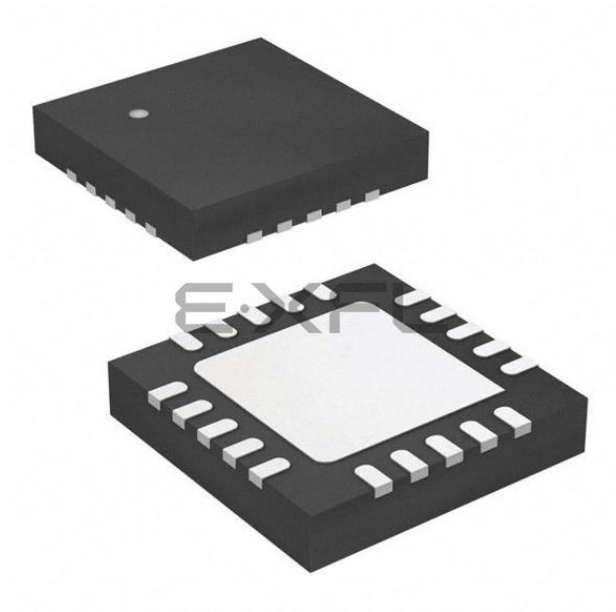

**Welcome to E-XFL.COM**

## 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 | 20MHz |
| Connectivity | - |
| Peripherals | Brown-out Detect/Reset, LED, POR, PWM, WDT |
| Number of I/O | 17 |
| Program Memory Size | 1KB (1K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-VQFN Exposed Pad |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f0131qh020eg |

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

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No. |
|---|---|---|---|---|
| **Interrupt Controller (cont'd)** | | | | |
| FCE | Shared interrupt select | IRQSS | 00 | 66 |
| FCF | Interrupt control | IRQCTL | 00 | 67 |
| **GPIO Port A** | | | | |
| FD0 | Port A address | PAADDR | 00 | 39 |
| FD1 | Port A control | PACTL | 00 | 41 |
| FD2 | Port A input data | PAIN | XX | 41 |
| FD3 | Port A output data | PAOUT | 00 | 41 |
| **GPIO Port B** | | | | |
| FD4 | Port B address | PBADDR | 00 | 39 |
| FD5 | Port B control | PBCTL | 00 | 41 |
| FD6 | Port B input data | PBIN | XX | 41 |
| FD7 | Port B output data | PBOUT | 00 | 41 |
| **GPIO Port C** | | | | |
| FD8 | Port C address | PCADDR | 00 | 39 |
| FD9 | Port C control | PCCTL | 00 | 41 |
| FDA | Port C input data | PCIN | XX | 41 |
| FDB | Port C output data | PCOUT | 00 | 41 |
| **GPIO Port D** | | | | |
| FDC | Port D address | PDADDR | 00 | 39 |
| FDD | Port D control | PDCTL | 00 | 41 |
| FDE | Reserved | — | XX | |
| FDF | Port D output data | PDOUT | 00 | 41 |
| FE0–FEF | Reserved | — | XX | |
| **Watchdog Timer (WDT)** | | | | |
| FF0 | Reset status | RSTSTAT | XX | 95 |
| | Watchdog Timer control | WDTCTL | XX | 95 |
| FF1 | Watchdog Timer reload upper byte | WDTU | FF | 96 |
| FF2 | Watchdog Timer reload high byte | WDTH | FF | 96 |
| FF3 | Watchdog Timer reload low byte | WDTL | FF | 97 |
| FF4–FF5 | Reserved | — | XX | |

Note: XX = Undefined.

**Table 16. Port Alternate Function Mapping (Continued)**

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Set Register AFS1 |
|---|---|---|---|---|
| **Port C[3]** | PC0 | Reserved | | AFS1[0]: 0 |
| | | ANA4/CINP | ADC or comparator input | AFS1[0]: 1 |
| | PC1 | Reserved | | AFS1[1]: 0 |
| | | ANA5/CINN | ADC or comparator input | AFS1[1]: 1 |
| | PC2 | Reserved | | AFS1[2]: 0 |
| | | ANA6 | ADC analog input | AFS1[2]: 1 |
| | PC3 | COUT | Comparator output | AFS1[3]: 0 |
| | | Reserved | | AFS1[3]: 1 |
| | PC4 | Reserved | | AFS1[4]: 0 |
| | | | | AFS1[4]: 1 |
| | PC5 | Reserved | | AFS1[5]: 0 |
| | | | | AFS1[5]: 1 |
| | PC6 | Reserved | | AFS1[6]: 0 |
| | | | | AFS1[6]: 1 |
| | PC7 | Reserved | | AFS1[7]: 0 |
| | | | | AFS1[7]: 1 |
| **Port D[1]** | PD0 | RESET | Default to be Reset function | N/A |

