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, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f081ash020sg

Email: info@E-XFL.COM

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

List of Tables

Table 1.	Z8 Encore! XP F082A Series Family Part Selection Guide 2
Table 2.	Signal Descriptions
Table 3.	Pin Characteristics (20- and 28-pin Devices) 13
Table 4.	Pin Characteristics (8-Pin Devices) 14
Table 5.	Z8 Encore! XP F082A Series Program Memory Maps 16
Table 6.	Z8 Encore! XP F082A Series Flash Memory Information Area Map 17
Table 7.	Register File Address Map 18
Table 8.	Reset and Stop Mode Recovery Characteristics and Latency 23
Table 9.	Reset Sources and Resulting Reset Type
Table 10.	Stop Mode Recovery Sources and Resulting Action
Table 11.	Reset Status Register (RSTSTAT)
Table 12.	Reset and Stop Mode Recovery Bit Descriptions
Table 13.	Power Control Register 0 (PWRCTL0)
Table 14.	Port Availability by Device and Package Type
Table 15.	Port Alternate Function Mapping (Non 8-Pin Parts) 40
Table 16.	Port Alternate Function Mapping (8-Pin Parts)
Table 17.	GPIO Port Registers and Subregisters
Table 18.	Port A–D GPIO Address Registers (PxADDR)
Table 19.	Port A–D GPIO Address Registers by Bit Description
Table 20.	Port A–D Control Registers (PxCTL)
Table 21.	Port A–D Data Direction Subregisters (PxDD) 46
Table 22.	Port A–D Alternate Function Subregisters (PxAF)
Table 23.	Port A–D Output Control Subregisters (PxOC)
Table 24.	Port A–D High Drive Enable Subregisters (PxHDE)
Table 25.	Port A–D Stop Mode Recovery Source Enable Subregisters (PxSMRE) 49
Table 26.	Port A–D Pull-Up Enable Subregisters (PxPUE)
Table 27.	Port A–D Alternate Function Set 2 Subregisters (PxAFS2) 51
Table 28.	Port A–D Alternate Function Set 1 Subregisters (PxAFS1) 51

Block Diagram

Figure 1 displays the block diagram of the architecture of the Z8 Encore! XP F082A Series devices.





Program Memory Address (Hex)	Function
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-03FF	Program Memory
Note: *See Table 32 on page 56 for a list of	the interrupt vectors.

Data Memory

The Z8 Encore! XP F082A Series does not use the eZ8 CPU's 64 KB Data Memory address space.

Flash Information Area

Table 6 describes the Z8 Encore! XP F082A Series Flash Information Area. This 128B Information Area is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash Information Area is mapped into the Program Memory and overlays the 128 bytes at addresses FE00H to FF7FH. When the Information Area access is enabled, all reads from these Program Memory addresses return the Information Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

Table 6. Z8 Encore! XP F082A Series Flash Memory Information Area Map

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog Option Bits/Calibration Data
FE40–FE53	Part Number 20-character ASCII alphanumeric code Left-justified and filled with FFH
FE54–FE5F	Reserved
FE60–FE7F	Zilog Calibration Data
FE80–FFFF	Reserved



Note: Asserting any power control bit disables the targeted block regardless of any enable bits contained in the target block's control registers.

>

Port A–D Pull-up Enable Subregisters

The Port A–D Pull-up Enable Subregister, shown in Table 26, is accessed through the Port A–D Control Register by writing 06H to the Port A–D Address Register. Setting the bits in the Port A–D Pull-up Enable subregisters enables a weak internal resistive pull-up on the specified port pins.

Bit	7	6	5	4	3	2	1	0	
Field	PPUE7	PPUE6	PPUE5	PPUE4	PPUE3	PPUE2	PPUE1	PPUE0	
RESET		00H (Ports A-C); 01H (Port D); 04H (Port A of 8-pin device)							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Address	If 06H in Port A–D Address Register, accessible through the Port A–D Control Register								
Bit	Descriptio	n							

Table	20			D	Frable	Culture al			ì
lable A	26.	Port	A-D	Pull-Up	Enable	Subregi	sters (PXPUE	.)

Bit	Description
[7:0]	Port Pull-up Enabled
PPUEx	0 = The weak pull-up on the port pin is disabled.
	1 = The weak pull-up on the port pin is enabled.
Note: x ii	ndicates the specific GPIO port pin number (7–0).

