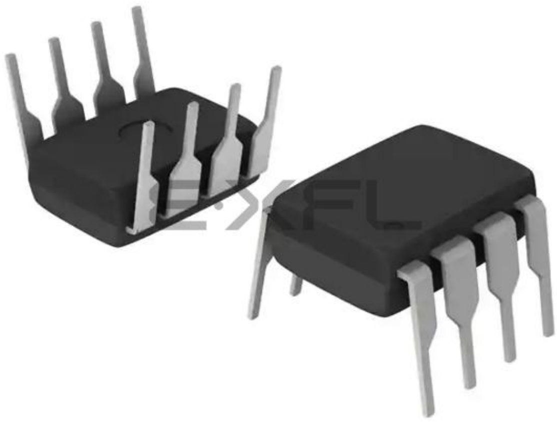



Welcome to E-XFL.COM

## 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 "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | eZ8 |
| Core Size | 8-Bit |
| Speed | 5MHz |
| Connectivity | IrDA, UART/USART |
| Peripherals | Brown-out Detect/Reset, LED, POR, PWM, WDT |
| Number of I/O | 6 |
| Program Memory Size | 2KB (2K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 8-DIP (0.300", 7.62mm) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f0213pb005sg |

# Block Diagram

Figure 1 displays a block diagram of the F0823 Series architecture.