Notes:
1. Because there is only a single alternate function for each Port A and Port D (PD0) pin, the Alternate Function Set registers are not implemented for Port A and Port D (PD0). Enabling alternate function selections (as described in the Port A–D Alternate Function Subregisters section on page 42) automatically enables the associated alternate function.
2. Because there are at most two choices of alternate functions for any Port B pin, the AFS2 Alternate Function Set Register is implemented but is not used to select the function. Additionally, alternate function selection (as described in the Port A–D Alternate Function Subregisters section on page 42) must also be enabled.
3. Because there are at most two choices of alternate functions for any Port C pin, the AFS2 Alternate Function Set Register is implemented but is not used to select the function. Additionally, alternate function selection (as described in the Port A–D Alternate Function Subregisters section on page 42) must also be enabled.

## Port A–D Control Registers

The Port A–D Control registers, shown in Table 20, set the GPIO port operation. The value in the corresponding Port A–D Address Register determines which subregister is read from or written to by a Port A–D Control Register transaction.

**Table 20. Port A–D Control Registers (P*x*CTL)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | PCTL | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FD1H, FD5H, FD9H, FDDH | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>PCTL | **Port Control**<br>The Port Control Register provides access to all subregisters that configure the GPIO port operation. |

## Port A–D Data Direction Subregisters

The Port A–D Data Direction Subregister, shown in Table 21, is accessed through the Port A–D Control Register by writing 01H to the Port A–D Address Register.

**Table 21. Port A–D Data Direction Subregisters (PxDD)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | DD7 | DD6 | DD5 | DD4 | DD3 | DD2 | DD1 | DD0 |
| 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 | If 01H in Port A–D Address Register, accessible through the Port A–D Control Register | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>DDx | **Data Direction**<br>These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction Register setting.<br>0 = Output. Data in the Port A–D Output Data Register is driven onto the port pin.<br>1 = Input. The port pin is sampled and the value written into the Port A–D Input Data Register. The output driver is tristated. |

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

# Interrupt Controller

The Interrupt Controller on the Z8 Encore!® F0830 Series products prioritize the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of the Interrupt Controller include:

- Seventeen interrupt sources using sixteen unique interrupt vectors:
  - Twelve GPIO port pin interrupt sources
  - Five on-chip peripheral interrupt sources (Comparator Output interrupt shares one interrupt vector with PA6)♯
- Flexible GPIO interrupts
  - Eight selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts♯
- Three levels of individually programmable interrupt priority
- Watchdog Timer can be configured to generate an interrupt♯
  m

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 User Manual (UM0128), which is available for download at www.zilog.com.

## Interrupt Vector Listing

Table 34 lists 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 odd program memory address.

**Note:** Some port interrupts are not available on the 20-pin and 28-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

## 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 42 and 43, form a priority-encoded enabling service for interrupts in the Interrupt Request 1 Register. Priority is generated by setting the bits in each register.

**Table 41. IRQ1 Enable and Priority Encoding**

| IRQ1ENH[*x*] | IRQ1ENL[*x*] | Priority | Description |
|---|---|---|---|
| 0 | 0 | Disabled | Disabled |
| 0 | 1 | Level 1 | Low |
| 1 | 0 | Level 2 | Nominal |
| 1 | 1 | Level 3 | High |

Note: x indicates register bits in the address range 7–0.

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | PA7ENH | 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]♯ PA7ENH | **Port A Bit[7] 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**<br>See the interrupt port select register for selection of either Port A or Port D as the interrupt source. |

Note: x indicates register bits in the address range 5–0.

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

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

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

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

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

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

## PWM DUAL OUTPUT Mode

In PWM DUAL OUTPUT Mode, the timer outputs a PWM output signal pair (basic PWM signal and its complement) through two GPIO port pins. The timer input is the system clock. The timer first counts up to 16-bit PWM match value stored in the timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the timer output toggles. The timer continues counting until it reaches the reload 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.

If the TPOL bit in the Timer Control Register is set to 1, the timer output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The timer output signal returns to a High (1) after the timer reaches the reload value and is reset to 0001H.