Port A–D Alternate Function Set 1 Subregisters

The Port A–D Alternate Function Set1 Subregister, shown in Table 27, is accessed through the Port A–D Control Register by writing 07H to the Port A–D Address Register. The Alternate Function Set 1 subregisters selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register are defined in the <u>GPIO Alternate Functions</u> section on page 37.

Note: Alternate function selection on port pins must also be enabled as described in the <u>Port A</u>– <u>D Alternate Function Subregisters</u> section on page 47.

Table 39. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	7	6	5	4	3	2	1	0	
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	Reserved	Reserved	ADCENH	
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				FC	1H				
Bit	Description								
[7]	Reserved This bit is reserved and must be programmed to 0.								
[6] T1ENH	Timer 1 Interrupt Request Enable High Bit								
[5] T0ENH	Timer 0 Interrupt Request Enable High Bit								
[4]	UART 0 Re	ceive Inter	rupt Reque	st Enable H	igh Bit				

U0RENH	
[3] U0TENH	UART 0 Transmit Interrupt Request Enable High Bit
[2:1]	Reserved These bits are reserved and must be programmed to 00.
[0] ADCENH	ADC Interrupt Request Enable High Bit

Table 40. IRQU Enable Low Bit Register (IRQUENL)
--

Bit	7	6	5	4	3	2	1	0	
Field	Reserved	T1ENL	T0ENL	U0RENL	U0TENL	Reserved	Reserved	ADCENL	
RESET	0	0	0	0	0	0	0	0	
R/W	R	R/W	R/W	R/W	R/W	R	R	R/W	
Address	FC2H								

Bit	Description
[7]	Reserved This bit is reserved and must be programmed to 0.
[6] T1ENL	Timer 1 Interrupt Request Enable Low Bit
[5] T0ENL	Timer 0 Interrupt Request Enable Low Bit

Bit	Description (Continued)
[4] U0RENL	UART 0 Receive Interrupt Request Enable Low Bit
[3] U0TENL	UART 0 Transmit Interrupt Request Enable Low Bit
[2:1]	Reserved
	These bits are reserved and must be programmed to 00.
[0] ADCENL	ADC Interrupt Request Enable Low Bit

IRQ1 Enable High and Low Bit Registers

Table 41 describes the priority control for IRQ1. The IRQ1 Enable High and Low Bit registers, shown in Tables 41 and 42, form a priority-encoded enabling for interrupts in the Interrupt Request 1 Register.

IRQ1ENH[x	IRQ1ENL[<i>x</i>]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Medium
1	1	Level 3	High
Note: x indica	tes register bits 0–7		

Table 41. IRQ1 Enable and Priority Encoding

Table 42. IRQ1	Enable	High Bit	Register	(IRQ1ENH)
----------------	--------	-----------------	----------	-----------

Bit	7	6	5	4	3	2	1	0
Field	PA7VENH	PA6CENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
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		FC4H						

Bit	Description
[7] PA7VENH	Port A Bit[7] or LVD Interrupt Request Enable High Bit
[6] PA6CENH	Port A Bit[7] or Comparator Interrupt Request Enable High Bit
[5:0] PAxENH	Port A Bit[x] Interrupt Request Enable High Bit

See the <u>Shared Interrupt Select Register (IRQSS) Register</u> on page 68 for selection of either the LVD or the comparator as the interrupt source.

Table 43.	IRQ1 E	Enable L	ow Bit	Register	(IRQ1ENL)
					·····

Bit	7	6	5	4	3	2	1	0
Field	PA7VENL	PA6CENL	PA5ENL	PA4ENL	PA3ENL	PA2ENL	PA1ENL	PA0ENL
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				FC	5H			

Bit	Description
[7] PA7VENL	Port A Bit[7] or LVD Interrupt Request Enable Low Bit
[6] PA6CENL	Port A Bit[6] or Comparator Interrupt Request Enable Low Bit
[5:0] PAxENL	Port A Bit[x] Interrupt Request Enable Low Bit

IRQ2 Enable High and Low Bit Registers

Table 44 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers, shown in Tables 44 and 45, form a priority-encoded enabling for interrupts in the Interrupt Request 2 Register.

