash memory with in-circuit programming capability
- 256B, 512B, or 1KB register RAM
- Up to 128B nonvolatile data storage (NVDS)
- Internal precision oscillator trimmed to $\pm 1\%$ accuracy
- External crystal oscillator, operating up to 20MHz
- Optional 8-channel, 10-bit analog-to-digital converter (ADC)
- Optional on-chip temperature sensor
- On-chip analog comparator
- Optional on-chip low-power operational amplifier (LPO)
- Full-duplex UART
- The UART baud rate generator (BRG) can be configured and used as a basic 16-bit timer
- Infrared Data Association (IrDA)-compliant infrared encoder/decoders, integrated with the UART
- Two enhanced 16-bit timers with capture, compare and PWM capability
- Watchdog Timer (WDT) with dedicated internal RC oscillator
- Up to 20 vectored interrupts
- 6 to 25 I/O pins depending upon package
- Up to thirteen 5 V-tolerant input pins

Internal Precision Oscillator

The internal precision oscillator (IPO) is a trimmable clock source that requires no external components.

Temperature Sensor

The optional temperature sensor produces an analog output proportional to the device temperature. This signal can be sent to either the ADC or the analog comparator.

Analog Comparator

The analog comparator compares the signal at an input pin with either an internal programmable voltage reference or a second input pin. The comparator output can be used to drive either an output pin or to generate an interrupt.

External Crystal Oscillator

The crystal oscillator circuit provides highly accurate clock frequencies with the use of an external crystal, ceramic resonator or RC network.

Low Voltage Detector

The low voltage detector (LVD) is able to generate an interrupt when the supply voltage drops below a user-programmable level. The LVD is available on 8-pin devices only.

On-Chip Debugger

The Z8 Encore! XP F082A Series products feature an integrated on-chip debugger (OCD) accessed via a single-pin interface. The OCD provides a rich-set of debugging capabilities, such as reading and writing registers, programming Flash memory, setting breakpoints and executing code.

Universal Asynchronous Receiver/Transmitter

The full-duplex universal asynchronous receiver/transmitter (UART) is included in all Z8 Encore! XP package types. The UART supports 8- and 9-bit data modes and selectable parity. The UART also supports multi-drop address processing in hardware. The UART baud rate generator (BRG) can be configured and used as a basic 16-bit timer.

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

)	Table 4. Pin Characteristics (8-Pin Devices)									
Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt- Trigger Input	Open Drain Output	5V Tolerance		
PA0/DBG	I/O	I (but can change during reset if key sequence detected)	N/A	Yes	Programma- ble Pull-up	Yes	Yes, Programma- ble	Yes, unless pull-ups enabled		
PA1	I/O	I	N/A	Yes	Programma- ble Pull-up	Yes	Yes, Programma- ble	Yes, unless pull-ups enabled		
RESET/ PA2	I/O	I/O (defaults to RESET)	Low (in Reset mode)	Yes	Programma- ble for PA2; alw <u>ays on f</u> or RESET	Yes	Programma- ble for PA2; alw <u>ays on f</u> or RESET	Yes, unless pull-ups enabled		
PA[5:3]	I/O	I	N/A	Yes	Programma- ble Pull-up	Yes	Yes, Programma- ble	Yes, unless pull-ups enabled		
V _{DD}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A		
V _{SS}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A		

15

Address Space

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

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral and general-purpose I/O port control registers.
- The Program Memory contains addresses for all memory locations having executable code and/or data.
- The Data Memory contains addresses for all memory locations that contain data only.

These three address spaces are covered briefly in the following subsections. For more information about eZ8 CPU and its address space, refer to the <u>eZ8 CPU Core User Manual</u> (<u>UM0128</u>), which is available for download on <u>www.zilog.com</u>.

Register File

The Register File address space in the Z8 Encore! MCU is 4 KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as sources are read and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4 KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256 B control register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The Z8 Encore! XP^{TM} F082A Series devices contain 256 B to 1KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

Program Memory

The eZ8 CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP F082A Series devices contain 1 KB to 8KB of on-chip Flash memory in the Program Memory address space, depending on the device. Reading from Program Memory

Bit	Description (Continued)
[4] 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.
[3]	UART 0 Transmitter Interrupt Request
U0TXI	0 = No interrupt request is pending for the UART 0 transmitter.
	1 = An interrupt request from the UART 0 transmitter is awaiting service.
[2:1]	Reserved
	These bits are reserved and must be programmed to 00.
[0]	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, shown in Table 36, 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.