If the TPOL bit in the Timer Control Register is set to 0, the timer output signal begins as a Low (0) and transitions to a High (1) when the timer value matches the PWM value. The timer output signal returns to a Low (0) after the timer reaches the reload value and is reset to 0001H.

The timer also generates a second PWM output signal: the timer output complement. The timer output complement is the complement of the timer output PWM signal. A programmable deadband delay can be configured to time delay (0 to 128 system clock cycles) PWM output transitions on these two pins from a Low to a High (inactive to active) to ensure a time gap between the deassertion of one PWM output to the assertion of its complement.

| Bit | Description (Continued) |
|---|---|
| [6]# TPOL | **Timer Input/Output Polarity**<br>Operation of this bit is a function of the current operating mode of the timer.<br>**ONE-SHOT Mode**<br>When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled, the timer output signal is complemented on timer reload.<br>**CONTINUOUS Mode**<br>When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled and reloaded, the timer output signal is complemented.<br>**COUNTER Mode**<br>If the timer is disabled, the timer output signal is set to the value of this bit. If the timer is enabled the timer output signal is complemented after timer reload.<br>0 = Count occurs on the rising edge of the timer input signal.<br>1 = Count occurs on the falling edge of the timer input signal.<br>**PWM SINGLE OUTPUT Mode**<br>0 = Timer output is forced Low (0), when the timer is disabled. The timer output is forced High (1) when the timer is enabled and the PWM count matches and the timer output is forced Low (0) when the timer is enabled and reloaded.<br>1 = Timer output is forced High (1), when the timer is disabled. The timer output is forced low(0), when the timer is enabled and the PWM count matches and forced High (1) when the timer is enabled and reloaded.<br>**CAPTURE Mode**<br>0 = Count is captured on the rising edge of the timer input signal.<br>1 = Count is captured on the falling edge of the timer input signal.<br>**COMPARE Mode**<br>When the timer is disabled, the timer output signal is set to the value of this bit. When the timer is enabled and reloaded, the timer output signal is complemented.<br>**GATED Mode**<br>0 = Timer counts when the timer input signal is High (1) and interrupts are generated on the falling edge of the timer input.<br>1 = Timer counts when the timer input signal is Low (0) and interrupts are generated on the rising edge of the timer input.<br>**CAPTURE/COMPARE Mode**<br>0 = Counting is started on the first rising edge of the timer input signal. The current count is captured on subsequent rising edges of the timer input signal.<br>1 = Counting is started on the first falling edge of the timer input signal. The current count is captured on subsequent falling edges of the timer input signal. |

| Bit | Description (Continued) |
|---|---|
| [2:0]<br>TMODE | **Timer Mode**<br>This field along with the TMODEHI bit in TxCTL0 register determines the operating mode of the timer. TMODEHI is the most significant bit of the timer mode selection value.<br>0000 = ONE-SHOT Mode.<br>0001 = CONTINUOUS Mode.<br>0010 = COUNTER Mode.<br>0011 = PWM SINGLE OUTPUT Mode.<br>0100 = CAPTURE Mode.<br>0101 = COMPARE Mode.<br>0110 = GATED Mode.<br>0111 = CAPTURE/COMPARE Mode.<br>1000 = PWM DUAL OUTPUT Mode.<br>1001 = CAPTURE RESTART Mode.<br>1010 = COMPARATOR COUNTER Mode. |

# Watchdog Timer

The Watchdog Timer (WDT) protects from corrupted or unreliable software, power faults and other system-level problems which can place the Z8 Encore! F0830 Series devices into unsuitable operating states. The features of the 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 retriggerable one-shot timer that resets or interrupts the Z8 Encore! F0830 Series devices when the WDT reaches its terminal count. The WDT uses a dedicated on-chip RC oscillator as its clock source. The WDT operates only in 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 WDT to operate immediately on 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 calculated using the following equation:

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

where the WDT reload value is the 24-bit decimal value provided 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.

**Table 58. Watchdog Timer Approximate Time-Out Delays**

| WDT Reload Value (Hex) | WDT Reload Value (Decimal) | Approximate Time-Out Delay (with 10KHz Typical WDT Oscillator Frequency) | |
|---|---|---|---|
| | | Typical | Description |
| 000004 | 4 | 400µs | Minimum time-out delay |
| 000400 | 1024 | 102ms | Default time-out delay |
| FFFFFF | 16,777,215 | 28 minutes | Maximum time-out delay |

# Watchdog Timer Control Register Definitions

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

Watchdog Timer Control Register (WDTCTL): see page 95

Watchdog Timer Reload Low Byte Register (WDTL): see page 97

Watchdog Timer Reload Upper Byte Register (WDTU): see page 96

Watchdog Timer Reload High Byte Register (WDTH): see page 96

## Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) Register is a write-only control register. Writing the unlock sequence: 55H, AAH 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 have 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.