Table 46.	IRQ2 Enable	Low Bit	Register	(IRQ2ENL)
			riogiotoi	(

Bit	7	6	5	4	3	2	1	0
Field		Rese	erved		C3ENL	C2ENL	C1ENL	C0ENL
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				FC	8H			

Bit	Description
[7:4]	Reserved These bits are reserved and must be programmed to 0000.
[3] C3ENL	Port C3 Interrupt Request Enable Low Bit
[2] C2ENL	Port C2 Interrupt Request Enable Low Bit
[1] C1ENL	Port C1 Interrupt Request Enable Low Bit
[0] C0ENL	Port C0 Interrupt Request Enable Low Bit

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register, shown in Table 47, determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A input pin.

Bit	7	6	5	4	3	2	1	0
Field	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FCDH							

	Table 47	. Interrupt	Edge	Select	Register	(IRQES)
--	----------	-------------	------	--------	----------	---------

Bit	Description
[7:0]	Interrupt Edge Select x
IESx	0 = An interrupt request is generated on the falling edge of the PAx input.
	1 = An interrupt request is generated on the rising edge of the PAx input.
Note:	x indicates the specific GPIO port pin number (0–7).

delay ensures a time gap between the deassertion of one PWM output to the assertion of its complement.

Observe the following steps for configuring a timer for PWM DUAL OUTPUT Mode and initiating the PWM operation:

- 1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for PWM DUAL OUTPUT Mode by writing the TMODE bits in the TxCTL1 Register and the TMODEHI bit in TxCTL0 Register
 - 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 PWM Control Register to set the PWM dead band delay value. The deadband delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM high and low byte registers). It must also be less than the duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).
- 5. 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.
- 6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
- 8. 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) = $\frac{\text{Reload Value xPrescale}}{\text{System Clock Frequency (Hz)}}$

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

Watchdog Timer

The Watchdog Timer (WDT) protects against corrupt or unreliable software, power faults and other system-level problems which may place the Z8 Encore! XP F082A Series devices into unsuitable operating states. The features of Watchdog Timer include:

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

Operation

The Watchdog Timer is a one-shot timer that resets or interrupts the Z8 Encore! XP F082A Series devices when the WDT reaches its terminal count. The Watchdog Timer uses a dedicated on-chip RC oscillator as its clock source. The Watchdog Timer operates in only two modes: ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. Perform an enable by executing the WDT instruction or by setting the WDT_AO Flash option bit. The WDT_AO bit forces the Watchdog Timer to operate immediately upon reset, even if a WDT instruction has not been executed.

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

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

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

WDT Reload Value	WDT Reload Value	Approximate (with 10kHz typical W	Time-Out Delay /DT oscillator frequency)
(Hex)	(Decimal)	Typical	Description
000004	4	400 μs	Minimum time-out delay
FFFFF	16,777,215	28 minutes	Maximum time-out delay

Table 58. Watchdog Timer Approximate Time-Out Delays

into the Watchdog Timer Reload registers results in a one-second time-out at room temperature and 3.3V supply voltage. Time-outs other than one second may be obtained by scaling the calibration values up or down as required.

Note: The Watchdog Timer accuracy still degrades as temperature and supply voltage vary. See <u>Table 137</u> on page 235 for details.

Watchdog Timer Control Register Definitions

This section defines the features of the following Watchdog Timer Control registers.

Watchdog Timer Control Register (WDTCTL): see page 96

Watchdog Timer Reload Upper Byte Register (WDTU): see page 97

Watchdog Timer Reload High Byte Register (WDTH): see page 97

Watchdog Timer Reload Low Byte Register (WDTL): see page 98

Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) Register is a write-only control register. Writing the 55H, AAH unlock sequence to the WDTCTL Register address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL Register address produce no effect on the bits in the WDTCTL Register. The locking mechanism prevents spurious writes to the reload registers. This register address is shared with the read-only Reset Status Register.

Bit	7	6	5	4	3	2	1	0
Field	WDTUNLK							
RESET	Х	Х	Х	Х	Х	Х	Х	Х
R/W	W W W W W W							
Address	s FF0H							
Note: X =	Note: X = Undefined.							

ы	Description
[7:0]	Watchdog Timer Unlock
WDTUNLK	The software must write the correct unlocking sequence to this register before it is allowed
	to modify the contents of the Watchdog Timer reload registers.

Description

Dit

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated interrupt service routine (ISR) performs the following:

- 1. Checks the UART Status 0 Register to determine the source of the interrupt error, break, or received data.
- 2. Reads the data from the UART Receive Data Register if the interrupt was because of data available. If operating in MULTIPROCESSOR (9-bit) Mode, further actions may be required depending on the MULTIPROCESSOR Mode bits MPMD[1:0].
- 3. Clears the UART Receiver interrupt in the applicable Interrupt Request Register.
- 4. Executes the IRET instruction to return from the interrupt-service routine and await more data.

