



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
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	6
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	A/D 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f082asb020eg">https://www.e-xfl.com/product-detail/zilog/z8f082asb020eg</a>

ADC Data High Byte Register .....	136
ADC Data Low Byte Register .....	137
Low Power Operational Amplifier .....	139
Operation .....	139
Comparator .....	140
Operation .....	140
Comparator Control Register Definition .....	141
Temperature Sensor .....	144
Temperature Sensor Operation .....	144
Flash Memory .....	146
Architecture .....	146
Flash Information Area .....	147
Operation .....	147
Flash Operation Timing Using the Flash Frequency Registers .....	149
Flash Code Protection Against External Access .....	149
Flash Code Protection Against Accidental Program and Erasure .....	149
Byte Programming .....	151
Page Erase .....	152
Mass Erase .....	152
Flash Controller Bypass .....	152
Flash Controller Behavior in DEBUG Mode .....	153
Flash Control Register Definitions .....	153
Flash Control Register .....	153
Flash Status Register .....	155
Flash Page Select Register .....	156
Flash Sector Protect Register .....	157
Flash Frequency High and Low Byte Registers .....	157
Flash Option Bits .....	159
Operation .....	159
Option Bit Configuration By Reset .....	159
Option Bit Types .....	160
Reading the Flash Information Page .....	161
Flash Option Bit Control Register Definitions .....	161
Trim Bit Address Register .....	161
Trim Bit Data Register .....	162
Flash Option Bit Address Space .....	162
Flash Program Memory Address 0000H .....	162
Flash Program Memory Address 0001H .....	164
Trim Bit Address Space .....	165

Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #2 of 2 .....	182
Figure 26. OCD Data Format .....	183
Figure 27. Recommended 20MHz Crystal Oscillator Configuration .....	199
Figure 28. Connecting the On-Chip Oscillator to an External RC Network .....	201
Figure 29. Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45k $\Omega$ Resistor .....	202
Figure 30. Opcode Map Cell Description .....	222
Figure 31. First Opcode Map .....	224
Figure 32. Second Opcode Map after 1FH .....	225
Figure 33. Typical Active Mode IDD Versus System Clock Frequency .....	231
Figure 34. Port Input Sample Timing .....	240
Figure 35. GPIO Port Output Timing .....	241
Figure 36. On-Chip Debugger Timing .....	242
Figure 37. UART Timing With CTS .....	243
Figure 38. UART Timing Without CTS .....	244

without initiating an interrupt (if enabled for that pin).

---

## Stop Mode Recovery Using the External $\overline{\text{RESET}}$ Pin

When the Z8 Encore! XP F082A Series device is in STOP Mode and the external  $\overline{\text{RESET}}$  pin is driven Low, a system reset occurs. Because of a glitch filter operating on the  $\overline{\text{RESET}}$  pin, the Low pulse must be greater than the minimum width specified, or it is ignored. See the [Electrical Characteristics](#) chapter on page 226 for details.

## Low Voltage Detection

In addition to the Voltage Brown-Out (VBO) Reset described above, it is also possible to generate an interrupt when the supply voltage drops below a user-selected value. For details about configuring the Low Voltage Detection (LVD) and the threshold levels available, see the [Trim Option Bits at Address 0003H \(TLVD\) Register](#) on page 166. The LVD function is available on the 8-pin product versions only.

When the supply voltage drops below the LVD threshold, the LVD bit of the Reset Status (RSTSTAT) Register is set to one. This bit remains one until the low-voltage condition goes away. Reading or writing this bit does not clear it. The LVD circuit can also generate an interrupt when so enabled, see the [GPIO Mode Interrupt Controller](#) chapter on page 55. The LVD bit is not latched; therefore, enabling the interrupt is the only way to guarantee detection of a transient low voltage event.

The LVD functionality depends on circuitry shared with the VBO block; therefore, disabling the VBO also disables the LVD.

## Reset Register Definitions

The following sections define the Reset registers.

### Reset Status Register

The read-only Reset Status (RSTSTAT) Register, shown in Table 11, indicates the source of the most recent Reset event, indicates a Stop Mode Recovery event and indicates a Watchdog Timer time-out. Reading this register resets the upper four bits to 0. This register shares its address with the write-only Watchdog Timer Control Register.

Table 12 lists the bit settings for Reset and Stop Mode Recovery events.

---

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

---

## GPIO Mode Interrupt Controller

The interrupt controller on the Z8 Encore! XP F082A Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 possible interrupt sources with 18 unique interrupt vectors:
  - Twelve GPIO port pin interrupt sources (two interrupt vectors are shared)
  - Eight on-chip peripheral interrupt sources (two interrupt vectors are shared)
- Flexible GPIO interrupts:
  - Eight selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Watchdog Timer and LVD can be configured to generate an interrupt
- Supports vectored and polled interrupts

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt controller has no effect on operation. For more information about interrupt servicing by the eZ8 CPU, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#), which is available for download on [www.zilog.com](http://www.zilog.com).