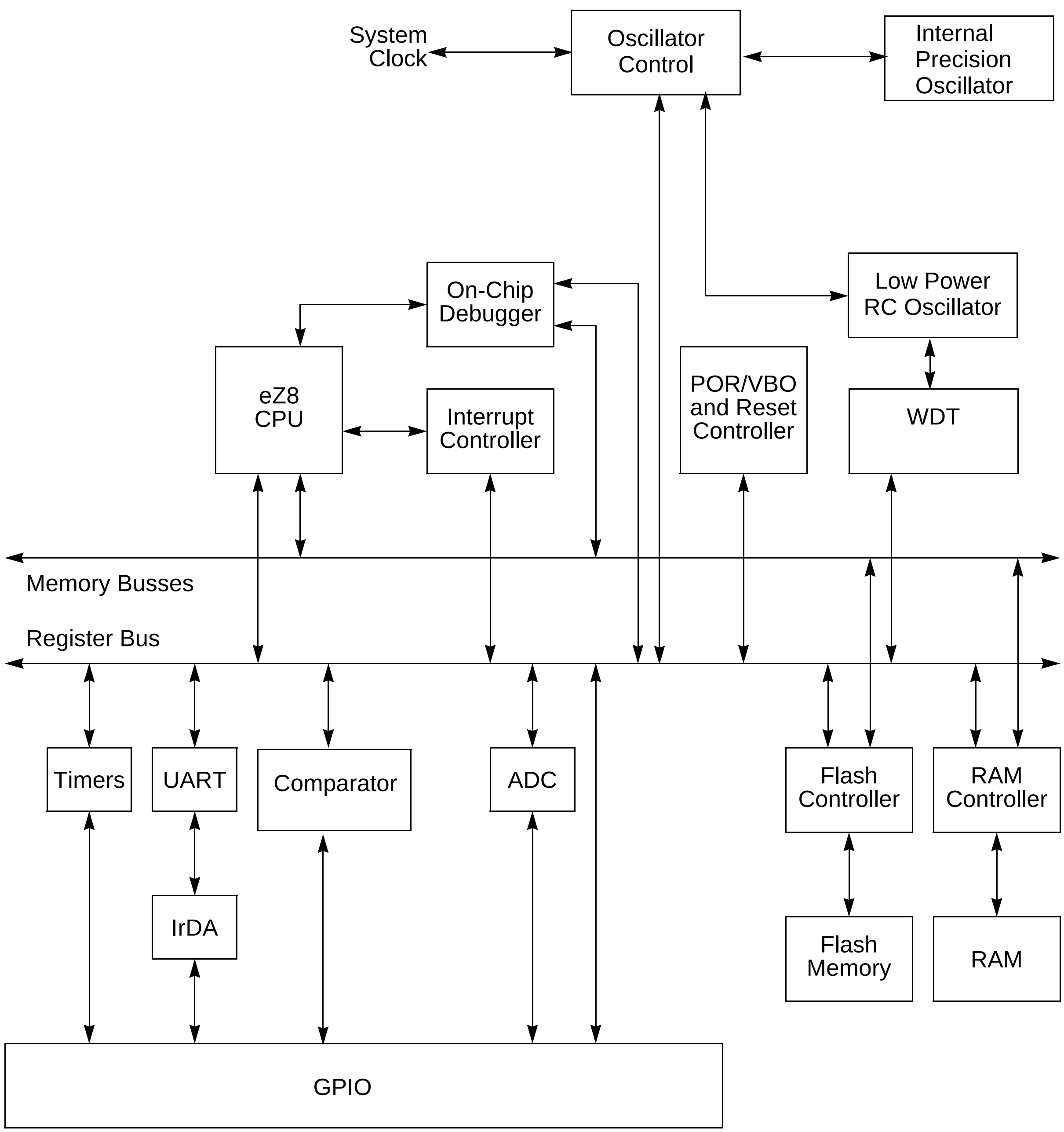



**Figure 1. Z8 Encore! XP F0823 Series Block Diagram**

**Table 8. Register File Address Map (Continued)**

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No. |
|---|---|---|---|---|
| **Timer 1 (cont'd)** | | | | |
| F0A | Timer 1 Reload High Byte | T1RH | FF | 85 |
| F0B | Timer 1 Reload Low Byte | T1RL | FF | 85 |
| F0C | Timer 1 PWM High Byte | T1PWMH | 00 | 86 |
| F0D | Timer 1 PWM Low Byte | T1PWML | 00 | 86 |
| F0E | Timer 1 Control 0 | T1CTL0 | 00 | 87 |
| F0F | Timer 1 Control 1 | T1CTL1 | 00 | 84 |
| F10–F3F | Reserved | — | XX | |
| **UART** | | | | |
| F40 | UART0 Transmit Data | U0TXD | XX | 109 |
| | UART0 Receive Data | U0RXD | XX | 109 |
| F41 | UART0 Status 0 | U0STAT0 | 0000011Xb | 110 |
| F42 | UART0 Control 0 | U0CTL0 | 00 | 112 |
| F43 | UART0 Control 1 | U0CTL1 | 00 | 112 |
| F44 | UART0 Status 1 | U0STAT1 | 00 | 111 |
| F45 | UART0 Address Compare | U0ADDR | 00 | 115 |
| F46 | UART0 Baud Rate High Byte | U0BRH | FF | 115 |
| F47 | UART0 Baud Rate Low Byte | U0BRL | FF | 115 |
| F48–F6F | Reserved | — | XX | |
| **Analog-to-Digital Converter (ADC)** | | | | |
| F70 | ADC Control 0 | ADCCTL0 | 00 | 127 |
| F71 | ADC Control 1 | ADCCTL1 | 80 | 127 |
| F72 | ADC Data High Byte | ADCD_H | XX | 130 |
| F73 | ADC Data Low Bits | ADCD_L | XX | 130 |
| F74–F7F | Reserved | — | XX | |
| **Low Power Control** | | | | |
| F80 | Power Control 0 | PWRCTL0 | 80 | 32 |
| F81 | Reserved | — | XX | |
| **LED Controller** | | | | |
| F82 | LED Drive Enable | LEDEN | 00 | 51 |
| F83 | LED Drive Level High Byte | LEDLVLH | 00 | 52 |

Note: XX=Undefined.

## Interrupt Control Register

The Interrupt Control (IRQCTL) Register (Table 50) contains the master enable bit for all interrupts.

**Table 50. Interrupt Control Register (IRQCTL)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Field** | IRQE | Reserved | | | | | | |
| **RESET** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **R/W** | R/W | R | R | R | R | R | R | R |
| **Address** | FCFH | | | | | | | |

| Bit | Description |
|---|---|
| [7]<br>IRQE | **Interrupt Request Enable**<br>This bit is set to 1 by executing an Enable Interrupts (EI) or Interrupt Return (IRET) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, reset or by a direct register write of a 0 to this bit.<br>0 = Interrupts are disabled.<br>1 = Interrupts are enabled. |
| [6:0] | **Reserved**<br>These bits are reserved and must be programmed to 0000000 when read. |

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 TxCTL1 Register.