Clear To Send (CTS) Operation

The CTS pin, if enabled by the CTSE bit of the UART Control 0 Register, performs flow control on the outgoing transmit datastream. The Clear To Send ($\overline{\text{CTS}}$) input pin is sampled one system clock before beginning any new character transmission. To delay transmission of the next data character, an external receiver must deassert $\overline{\text{CTS}}$ at least one system clock cycle before a new data transmission begins. For multiple character transmissions, this action is typically performed during Stop Bit transmission. If $\overline{\text{CTS}}$ deasserts in the middle of a character transmission, the current character is sent completely.

MULTIPROCESSOR (9-bit) Mode

The UART features a MULTIPROCESSOR (9-bit) Mode that uses an extra (9th) bit for selective communication when a number of processors share a common UART bus. In MULTIPROCESSOR Mode (also referred to as *9-bit Mode*), the multiprocessor bit (MP) is transmitted immediately following the 8-bits of data and immediately preceding the Stop bit(s) as displayed in Figure 13. The character format is:



Figure 13. UART Asynchronous MULTIPROCESSOR Mode Data Format

byte indicating an overrun error, the Receive Data Register must be read again to clear the error bits is the UART Status 0 Register. Updates to the Receive Data Register occur only when the next data word is received.

UART Data and Error Handling Procedure

Figure 15 displays the recommended procedure for use in UART receiver interrupt service routines.



Figure 15. UART Receiver Interrupt Service Routine Flow

Baud Rate Generator Interrupts

If the baud rate generator (BRG) interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This condition allows the Baud

#3	#3	#3	#3

4. Round the result and discard the least significant two bytes (equivalent to dividing by 2^{16}).

#3	#3	#3	#3
_			
0x00	0x00	0x80	0x00
=			
T	Γ	_	
#4 MSB	#4 LSB		

5. Determine the sign of the gain correction factor using the sign bits from <u>Step 2</u>. If the offset-corrected ADC value *and* the gain correction word both have the same sign, then the factor is positive and remains unchanged. If they have differing signs, then the factor is negative and must be multiplied by -1.

#5 MSB	#5 LSB
--------	--------

6. Add the gain correction factor to the original offset corrected value.

#5 MSB	#5 LSB
+	
#1 MSB	#1 LSB
=	
#6 MSB	#6 LSB
1	1

7. Shift the result to the right, using the sign bit determined in <u>Step 1</u>, to allow for the detection of computational overflow.

	$s \rightarrow$	#6 MSB	#6 LSB
--	-----------------	--------	--------

132

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>.

Trim Bit Data Register

The Trim Bid Data (TRMDR) Register contains the read or write data for access to the trim option bits (Table 87).

Bit	7	6	5	4	3	2	1	0
Field	TRMDR: Trim Bit Data							
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	FF7H							

Table 87. Trim Bit Data Register (TRMDR)

Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses 0000H and 0001H are reserved for the user-programmable Flash option bits.

Flash Program Memory Address 0000H

Bit	7	6	5	4	3	2	1	0
Field	WDT_RES	WDT_AO	OSC_S	EL[1:0]	VBO_AO	FRP	Reserved	FWP
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 0000H							
Note: U = Unchanged by Reset, R/W = Read/Write.								

Table 88. Flash Option Bits at Program Memory Address 0000H

Note.	0 = 0 for angle by Reset. $R/W = Read/Whee.$
	

Bit	Description
[7] WDT_RES	 Watchdog Timer Reset 0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request. 1 = Watchdog Timer time-out causes a system reset. This setting is the default for unprogrammed (erased) Flash.
[6] WDT_AO	 Watchdog Timer Always On 0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled. 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.

Nonvolatile Data Storage

The Z8 Encore! XP F082A Series devices contain a nonvolatile data storage (NVDS) element of up to 128 bytes. This memory can perform over 100,000 write cycles.

Operation

The NVDS is implemented by special purpose Zilog software stored in areas of program memory, which are not user-accessible. These special-purpose routines use the Flash memory to store the data. The routines incorporate a dynamic addressing scheme to maximize the write/erase endurance of the Flash.

Note: Different members of the Z8 Encore! XP F082A Series feature multiple NVDS array sizes; see the <u>Part Selection Guide</u> section on page 2 for details. Devices containing 8KB of Flash memory do not include the NVDS feature.

NVDS Code Interface

Two routines are required to access the NVDS: a write routine and a read routine. Both of these routines are accessed with a CALL instruction to a predefined address outside of the user-accessible program memory. Both the NVDS address and data are single-byte values. Because these routines disturb the working register set, user code must ensure that any required working register values are preserved by pushing them onto the stack or by changing the working register pointer just prior to NVDS execution.

During both read and write accesses to the NVDS, interrupt service is NOT disabled. Any interrupts that occur during the NVDS execution must take care not to disturb the working register and existing stack contents or else the array may become corrupted. Disabling interrupts before executing NVDS operations is recommended.

Use of the NVDS requires 15 bytes of available stack space. Also, the contents of the working register set are overwritten.

For correct NVDS operation, the Flash Frequency registers must be programmed based on the system clock frequency (see **the** <u>Flash Operation Timing Using the Flash Frequency</u> <u>Registers</u> **section on page 149**).

Serial Break leaves the device in DEBUG Mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns High. Because of the open-drain nature of the DBG pin, the host can send a Serial Break to the OCD even if the OCD is transmitting a character.

OCD Unlock Sequence (8-Pin Devices Only)

Because of pin-sharing on the 8-pin device, an unlock sequence must be performed to access the DBG pin. If this sequence is not completed during a system reset, then the PA0/DBG pin functions only as a GPIO pin.

The following sequence unlocks the DBG pin:

- 1. Hold PA2/RESET Low.
- 2. Wait 5ms for the internal reset sequence to complete.
- 3. Send the following bytes serially to the debug pin:

```
DBG \leftarrow 80H (autobaud)
DBG \leftarrow EBH
DBG \leftarrow 5AH
DBG \leftarrow 70H
DBG \leftarrow CDH (32-bit unlock key)
```

- Release PA2/RESET. The PA0/DBG pin is now identical in function to that of the DBG pin on the 20-/28-pin device. To enter DEBUG Mode, reautobaud and write 80H to the OCD Control Register (see the <u>On-Chip Debugger Commands</u> section on page 186).
- **Caution:** Between <u>Steps 3</u> and <u>4</u>, there is an interval during which the 8-pin device is neither in RE-SET nor DEBUG Mode. If a device has been erased or has not yet been programmed, all program memory bytes contain FFH. The CPU interprets this value as an illegal instruction; therefore some irregular behavior can occur before entering DEBUG Mode, and the register values after entering DEBUG Mode will differ from their specified reset values. However, none of these irregularities prevent the programming of Flash memory. Before beginning system debug, Zilog recommends that some legal code be programmed into the 8-pin device and that a RESET occurs.

Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. If Breakpoints are enabled, the OCD enters DEBUG Mode and idles the eZ8 CPU. If Breakpoints are not

Read Memory, Write Register, Read Register, Read Memory CRC, Step Instruction, Stuff Instruction and Execute Instruction commands.

DBG \leftarrow 03H DBG \rightarrow RuntimeCounter[15:8] DBG \rightarrow RuntimeCounter[7:0]

Write OCD Control Register (04H). The Write OCD Control Register command writes the data that follows to the OCDCTL Register. When the Flash Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of returning the device to normal operating mode is to reset the device.

DBG \leftarrow 04H DBG \leftarrow OCDCTL[7:0]

Read OCD Control Register (05H). The Read OCD Control Register command reads the value of the OCDCTL Register.

DBG \leftarrow 05H DBG \rightarrow OCDCTL[7:0]

Write Program Counter (06H). The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG Mode or if the Flash Read Protect Option bit is enabled, the Program Counter (PC) values are discarded.

DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]

Read Program Counter (07H). The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG Mode or if the Flash Read Protect Option bit is enabled, this command returns FFFFH.

```
DBG \leftarrow 07H
DBG \rightarrow ProgramCounter[15:8]
DBG \rightarrow ProgramCounter[7:0]
```

Write Register (08H). The Write Register command writes data to the Register File. Data can be written 1–256 bytes at a time (256 bytes can be written by setting size to 0). If the device is not in DEBUG Mode, the address and data values are discarded. If the Flash Read Protect Option bit is enabled, only writes to the Flash Control registers are allowed and all other register write data values are discarded.

DBG \leftarrow 08H DBG \leftarrow {4'h0,Register Address[11:8]} DBG \leftarrow Register Address[7:0] DBG \leftarrow Size[7:0] DBG \leftarrow 1-256 data bytes