Bit	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
Address	FC3H							

Table 36. Interrupt Request 1 Register (IRQ1)

Bit	Description						
[7]	Port A Pin 7 or LVD Interrupt Request						
PA7V	I 0 = No interrupt request is pending for GPIO Port A or LVD.						
	1 = An interrupt request from GPIO Port A or LVD.						
[6]	Port A Pin 6 or Comparator Interrupt Request						
PA6C	I 0 = No interrupt request is pending for GPIO Port A or Comparator.						
	1 = An interrupt request from GPIO Port A or Comparator.						
[5:0]	Port A Pin <i>x</i> Interrupt Request						
PA5I	0 = No interrupt request is pending for GPIO Port A pin x.						
	1 = An interrupt request from GPIO Port A pin <i>x</i> is awaiting service.						
Note:	x indicates the specific GPIO port pin number (0–5).						

ADCI

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	Reserved				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

enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer Reload.

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

- 1. Write to the Timer Control Register to:
 - Disable the timer.
 - Configure the timer for COUNTER Mode.
 - Select either the rising edge or falling edge of the Timer Input signal for the count. This selection also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function is not required to be enabled.
- 2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in COUNTER Mode. After the first timer Reload in COUNTER Mode, counting always begins at the reset value of 0001H. In COUNTER Mode the Timer High and Low Byte registers must be written with the value 0001H.
- 3. Write to the Timer Reload High and Low Byte registers to set the reload value.
- 4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 5. Configure the associated GPIO port pin for the Timer Input alternate function.
- 6. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
- 7. Write to the Timer Control Register to enable the timer.

In COUNTER Mode, the number of Timer Input transitions since the timer start is computed via the following equation:

COUNTER Mode Timer Input Transitions = Current Count Value-Start Value

COMPARATOR COUNTER Mode

In COMPARATOR COUNTER Mode, the timer counts input transitions from the analog comparator output. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the comparator output signal. In COMPAR-ATOR COUNTER Mode, the prescaler is disabled.

- 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 Description (Continued)

[6] **GATED Mode**

- TPOL (cont'd)
- 0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input.
 - 1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input.

CAPTURE/COMPARE Mode

- 0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal.
- 1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.

PWM DUAL OUTPUT Mode

- 0 = Timer Output is forced Low (0) and Timer Output Complement is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon reload. When enabled, the Timer Output Complement is forced Low (0) upon PWM count match and forced High (1) upon reload. The PWMD field in TxCTL0 Register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to High (1).
- 1 = Timer Output is forced High (1) and Timer Output Complement is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon reload. When enabled, the Timer Output Complement is forced High (1) upon PWM count match and forced Low (0) upon reload. The PWMD field in TxCTL0 Register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to Low (0).

CAPTURE RESTART Mode

- 0 = Count is captured on the rising edge of the Timer Input signal.
- 1 = Count is captured on the falling edge of the Timer Input signal.

COMPARATOR COUNTER Mode

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 upon timer Reload. Also:

0 =Count is captured on the rising edge of the comparator output.

1 = Count is captured on the falling edge of the comparator output.

Caution: When the Timer Output alternate function TxOUT on a GPIO port pin is enabled, TxOUT changes to whatever state the TPOL bit is in. The timer does not need to be enabled for that to happen. Also, the Port Data Direction Subregister is not required to be set to output on TxOUT. Changing the TPOL bit with the timer enabled and running does not immediately change the TxOUT.

Table 62. Watchdog Timer Reload Low Byte Register (WDTL)

Bit	7	6	5	4	3	2	1	0			
Field		WDTL									
RESET		00H									
R/W		R/W*									
Address	FF3H										
Note: A re	Note: A read returns the current WDT count value; a write sets the appropriate reload value.										
Di+	Doscriptio	n									

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

111

Table 63. UART Control 0 Register (U0CTL0)