6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

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

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

## CAPTURE RESTART Mode

In CAPTURE RESTART Mode, the current timer count value is recorded when the acceptable external Timer Input transition occurs. The capture count value is written to the Timer PWM High and Low Byte registers. The timer input is the system clock. The `TPOL` bit in the Timer Control Register determines if the capture occurs on a rising edge or a falling edge of the Timer Input signal. When the capture event occurs, an interrupt is generated and the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The `INPCAP` bit in TxCTL1 Register is set to indicate the timer interrupt is because of an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The `INPCAP` bit in TxCTL1 Register is cleared to indicate the timer interrupt is not caused by an input capture event.

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

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE RESTART Mode; setting the mode also involves writing to `TMODEHI` bit in TxCTL1 Register
   - Set the prescale value
   - Set the capture edge (rising or falling) for the Timer Input
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.

**Table 61. Watchdog Timer Reload Upper Byte Register (WDTU)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | WDTU | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |
| Address | FF1H | | | | | | | |
| Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value. | | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>WDTU | **WDT Reload Upper Byte**<br>Most significant byte (MSB), Bits[23:16], of the 24-bit WDT reload value. |

**Table 62. Watchdog Timer Reload High Byte Register (WDTH)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | WDTH | | | | | | | |
| RESET | 04H | | | | | | | |
| R/W | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |
| Address | FF2H | | | | | | | |
| Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value. | | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>WDTH | **WDT Reload High Byte**<br>Middle byte, Bits[15:8], of the 24-bit WDT reload value. |

**Table 63. Watchdog Timer Reload Low Byte Register (WDTL)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | WDTL | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |
| Address | FF3H | | | | | | | |
| Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value. | | | | | | | | |

– CEN resets to 0 to indicate the conversion is complete

6. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

## Continuous Conversion

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

---

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

---

Observe the following steps for setting up the ADC and initiating continuous conversion:

1. Enable the acceptable analog input by configuring the general-purpose I/O pins for alternate function. This action disables the digital input and output driver.
2. Write the ADC Control/Status Register 1 to configure the ADC:
   – Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in the ADC Control/Status Register 1.
3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control Register can be written simultaneously:
   – Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
   – Set CONT to 1 to select continuous conversion.
   – If the internal VREF must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
   – Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in ADC Control Register 0.
   – Set CEN to 1 to start the conversions.

| Bit | Description (Continued) |
|---|---|
| [3:0]<br>ANAIN | **Analog Input Select**<br>These bits select the analog input for conversion. Not all port pins in this list are available in all packages for Z8 Encore! XP F0823 Series. For information about the port pins available with each package style, see **the** Pin Description section on page 7. Do not enable unavailable analog inputs. Usage of these bits changes depending on the buffer mode selected in ADC Control/Status Register 1.<br>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.<br>**Single-Ended:**<br>0000 = ANA0.<br>0001 = ANA1.<br>0010 = ANA2.<br>0011 = ANA3.<br>0100 = ANA4.<br>0101 = ANA5.<br>0110 = ANA6.<br>0111 = ANA7.<br>1000 = Reserved.<br>1001 = Reserved.<br>1010 = Reserved.<br>1011 = Reserved.<br>1100 = Reserved.<br>1101 = Reserved.<br>1110 = Reserved.<br>1111 = Reserved. |

# Comparator

Z8 Encore! XP F0823 Series devices feature a general purpose comparator that compares two analog input signals. A GPIO (`CINP`) pin provides the positive comparator input. The negative input (`CINN`) can be taken from either an external GPIO pin or an internal reference. The output is available as an interrupt source or can be routed to an external pin using the GPIO multiplex.

The features of the comparator include:

- Two inputs which can be connected up using the GPIO multiplex (MUX)
- One input can be connected to a programmable internal reference
- One input can be connected to the on-chip temperature sensor
- Output can be either an interrupt source or an output to an external pin

## Operation

One of the comparator inputs can be connected to an internal reference which is a user selectable reference that is user programmable with 200 mV resolution.

The comparator can be powered down to save on supply current. For details, see the Power Control Register 0 section on page 31.

**!** **Caution:** Because of the propagation delay of the comparator, Zilog does not recommend enabling or reconfiguring the comparator without first disabling the interrupts and waiting for the comparator output to settle. Doing so can result in spurious interrupts.