### Interrupt Vector Listing

Table 34 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

---

► **Note:** Some port interrupts are not available on the 8- and 20-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

---

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 dead-band 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:

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

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.

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

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

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

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

## CAPTURE Mode

In CAPTURE Mode, the current timer count value is recorded when the appropriate 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 timer continues counting. The INPCAP bit in TxCTL0 Register is set to indicate the timer interrupt is because of an input capture event.

The timer continues counting up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt and continues counting. The INPCAP bit in TxCTL0 Register clears indicating the timer interrupt is not because of an input capture event.

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

1. Write to the Timer Control Register to:
  - Disable the timer
  - Configure the timer for CAPTURE Mode
  - 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).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Clear the Timer PWM High and Low Byte registers to 0000H. Clearing these registers allows the software to determine if interrupts were generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt was generated by a Reload.



- Configure the timer for GATED Mode
  - Set the prescale value
2. Write to the Timer High and Low Byte registers to set the starting count value. Writing these registers only affects the first pass in GATED Mode. After the first timer reset in GATED Mode, counting always begins at the reset value of 0001H.
  3. Write to the Timer Reload High and Low Byte registers to set the reload 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 is generated for both input deassertion and reload events. If appropriate, configure the timer interrupt to be generated only at the input deassertion 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. Assert the Timer Input signal to initiate the counting.

### **CAPTURE/COMPARE Mode**

In CAPTURE/COMPARE Mode, the timer begins counting on the first external Timer Input transition. The acceptable transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the system clock.

Every subsequent acceptable transition (after the first) of the Timer Input signal captures the current count value. The Capture value is written to the Timer PWM High and Low Byte registers. When the Capture event occurs, an interrupt is generated, 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 caused by 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 Compare 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 because of an input capture event.

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

1. Write to the Timer Control Register to:
  - Disable the timer
  - Configure the timer for CAPTURE/COMPARE Mode
  - Set the prescale value

## Timer Reload High and Low Byte Registers

The Timer 0–1 Reload High and Low Byte (TxRH and TxRL) registers, shown in Tables 54 and 55, store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the Timer Reload High Byte Register are stored in a temporary holding register. When a write to the Timer Reload Low Byte Register occurs, the temporary holding register value is written to the Timer High Byte Register. This operation allows simultaneous updates of the 16-bit Timer reload value.

In COMPARE Mode, the Timer Reload High and Low Byte registers store the 16-bit Compare value.

**Table 54. Timer 0–1 Reload High Byte Register (TxRH)**

Bit	7	6	5	4	3	2	1	0
Field	TRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F02H, F0AH							

**Table 55. Timer 0–1 Reload Low Byte Register (TxRL)**

Bit	7	6	5	4	3	2	1	0
Field	TRL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F03H, F0BH							

Bit	Description
[7:0] TRH, TRL	<b>Timer Reload Register High and Low</b> These two bytes form the 16-bit reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value which initiates a timer reload to 0001H. In COMPARE Mode, these two bytes form the 16-bit Compare value.

Bit	Description (Continued)
[2] BRGCTL	<p><b>Baud Rate Control</b></p> <p>This bit causes an alternate UART behavior depending on the value of the REN bit in the UART Control 0 Register. When the UART receiver is not enabled (REN=0), this bit determines whether the Baud Rate Generator issues interrupts.</p> <p>0 = Reads from the Baud Rate High and Low Byte registers return the BRG reload value.</p> <p>1 = The Baud Rate Generator generates a receive interrupt when it counts down to 0.</p> <p>Reads from the Baud Rate High and Low Byte registers return the current BRG count value.</p> <p>When the UART receiver is enabled (REN=1), this bit allows reads from the Baud Rate registers to return the BRG count value instead of the reload value.</p> <p>0 = Reads from the Baud Rate High and Low Byte registers return the BRG reload value.</p> <p>1 = Reads from the Baud Rate High and Low Byte registers return the current BRG count value. Unlike the Timers, there is no mechanism to latch the Low Byte when the High Byte is read.</p>
[1] RDAIRQ	<p><b>Receive Data Interrupt Enable</b></p> <p>0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller.</p> <p>1 = Received data does not generate an interrupt request to the Interrupt Controller. Only receiver errors generate an interrupt request.</p>
[0] IREN	<p><b>Infrared Encoder/Decoder Enable</b></p> <p>0 = Infrared Encoder/Decoder is disabled. UART operates normally.</p> <p>1 = Infrared Encoder/Decoder is enabled. The UART transmits and receives data through the Infrared Encoder/Decoder.</p>

The following code example illustrates how to safely enable the comparator:

```
di
ld cmp0, r0 ; load some new configuration
nop
nop          ; wait for output to settle
clr irq0 ; clear any spurious interrupts pending
ei
```

## Comparator Control Register Definition

The Comparator Control Register (CMP0) configures the comparator inputs and sets the value of the internal voltage reference.