**Table 59. Watchdog Timer Control Register (WDTCTL)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | WDTUNLK | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | W | W | W | W | W | W | W | W |
| Address | FF0H | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>WDTUNLK | **Watchdog Timer Unlock**<br>The user software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer Reload registers. |

# 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 Register, shown in Table 78, contains the target address to access the trim option bits. Trim bit addresses in the range 00h–1Fh map to the information area at addresses 20h–3Fh, as shown in Table 79.

**Table 78. 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 | | | | | | | |

**Table 79. Trim Bit Address Map**

| Trim Bit Address | Information Area Address |
|---|---|
| 00h | 20h |
| 01h | 21h |
| 02h | 22h |
| 03h | 23h |
| : | : |
| 1Fh | 3Fh |

## Trim Bit Data Register

The Trim Bit Data Register, shown in Table 80, contains the read or write data to access the trim option bits.

- Watchdog Timer reset
- Asserting the $\overline{\text{RESET}}$ pin Low to initiate a reset
- Driving the DBG pin Low while the device is in STOP Mode initiates a system reset

## OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 start bit, 8 data bits (least-significant bit first) and 1 stop bit. See Figure 23.

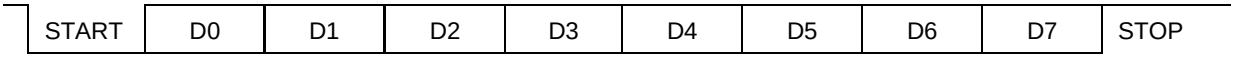


**Figure 23. OCD Data Format**

## OCD Autobaud Detector/Generator

To run over a range of baud rates (data bits per second) with various system clock frequencies, the On-Chip Debugger contains an autobaud detector/generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the 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 autobaud detector measures this period and sets the OCD baud rate generator accordingly.

The autobaud 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 94 lists minimum and recommended maximum baud rates for sample crystal frequencies.

**Table 94. 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 (32 KHz) | 4.096 | 2400 | 0.064 |

**Table 95. On-Chip Debugger Command Summary (Continued)**

| Debug Command | Command Byte | Enabled when not in DEBUG Mode? | Disabled by Flash Read Protect Option Bit |
|---|---|---|---|
| Read Program Memory CRC | 0EH | – | – |
| Reserved | 0FH | – | – |
| Step Instruction | 10H | – | Disabled |
| Stuff Instruction | 11H | – | Disabled |
| Execute Instruction | 12H | – | Disabled |
| Reserved | 13H–FFH | – | – |

In the following bulleted list of OCD commands, data and commands sent from the host to the OCD are identified by `DBG 8 Command/Data`. Data sent from the OCD back to the host is identified by `DBG Data`.

**Read OCD Revision (00H).** The read OCD revision command determines the version of the On-Chip Debugger. If OCD commands are added, removed or changed this revision number changes.

```
DBG ← 00H#
DBG → OCDRev[15:8] (Major revision number)#
DBG → OCDRev[7:0] (Minor revision number)
```

**Read OCD Status Register (02H).** The read OCD Status Register command reads the OCDSTAT register.

```
DBG ← 02H#
DBG → OCDSTAT[7:0]
```

**Read Runtime Counter (03H).** The runtime counter counts system clock cycles in between breakpoints. The 16-bit runtime counter counts from `0000H` and stops at the maximum count of `FFFFH`. The runtime counter is overwritten during the write memory, read memory, write register, read register, read memory CRC, step instruction, stuff instruction and execute instruction commands.

```
DBG ← 03H#
DBG → RuntimeCounter[15:8]#
DBG → 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. To return the device to normal operating mode, the device must be reset.