The following example shows how to safely enable the comparator:

```
di
ld cmp0
nop
nop      ; wait for output to settle
clr irq0 ; clear any spurious interrupts pending
ei
```

These serial numbers are stored in the Flash information page (for more details, see the Reading the Flash Information Page section on page 148 and the Serialization Data section on page 154) and are unaffected by mass erasure of the device's Flash memory.

### Randomized Lot Identification Bits

As an optional feature, Zilog is able to provide a factory-programmed random lot identifier. With this feature, all devices in a given production lot are programmed with the same random number. This random number is uniquely regenerated for each successive production lot and is not likely to be repeated.

The randomized lot identifier is a 32-byte binary value, stored in the flash information page (for more details, see the Reading the Flash Information Page section on page 148 and the Randomized Lot Identifier section on page 154) and is unaffected by mass erasure of the device's flash memory.

## Reading the Flash Information Page

The following code example shows how to read data from the Flash Information Area.

```
; get value at info address 60 (FE60h)
ldx FPS, #%80 ; enable access to flash info page
ld R0, #%FE
ld R1, #%60
ldc R2, @RR0 ; R2 now contains the calibration value
```

# Flash Option Bit Control Register Definitions

This section briefly describes the features of the Trim Bit Address and Data registers.

## Trim Bit Address Register

The Trim Bit Address (TRMADR) Register contains the target address for an access to the trim option bits.

**Table 87. Trim Bit Address Register (TRMADR)**

| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| **Field** | TRMADR: Trim Bit Address (00H to 1FH) | | | | | | | |
| **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** | FF6H | | | | | | | |

# Operation

The following section describes the operation of the OCD.

## OCD Interface

The OCD uses the DBG pin for communication with an external host. This one-pin interface is a bidirectional open-drain interface that transmits and receives data. Data transmission is half-duplex, in that transmit and receive cannot occur simultaneously. The serial data on the DBG pin is sent using the standard asynchronous data format defined in RS-232. This pin creates an interface from the F0823 Series products to the serial port of a host PC using minimal external hardware.Two different methods for connecting the DBG pin to an RS-232 interface are displayed in Figure 23 and Figure 24. The recommended method is the buffered implementation depicted in Figure 24. The DBG pin has a internal pull-up resistor which is sufficient for some applications (for more details about the pull-up current, see the Electrical Characteristics chapter on page 196). For OCD operation at higher data rates or in noisy systems, Zilog recommends an external pull-up resistor.

**!** **Caution:** For operation of the OCD, all power pins ($V_{DD}$ and $AV_{DD}$) must be supplied with power, and all ground pins ($V_{SS}$ and $AV_{SS}$) must be properly grounded. The DBG pin is open-drain and may require an external pull-up resistor to ensure proper operation.