**Table 77. Comparator Control Register (CMP0)**

Bit	7	6	5	4	3	2	1	0
Field	INPSEL	INNSEL	REFLVL				Reserved (20-/28-pin) REFLVL (8-pin)	
RESET	0	0	0	1	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F90H							

Bit	Description
[7] INPSEL	<b>Signal Select for Positive Input</b> 0 = GPIO pin used as positive comparator input. 1 = Temperature sensor used as positive comparator input.
[6] INNSEL	<b>Signal Select for Negative Input</b> 0 = Internal reference disabled, GPIO pin used as negative comparator input. 1 = Internal reference enabled as negative comparator input.

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 Temperature Sensor Calibration Data 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.

---

**! 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 [Third-Party Flash Programming Support for Z8 Encore! MCUs Application Note \(AN0117\)](#), which is available for download on [www.zilog.com](http://www.zilog.com).

## Temperature Sensor Calibration Data

**Table 98. Temperature Sensor Calibration High Byte at 003A (TSCALH)**

Bit	7	6	5	4	3	2	1	0
Field	TSCALH							
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 003A							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Bit	Description
[7:0]	<b>Temperature Sensor Calibration High Byte</b>
TSCALH	The TSCALH and TSCALL bytes combine to form the 12-bit temperature sensor offset calibration value. For more details, see Temperature Sensor Operation on page 139.

**Table 99. Temperature Sensor Calibration Low Byte at 003B (TSCALL)**

Bit	7	6	5	4	3	2	1	0
Field	TSCALL							
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 003B							
Note: U = Unchanged by Reset. R/W = Read/Write.								

Bit	Description
[7:0]	<b>Temperature Sensor Calibration Low Byte</b>
TSCALL	The TSCALH and TSCALL bytes combine to form the 12-bit temperature sensor offset calibration value. For usage details, see the <a href="#">Temperature Sensor Operation</a> section on page 144.

# 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 [Part Selection Guide](#) section on page 2 for details. Devices containing 8 KB 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 Flash Operation Timing Using the Flash Frequency Registers](#) section on page 149).



host is the character 80H. The character 80H has eight continuous bits Low (one Start bit plus 7 data bits), framed between High bits. The Auto-Baud Detector measures this period and sets the OCD Baud Rate Generator accordingly.

The Auto-Baud Detector/Generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. For optimal operation with asynchronous datastreams, the maximum recommended baud rate is the system clock frequency divided by 8. The maximum possible baud rate for asynchronous datastreams is the system clock frequency divided by 4, but this theoretical maximum is possible only for low noise designs with clean signals. Table 108 lists minimum and recommended maximum baud rates for sample crystal frequencies.

**Table 108. OCD Baud-Rate Limits**

System Clock Frequency (MHz)	Recommended Maximum Baud Rate (Kbps)	Recommended Standard PC Baud Rate (bps)	Minimum Baud Rate (Kbps)
20.0	2500.0	1,843,200	39
1.0	125.0	115,200	1.95
0.032768 (32kHz)	4.096	2,400	0.064

If the OCD receives a Serial Break (nine or more continuous bits Low) the Auto-Baud Detector/Generator resets. Reconfigure the Auto-Baud Detector/Generator by sending 80H.

## OCD Serial Errors

The On-Chip Debugger can detect any of the following error conditions on the DBG pin:

- Serial Break (a minimum of nine continuous bits Low)
- Framing Error (received `stop` bit is Low)
- Transmit Collision (OCD and host simultaneous transmission detected by the OCD)

When the OCD detects one of these errors, it aborts any command currently in progress, transmits a four character long Serial Break back to the host and resets the Auto-Baud Detector/Generator. A Framing Error or Transmit Collision may be caused by the host sending a Serial Break to the OCD. Because of the open-drain nature of the interface, returning a Serial Break back to the host only extends the length of the Serial Break if the host releases the Serial Break early.

The host transmits a Serial Break on the DBG pin when first connecting to the Z8 Encore! XP F082A Series devices or when recovering from an error. A Serial Break from the host resets the Auto-Baud Generator/Detector but does not reset the OCD Control Register. A

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 ← 80H (autobaud)  
DBG ← EBH  
DBG ← 5AH  
DBG ← 70H  
DBG ← CDH (32-bit unlock key)
4. 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 On-Chip Debugger Commands section on page 186).