Bit	7	6	5	4	3	2	1	0			
Field	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN			
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				F4	2H						
<u> </u>	D :										
Bit	Description	n 									
[/] TEN	This bit ena and the CT 0 = Transm 1 = Transm	This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is Low and the CTSE bit is 1, the transmitter is enabled. 0 = Transmitter disabled. 1 = Transmitter enabled.									
[6] REN	Receive Enable This bit enables or disables the receiver. 0 = Receiver disabled. 1 = Receiver enabled.										
[5] CTSE	CTS Enabl 0 = The CT 1 = The UA	<u>e</u> S signal has .RT recogniz	s no eff <u>ect o</u> zes the CTS	n the transm signal as ar	nitter. n enable cor	ntrol from the	e transmitter				
[4] PEN	Parity Enal This bit enal 0 = Parity is 1 = The transitional parity	ble ables or disa s disabled. nsmitter sen [,] bit.	bles parity. I ds data with	Even or odd	is determine al parity bit a	ed by the PS and the rece	SEL bit. eiver receive	s an addi-			
[3] PSEL	Parity Sele 0 = Even pa 1 = Odd pa	e ct arity is trans rity is transn	mitted and e	expected on a	all received all received o	data. data.					
[2] SBRK	Send Break This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so ensure that the transmitter has finished sending data before setting this bit. 0 = No break is sent. 1 = Forces a break condition by setting the output of the transmitter to zero.										
[1] STOP	Stop Bit Set 0 = The transition 1 = The transition 1	elect nsmitter sen nsmitter sen	ds one stop ds two stop	bit. bits.							
[0] LBEN	Loop Back Enable 0 = Normal operation. 1 = All transmitted data is looped back to the receiver.										

For the reserved values, all input switches are disabled to avoid leakage or other undesirable operation. ADC samples taken with reserved bit settings are undefined.

SINGLE-ENDED Mode:

0000 = ANA0 (transimpedance amp output when enabled) 0001 = ANA1 (transimpedance amp inverting input) 0010 = ANA2 (transimpedance amp noninverting input) 0011 = ANA3 0100 = ANA4 0101 = ANA5 0110 = ANA6 0111 = ANA7 1000 = Reserved 1001 = Reserved 1010 = Reserved 1011 = Reserved 1100 = Hold transimpedance input nodes (ANA1 and ANA2) to ground. 1101 = Reserved 1110 = Temperature Sensor.

1111 = Reserved.

DIFFERENTIAL Mode (noninverting input and inverting input respectively):

- 0000 = ANA0 and ANA1 0001 = ANA2 and ANA3 0010 = ANA4 and ANA5 0011 = ANA4 and ANA5 0011 = ANA1 and ANA0 0100 = ANA3 and ANA2 0101 = ANA5 and ANA4 0110 = ANA6 and ANA5 0111 = ANA0 and ANA5 1001 = ANA0 and ANA3 1001 = ANA0 and ANA4 1010 = ANA0 and ANA5 1011 = Reserved 1100 = Reserved 1101 = Reserved 1101 = Reserved1110 = Reserved
- 1111 = Manual Offset Calibration Mode

ADC Control/Status Register 1

The ADC Control/Status Register 1 (ADCCTL1) configures the input buffer stage, enables the threshold interrupts and contains the status of both threshold triggers. It is also used to select the voltage reference configuration.

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.

Caution: The byte at each address of the Flash memory cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs. Doing so may result in corrupted data at the target byte.

Page Erase

The Flash memory can be erased one page (512 bytes) at a time. Page Erasing the Flash memory sets all bytes in that page to the value FFH. The Flash Page Select Register identifies the page to be erased. Only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95h to the Flash Control Register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the Page Erase operation completes. If the Page Erase operation is performed using the On-Chip Debugger, poll the Flash Status Register to determine when the Page Erase operation is complete. When the Page Erase is complete, the Flash Controller returns to its locked state.

Mass Erase

The Flash memory can also be Mass Erased using the Flash Controller, but only by using the On-Chip Debugger. Mass Erasing the Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked and the Mass Erase successfully enabled, writing the 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 about bypassing the Flash Controller, refer to the <u>Third-Party Flash</u> <u>Programming Support for Z8 Encore! MCUs Application Note (AN0117)</u>, which is available for download on <u>www.zilog.com</u>.

Flash Program Memory Address 0001H

Table 89. Flash Options Bits at Program Memory Address 0001H

Bit	7	6	5	4	3	2	1	0
Field	Reserved			XTLDIS	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	Program Memory 0001H							

Note: U = Unchanged by Reset. R/W = Read/Write.

Bit	Description				
[7:5]	Reserved				
	These bits are reserved and must be programmed to 111.				
[4]	State of the Crystal Oscillator at Reset				
XTLDIS	This bit only enables the crystal oscillator. Its selection as a system clock must be performed manually.				
	0 = Crystal oscillator is enabled during reset, resulting in longer reset timing.				
	1 = Crystal oscillator is disabled during reset, resulting in shorter reset timing.				
	Caution: Programming the XTLDIS bit to zero on 8-pin versions of this device prevents any further communication via the debug pin due to the fact that the XIN and DBG functions are shared on pin 2 of this package. Do not program this bit to zero on 8-pin devices unless further debugging or Flash programming is not required.				
[3:0]	Reserved These bits are reserved and must be programmed to 1111				

Operation	Minimum Latency	Maximum Latency
Read (16 byte array)	875	9961
Read (64 byte array)	876	8952
Read (128 byte array)	883	7609
Write (16 byte array)	4973	5009
Write (64 byte array)	4971	5013
Write (128 byte array)	4984	5023
Illegal Read	43	43
Illegal Write	31	31

Table 107. NVDS Read Time

If NVDS read performance is critical to your software architecture, you can optimize your code for speed. Try the first suggestion below before attempting the second.