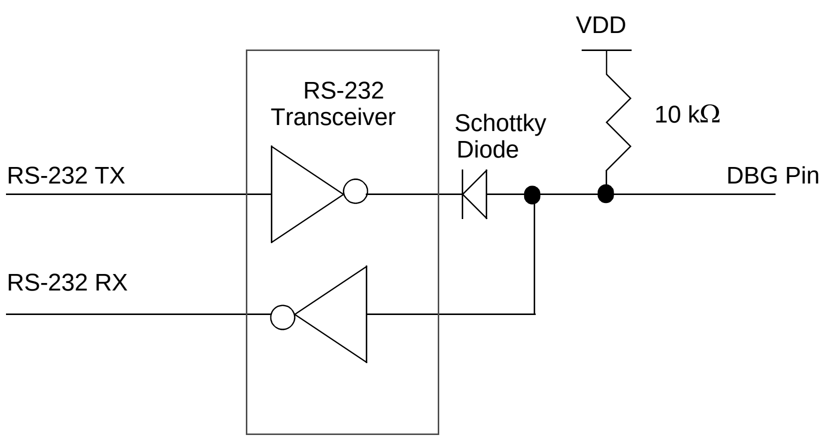



**Figure 23. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 1 of 2**

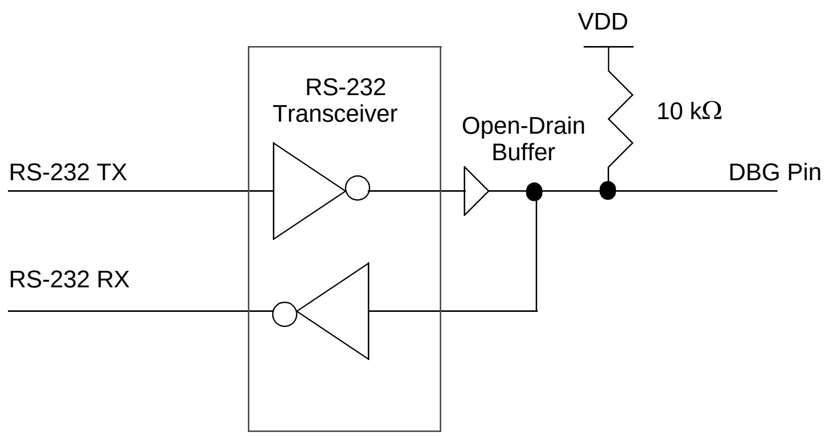



**Figure 24. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 2 of 2**

## 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 device enters DEBUG Mode following the operations below:

- The device enters DEBUG Mode after the eZ8 CPU executes a `BRK` (breakpoint) instruction
- If the DBG pin is held Low during the most recent clock cycle of System Reset, the part enters DEBUG Mode upon exiting System Reset

**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 OCD Autobaud Detector/Generator section on page 159).

# On-Chip Debugger Commands

The host communicates to the OCD 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 Z8 Encore! XP F0823 Series products. When this option is enabled, several of the OCD commands are disabled.

Table 101 is a summary of the OCD commands. Each OCD command is described in further detail in the pages that follow this table. Table 102 on page 167 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.

**Table 101. OCD Commands**

| 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 | – |
| Write Program Counter | 06H | – | Disabled. |
| Read Program Counter | 07H | – | Disabled. |
| Write Register | 08H | – | Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control Register. |
| Read Register | 09H | – | Disabled. |
| Write Program Memory | 0AH | – | Disabled. |
| Read Program Memory | 0BH | – | Disabled. |
| Write Data Memory | 0CH | – | Yes. |
| Read Data Memory | 0DH | – | – |
| Read Program Memory CRC | 0EH | – | – |
| Reserved | 0FH | – | – |
| Step Instruction | 10H | – | Disabled. |
| Stuff Instruction | 11H | – | Disabled. |
| Execute Instruction | 12H | – | Disabled. |
| Reserved | 13H–FFH | – | – |

**Stuff Instruction (11H).** The Stuff command steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0–4 bytes of the instruction are read from Program Memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a Breakpoint. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

```
DBG ← 11H
DBG ← opcode[7:0]
```

**Execute Instruction (12H).** The Execute command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over breakpoints. The number of bytes to send for the instruction depends on the opcode. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, this command reads and discards one byte.

```
DBG ← 12H
DBG ← 1–5 byte opcode
```

# On-Chip Debugger Control Register Definitions

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

## OCD Control Register

The OCD Control Register controls the state of the OCD. This register is used to enter or exit DEBUG Mode and to enable the `BRK` instruction. It also resets Z8 Encore! XP F0823 Series device.

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

## OCD Status Register

The OCD Status Register reports status information about the current state of the debugger and the system.

**Table 103. OCD Status Register (OCDSTAT)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | DBG | HALT | FRPENB | Reserved | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | Description |
|---|---|
| [7]<br>DBG | **Debug Status**<br>0 = NORMAL Mode.<br>1 = DEBUG Mode. |
| [6]<br>HALT | **HALT Mode**<br>0 = Not in HALT Mode.<br>1 = In HALT Mode. |
| [5]<br>FRPENB | **Flash Read Protect Option Bit Enable**<br>0 = FRP bit enabled to allow disabling of many OCD commands.<br>1 = FRP bit has no effect. |
| [4:0] | **Reserved**<br>These bits are reserved and must be 00000 when read. |