```
DBG ← 04H#
DBG ← OCDCTL[7:0]
```

When selecting a new clock source, the primary oscillator failure detection circuitry and the Watchdog Timer Oscillator failure circuitry must be disabled. If POFEN and WOFEN are not disabled prior to a clock switch-over, it is possible to generate an interrupt for a failure of either oscillator. The failure detection circuitry can be enabled anytime after a successful write of OSCSEL in the Oscillator Control Register.

The Internal Precision Oscillator is enabled by default. If the user code changes to a different oscillator, it may be appropriate to disable the IPO for power savings. Disabling the IPO does not occur automatically.

## Clock Failure Detection and Recovery

### Primary Oscillator Failure

The Z8F04xA family devices can generate nonmaskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer Oscillator to drive the system clock. The Watchdog Timer Oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer is the primary oscillator. It is also unavailable if the Watchdog Timer Oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function outlined in the Watchdog Timer chapter of this document.

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below 1 KHz ±50%. If an external signal is selected as the system oscillator, it is possible that a very slow but nonfailing clock can generate a failure condition. Under these conditions, do not enable the clock failure circuitry (POFEN must be deasserted in the OSCCTL Register).

### Watchdog Timer Failure

In the event of failure of a Watchdog Timer Oscillator, a similar nonmaskable interrupt-like event is issued. This event does not trigger an attendant clock switch-over, but alerts the CPU of the failure. After a Watchdog Timer failure, it is no longer possible to detect a primary oscillator failure. The failure detection circuitry does not function if the Watchdog Timer is used as the primary oscillator or if the Watchdog Timer Oscillator has been disabled. For either of these cases, it is necessary to disable the detection circuitry by deasserting the WDFEN bit of the OSCCTL Register.

The Watchdog Timer Oscillator failure detection circuit counts system clocks while looking for a Watchdog Timer clock. The logic counts 8004 system clock cycles before determining that a failure has occurred. The system clock rate determines the speed at which the Watchdog Timer failure is detected. A very slow system clock results in very slow detection times.

**Caution:** It is possible to disable the clock failure detection circuitry as well as all functioning clock sources. In this case, the Z8 Encore! F0830 Series device ceases functioning and can only be recovered by power-on-reset.

# Oscillator Control Register Definitions

The following section provides the bit definitions for the Oscillator Control Register.

## Oscillator Control Register

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. Writing the two step sequence E7H followed by 18H to the Oscillator Control Register unlocks it. The register is locked at successful completion of a register write to the OSCCTL.

Figure 24 displays the oscillator control clock switching flow. See Table 117 on page 189 to review the waiting times of various oscillator circuits.

**Table 99. Oscillator Control Register (OSCCTL)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | INTEN | XTLEN | 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]# INTEN | **Internal Precision Oscillator Enable**<br>1 = Internal Precision Oscillator is enabled.<br>0 = Internal Precision Oscillator is disabled. |
| [6]# XTLEN | **Crystal Oscillator Enable**<br>This setting overrides the GPIO register control for PA0 and PA1.<br>1 = Crystal oscillator is enabled.<br>0 = Crystal oscillator is disabled. |
| [5]# WDTEN | **Watchdog Timer Oscillator Enable**<br>1 = Watchdog Timer Oscillator is enabled.<br>0 = Watchdog Timer Oscillator is disabled. |

This example indicates that the source data is added to the destination data; the result is stored in the destination location.

## eZ8 CPU Instruction Classes

eZ8 CPU instructions can be divided functionally into the following groups:

- Arithmetic
- Bit manipulation
- Block transfer
- CPU control
- Load
- Logical
- Program control