---

**! Caution:** Between Steps 3 and 4, there is an interval during which the 8-pin device is neither in RESET 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

the Watchdog Timer failure can be detected. A very slow system clock results in very slow detection times.

---

**!** **Caution:** It is possible to disable the clock failure detection circuitry and all functioning clock sources. In this case, the Z8 Encore! XP F082A Series device ceases functioning and can only be recovered by Power-On-Reset.

---

## Oscillator Control Register Definitions

The Oscillator Control Register (OSCCTL) enables/disables the various oscillator circuits, enables/disables the failure detection/recovery circuitry and selects the primary oscillator, which becomes the system clock.

The Oscillator Control Register must be unlocked before writing. Unlock the Oscillator Control Register by writing the two-step sequence  $E7H$  followed by  $18H$ . The register is locked at successful completion of a register write to the OSCCTL.

**Table 113. Oscillator Control Register (OSCCTL)**

Bit	7	6	5	4	3	2	1	0
Field	INTEN	XTLEN	WDTEN	SOFEN	WDFEN	SCKSEL		
RESET	1	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F86H							

Bit	Description
[7] INTEN	<b>Internal Precision Oscillator Enable</b> 1 = Internal precision oscillator is enabled. 0 = Internal precision oscillator is disabled.
[6] XTLEN	<b>Crystal Oscillator Enable; this setting overrides the GPIO register control for PA0 and PA1</b> 1 = Crystal oscillator is enabled. 0 = Crystal oscillator is disabled.
[5] WDTEN	<b>Watchdog Timer Oscillator Enable</b> 1 = Watchdog Timer oscillator is enabled. 0 = Watchdog Timer oscillator is disabled.
[4] SOFEN	<b>System Clock Oscillator Failure Detection Enable</b> 1 = Failure detection and recovery of system clock oscillator is enabled. 0 = Failure detection and recovery of system clock oscillator is disabled.

		Lower Nibble (Hex)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Upper Nibble (Hex)	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7	3 ,															
	8																
	9																
	A			3.3 CPC r1,r2	3.4 CPC r1,lr2	4.3 CPC R2,R1	4.4 CPC IR2,R1	4.3 CPC R1,IM	4.4 CPC IR1,IM	5.3 CPCX ER2,ER1	5.3 CPCX IM,ER1						
	B																
	C	3.2 SRL R1	3.3 SRL IR1														
	D																
	E									5, 4 LDWX ER2,ER1							
	F																

Figure 32. Second Opcode Map after 1FH

## **G**

GATED mode 88  
 general-purpose I/O 36  
 GPIO 6, 36  
   alternate functions 37  
   architecture 37  
   control register definitions 44  
   input data sample timing 240  
   interrupts 44  
   port A-C pull-up enable sub-registers 50, 51  
   port A-H address registers 45  
   port A-H alternate function sub-registers 47  
   port A-H control registers 46  
   port A-H data direction sub-registers 46  
   port A-H high drive enable sub-registers 48  
   port A-H input data registers 52  
   port A-H output control sub-registers 47  
   port A-H output data registers 52, 53  
   port A-H stop mode recovery sub-registers 49  
   port availability by device 36  
   port input timing 240  
   port output timing 241

## **H**

H 207  
 HALT 209  
 halt mode 33, 209  
 hexadecimal number prefix/suffix 207

## **I**

I2C 6  
 IM 206  
 immediate data 206  
 immediate operand prefix 207  
 INC 208  
 increment 208  
 increment word 208  
 INCW 208  
 indexed 207  
 indirect address prefix 207  
 indirect register 206

indirect register pair 206  
 indirect working register 206  
 indirect working register pair 206  
 infrared encoder/decoder (IrDA) 120  
 Instruction Set 204  
 instruction set, eZ8 CPU 204  
 instructions  
   ADC 208  
   ADCX 208  
   ADD 208  
   ADDX 208  
   AND 210  
   ANDX 210  
   arithmetic 208  
   BCLR 209  
   BIT 209  
   bit manipulation 209  
   block transfer 209  
   BRK 211  
   BSET 209  
   BSWAP 209, 211  
   BTJ 211  
   BTJNZ 211  
   BTJZ 211  
   CALL 211  
   CCF 209  
   CLR 210  
   COM 210  
   CP 208  
   CPC 208  
   CPCX 208  
   CPU control 209  
   CPX 208  
   DA 208  
   DEC 208  
   DECW 208  
   DI 209  
   DJNZ 211  
   EI 209  
   HALT 209  
   INC 208  
   INCW 208  
   IRET 211  
   JP 211