**Table 105. Oscillator Control Register (OSCCTL)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **Field** | INTEN | Reserved | WDTEN | POFEN | 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]<br>INTEN | **Internal Precision Oscillator Enable**<br>1 = Internal precision oscillator is enabled.<br>0 = Internal precision oscillator is disabled. |
| [6] | **Reserved**<br>This bit is reserved and must be programmed to 0 during writes and to 0 when read. |
| [5]<br>WDTEN | **Watchdog Timer Oscillator Enable**<br>1 = Watchdog Timer oscillator is enabled.<br>0 = Watchdog Timer oscillator is disabled. |
| [4]<br>POFEN | **Primary Oscillator Failure Detection Enable**<br>1 = Failure detection and recovery of primary oscillator is enabled.<br>0 = Failure detection and recovery of primary oscillator is disabled. |
| [3]<br>WDFEN | **Watchdog Timer Oscillator Failure Detection Enable**<br>1 = Failure detection of Watchdog Timer oscillator is enabled.<br>0 = Failure detection of Watchdog Timer oscillator is disabled. |
| [2:0]<br>SCKSEL | **System Clock Oscillator Select**<br>000 = Internal precision oscillator functions as system clock at 5.53 MHz.<br>001 = Internal precision oscillator functions as system clock at 32 kHz.<br>010 = Reserved.<br>011 = Watchdog Timer oscillator functions as system clock.<br>100 = External clock signal on PB3 functions as system clock.<br>101 = Reserved.<br>110 = Reserved.<br>111 = Reserved. |

# Internal Precision Oscillator

The internal precision oscillator (IPO) is designed for use without external components. You can either manually trim the oscillator for a non-standard frequency or use the automatic factory-trimmed version to achieve a 5.53MHz frequency. The features of IPO include:

- On-chip RC oscillator that does not require external components
- Output frequency of either 5.53MHz or 32.8kHz (contains both a fast and a slow mode)
- Trimming possible through Flash option bits with user override
- Elimination of crystals or ceramic resonators in applications where high timing accuracy is not required

## Operation

An 8-bit trimming register, incorporated into the design, compensates for absolute variation of oscillator frequency. Once trimmed the oscillator frequency is stable and does not require subsequent calibration. Trimming is performed during manufacturing and is not necessary for you to repeat unless a frequency other than 5.53MHz (fast mode) or 32.8kHz (slow mode) is required. This trimming is done at +30°C and a supply voltage of 3.3 V, so accuracy of this operating point is optimal.

Power down this block for minimum system power. By default, the oscillator is configured through the Flash Option bits. However, the user code can override these trim values, as described in the Trim Bit Address Space section on page 151.

Select one of the two frequencies for the oscillator: 5.53MHz and 32.8kHz, using the OSCSEL bits in the Oscillator Control chapter on page 169.

Table 113. CPU Control Instructions

| Mnemonic | Operands | Instruction |
|---|---|---|
| ATM | — | Atomic Execution |
| CCF | — | Complement Carry Flag |
| DI | — | Disable Interrupts |
| EI | — | Enable Interrupts |
| HALT | — | HALT Mode |
| NOP | — | No Operation |
| RCF | — | Reset Carry Flag |
| SCF | — | Set Carry Flag |
| SRP | src | Set Register Pointer |
| STOP | — | STOP Mode |
| WDT | — | Watchdog Timer Refresh |

Table 114. Load Instructions

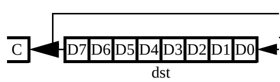
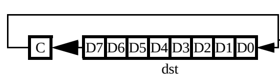
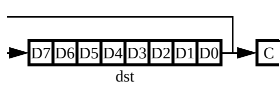
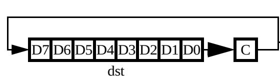
| Mnemonic | Operands | Instruction |
|---|---|---|
| CLR | dst | Clear |
| LD | dst, src | Load |
| LDC | dst, src | Load Constant to/from Program Memory |
| LDCI | dst, src | Load Constant to/from Program Memory and Auto-Increment Addresses |
| LDE | dst, src | Load External Data to/from Data Memory |
| LDEI | dst, src | Load External Data to/from Data Memory and Auto-Increment Addresses |
| LDWX | dst, src | Load Word using Extended Addressing |
| LDX | dst, src | Load using Extended Addressing |
| LEA | dst, X(src) | Load Effective Address |
| POP | dst | Pop |
| POPX | dst | Pop using Extended Addressing |
| PUSH | src | Push |
| PUSHX | src | Push using Extended Addressing |