- Rotate and shift#

Tables 105 through 112 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instructions can be considered as a subset of more than one category. Within these tables, the source operand is identified as src, the destination operand is dst and a condition code is cc.

**Table 105. Arithmetic Instructions**

| Mnemonic | Operands | Instruction |
|---|---|---|
| ADC | dst, src | Add with Carry |
| ADCX | dst, src | Add with Carry using Extended Addressing |
| ADD | dst, src | Add |
| ADDX | dst, src | Add using Extended Addressing |
| CP | dst, src | Compare |
| CPC | dst, src | Compare with Carry |
| CPCX | dst, src | Compare with Carry using Extended Addressing |
| CPX | dst, src | Compare using Extended Addressing |
| DA | dst | Decimal Adjust |
| DEC | dst | Decrement |
| DECW | dst | Decrement Word |
| INC | dst | Increment |

# AC Characteristics

The section provides information about the AC characteristics and timing. All AC timing information assumes a standard load of 50pF on all outputs.

**Table 117. AC Characteristics**

| | | $V_{DD}$ = 2.7 to 3.6V $T_A$ = 0°C to +70°C | | $V_{DD}$ = 2.7 to 3.6V $T_A$ = –40°C to +105°C | | | |
|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Min** | **Max** | **Min** | **Max** | **Units** | **Conditions** |
| $F_{SYSCLK}$ | System Clock Frequency | | | – | 20.0 | MHz | Read-only from Flash memory |
| | | | | 0.032768 | 20.0 | MHz | Program or erasure of the Flash memory |
| $F_{XTAL}$ | Crystal Oscillator Frequency | | | 1.0 | 20.0 | MHz | System clock frequencies below the crystal oscillator minimum require an external |
| $F_{IPO}$ | Internal Precision Oscillator Frequency | | | 0.032768 | 5.5296 | MHz | Oscillator is **not** adjustable over the entire range. User may select Min or Max value only. |
| $F_{IPO}$ | Internal Precision Oscillator Frequency | | | 5.31 | 5.75 | MHz | High speed with trimming |
| $F_{IPO}$ | Internal Precision Oscillator Frequency | | | 4.15 | 6.91 | MHz | High speed without trimming |
| $F_{IPO}$ | Internal Precision Oscillator Frequency | | | 30.7 | 33.3 | KHz | Low speed with trimming |
| $F_{IPO}$ | Internal Precision Oscillator Frequency | | | 24 | 40 | KHz | Low speed without trimming |
| $T_{XIN}$ | System Clock Period | | | 50 | – | ns | $T_{CLK} = 1/F_{sysclk}$ |
| $T_{XINH}$ | System Clock High Time | | | 20 | 30 | ns | $T_{CLK}$ = 50 ns |
| $T_{XINL}$ | System Clock Low Time | | | 20 | 30 | ns | $T_{CLK}$ = 50 ns |

**Table 128. Z8 Encore! XP F0830 Series Ordering Matrix**

| Part Number | Flash | RAM | NVDS | ADC Channels | Description |
|---|---|---|---|---|---|
| Z8F0131PJ020SG | 1KB | 256 | Yes | 0 | PDIP 28-pin |
| Z8F0131QJ020SG | 1KB | 256 | Yes | 0 | QFN 28-pin |
| **Extended Temperature: –40°C to 105°C** | | | | | |
| Z8F0130SH020EG | 1KB | 256 | Yes | 7 | SOIC 20-pin |
| Z8F0130HH020EG | 1KB | 256 | Yes | 7 | SSOP 20-pin |
| Z8F0130PH020EG | 1KB | 256 | Yes | 7 | PDIP 20-pin |
| Z8F0130QH020EG | 1KB | 256 | Yes | 7 | QFN 20-pin |
| Z8F0131SH020EG | 1KB | 256 | Yes | 0 | SOIC 20-pin |
| Z8F0131HH020EG | 1KB | 256 | Yes | 0 | SSOP 20-pin |
| Z8F0131PH020EG | 1KB | 256 | Yes | 0 | PDIP 20-pin |
| Z8F0131QH020EG | 1KB | 256 | Yes | 0 | QFN 20-pin |
| Z8F0130SJ020EG | 1KB | 256 | Yes | 8 | SOIC 28-pin |
| Z8F0130HJ020EG | 1KB | 256 | Yes | 8 | SSOP 28-pin |
| Z8F0130PJ020EG | 1KB | 256 | Yes | 8 | PDIP 28-pin |
| Z8F0130QJ020EG | 1KB | 256 | Yes | 8 | QFN 28-pin |
| Z8F0131SJ020EG | 1KB | 256 | Yes | 0 | SOIC 28-pin |
| Z8F0131HJ020EG | 1KB | 256 | Yes | 0 | SSOP 28-pin |
| Z8F0131PJ020EG | 1KB | 256 | Yes | 0 | PDIP 28-pin |
| Z8F0131QJ020EG | 1KB | 256 | Yes | 0 | QFN 28-pin |
| ZUSBSC00100ZACG | | | | | USB Smart Cable Accessory Kit |
| ZUSBOPTSC01ZACG | | | | | Opto-Isolated USB Smart Cable Accessory Kit |

# Part Number Suffix Designations

Zilog part numbers consist of a number of components, as indicated in the following example.

**Example.** Part number Z8F0830SH020SG is an 8-bit 20MHz Flash MCU with 8KB Program Memory and equipped with ADC and NVDS in a 20-pin SOIC package, operating within a 0ºC to +70ºC temperature range and built using lead-free solder.

# Customer Support

To share comments, get your technical questions answered or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.