- 1. Periodically refresh all addresses that are used. The optimal use of NVDS in terms of speed is to rotate the writes evenly among all addresses planned to use, bringing all reads closer to the minimum read time. Because the minimum read time is much less than the write time, however, actual speed benefits are not always realized.
- 2. Use as few unique addresses as possible to optimize the impact of refreshing, plus minimize the requirement for it.



Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #2 of 2

DEBUG Mode

The operating characteristics of the devices in DEBUG Mode are:

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

Entering DEBUG Mode

The operating characteristics of the devices entering DEBUG Mode are:

- The device enters DEBUG Mode after the eZ8 CPU executes a BRK (Breakpoint) instruction
- If the DBG pin is held Low during the final clock cycle of system reset, the part enters DEBUG Mode immediately (20-/28-pin products only)

Note: Holding the DBG pin Low for an additional 5000 (minimum) clock cycles after reset (making sure to account for any specified frequency error if using an internal oscillator) prevents a false interpretation of an Autobaud sequence (see the <u>OCD Auto-Baud Detector/Generator</u> section on page 183).

enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

Breakpoints in Flash Memory

The BRK instruction is opcode 00H, which corresponds to the fully programmed state of a byte in Flash memory. To implement a Breakpoint, write 00H to the required break address, overwriting the current instruction. To remove a Breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

Runtime Counter

The On-Chip Debugger contains a 16-bit Runtime Counter. It counts system clock cycles between Breakpoints. The counter starts counting when the On-Chip Debugger leaves DEBUG Mode and stops counting when it enters DEBUG Mode again or when it reaches the maximum count of FFFFH.

On-Chip Debugger Commands

The host communicates to the on-chip debugger by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG Mode, all OCD commands become available unless the user code and control registers are protected by programming the Flash Read Protect Option bit (FRP). The Flash Read Protect Option bit prevents the code in memory from being read out of the Z8 Encore! XP F082A Series device. When this option is enabled, several of the OCD commands are disabled. See Table 109.

<u>Table 110</u> on page 191 is a summary of the on-chip debugger commands. Each OCD command is described in further detail in the bulleted list following this table. Table 110 also indicates those commands that operate when the device is not in DEBUG Mode (normal operation) and those commands that are disabled by programming the Flash Read Protect Option bit.

Debug Command	Command Byte	Enabled when Not in DEBUG Mode?	Disabled by Flash Read Protect Option Bit
Read OCD Revision	00H	Yes	-
Reserved	01H	-	-
Read OCD Status Register	02H	Yes	-
Read Runtime Counter	03H	_	-
Write OCD Control Register	04H	Yes	Cannot clear DBGMODE bit.
Read OCD Control Register	05H	Yes	_

Table 109.	Debug	Command	Enable/Disa	ble

eZ8 CPU Instruction Set

This chapter describes the following features of the eZ8 CPU instruction set: <u>Assembly Language Programming Introduction</u>: see page 204 <u>Assembly Language Syntax</u>: see page 205 <u>eZ8 CPU Instruction Notation</u>: see page 206 <u>eZ8 CPU Instruction Classes</u>: see page 207 <u>eZ8 CPU Instruction Summary</u>: see page 212

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.

Abbreviation	Description	Abbreviation	Description
b	Bit position.	IRR	Indirect register pair.
СС	Condition code.	р	Polarity (0 or 1).
Х	8-bit signed index or displacement.	r	4-bit working register.
DA	Destination address.	R	8-bit register.
ER	Extended addressing register.	r1, R1, Ir1, Irr1, IR1, rr1, RR1, IRR1, ER1	Destination address.
IM	Immediate data value.	r2, R2, Ir2, Irr2, IR2, rr2, RR2, IRR2, ER2	Source address.
Ir	Indirect working register.	RA	Relative.
IR	Indirect register.	rr	Working register pair.
Irr	Indirect working register pair.	RR	Register pair.

Table 129. Opcode Map Abbreviations