**Table 118. eZ8 CPU Instruction Summary (Continued)**

| Assembly Mnemonic | Symbolic Operation | Address Mode dst | Address Mode src | Opcode(s) (Hex) | Flags C | Z | S | V | D | H | Fetch Cycles | Instr. Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LD dst, rc | dst ← src | r | IM | 0C-FC | – | – | – | – | – | – | 2 | 2 |
| | | r | X(r) | C7 | | | | | | | 3 | 3 |
| | | X(r) | r | D7 | | | | | | | 3 | 4 |
| | | r | Ir | E3 | | | | | | | 2 | 3 |
| | | R | R | E4 | | | | | | | 3 | 2 |
| | | R | IR | E5 | | | | | | | 3 | 4 |
| | | R | IM | E6 | | | | | | | 3 | 2 |
| | | IR | IM | E7 | | | | | | | 3 | 3 |
| | | Ir | r | F3 | | | | | | | 2 | 3 |
| | | IR | R | F5 | | | | | | | 3 | 3 |
| LDC dst, src | dst ← src | r | Irr | C2 | – | – | – | – | – | – | 2 | 5 |
| | | Ir | Irr | C5 | | | | | | | 2 | 9 |
| | | Irr | r | D2 | | | | | | | 2 | 5 |
| LDCI dst, src | dst ← src<br>r ← r + 1<br>rr ← rr + 1 | Ir | Irr | C3 | – | – | – | – | – | – | 2 | 9 |
| | | Irr | Ir | D3 | | | | | | | 2 | 9 |
| LDE dst, src | dst ← src | r | Irr | 82 | – | – | – | – | – | – | 2 | 5 |
| | | Irr | r | 92 | | | | | | | 2 | 5 |
| LDEI dst, src | dst ← src<br>r ← r + 1<br>rr ← rr + 1 | Ir | Irr | 83 | – | – | – | – | – | – | 2 | 9 |
| | | Irr | Ir | 93 | | | | | | | 2 | 9 |
| LDWX dst, src | dst ← src | ER | ER | 1FE8 | – | – | – | – | – | – | 5 | 4 |

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

**Table 118. eZ8 CPU Instruction Summary (Continued)**

| Assembly Mnemonic | Symbolic Operation | Address Mode dst | Address Mode src | Opcode(s) (Hex) | Flags C | Flags Z | Flags S | Flags V | Flags D | Flags H | Fetch Cycles | Instr. Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| POPX dst | dst ← @SP<br>SP ← SP + 1 | ER | | D8 | – | – | – | – | – | – | 3 | 2 |
| PUSH src | SP ← SP – 1<br>@SP ← src | R | | 70 | – | – | – | – | – | – | 2 | 2 |
| | | IR | | 71 | | | | | | | 2 | 3 |
| | | IM | | IF70 | | | | | | | 3 | 2 |
| PUSHX src | SP ← SP – 1<br>@SP ← src | ER | | C8 | – | – | – | – | – | – | 3 | 2 |
| RCF | C ← 0 | | | CF | 0 | – | – | – | – | – | 1 | 2 |
| RET | PC ← @SP<br>SP ← SP + 2 | | | AF | – | – | – | – | – | – | 1 | 4 |
| RL dst |  | R | | 90 | * | * | * | * | – | – | 2 | 2 |
| | | IR | | 91 | | | | | | | 2 | 3 |
| RLC dst |  | R | | 10 | * | * | * | * | – | – | 2 | 2 |
| | | IR | | 11 | | | | | | | 2 | 3 |
| RR dst |  | R | | E0 | * | * | * | * | – | – | 2 | 2 |
| | | IR | | E1 | | | | | | | 2 | 3 |
| RRC dst |  | R | | C0 | * | * | * | * | – | – | 2 | 2 |
| | | IR | | C1 | | | | | | | 2 | 3 |

Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

**Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)**

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|---|---|---|---|---|---|---|---|
| **Z8 Encore! XP F0823 Series with 2 KB Flash, 10-Bit Analog-to-Digital Converter** | | | | | | | | |
| **Standard Temperature: 0°C to 70°C** | | | | | | | | |
| Z8F0223PB005SG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0223QB005SG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0223SB005SG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0223SH005SG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0223HH005SG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0223PH005SG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0223SJ005SG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0223HJ005SG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0223PJ005SG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |
| **Extended Temperature: –40°C to 105°C** | | | | | | | | |
| Z8F0223PB005EG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0223QB005EG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0223SB005EG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0223SH005EG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0223HH005EG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0223PH005EG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0223SJ005EG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0223HJ005EG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0223PJ005EG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |