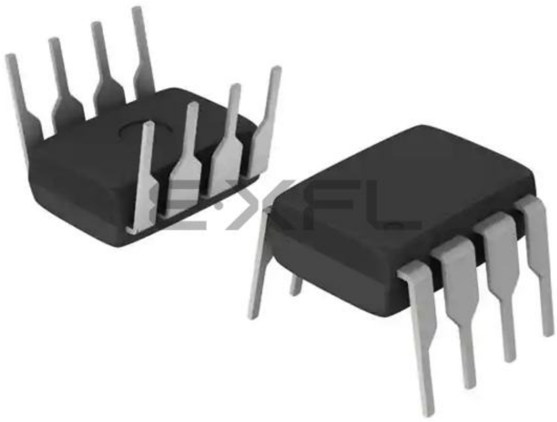

Welcome to E-XFL.COM

## 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 | Obsolete |
| 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 | 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 | A/D 4x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°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/z8f0123pb005ec |

# Table of Contents

**Overview . . . . . . . . . . 1**
Features . . . . . . . . . . 1
Part Selection Guide . . . . . . . . . . 2
Block Diagram . . . . . . . . . . 3
CPU and Peripheral Overview . . . . . . . . . . 4
    eZ8 CPU Features . . . . . . . . . . 4
    General-Purpose I/O . . . . . . . . . . 4
    Flash Controller . . . . . . . . . . 4
    Internal Precision Oscillator . . . . . . . . . . 5
    10-Bit Analog-to-Digital Converter . . . . . . . . . . 5
    Analog Comparator . . . . . . . . . . 5
    Universal Asynchronous Receiver/Transmitter . . . . . . . . . . 5
    Timers . . . . . . . . . . 5
    Interrupt Controller . . . . . . . . . . 5
    Reset Controller . . . . . . . . . . 6
    On-Chip Debugger . . . . . . . . . . 6
**Pin Description . . . . . . . . . . 7**
Available Packages . . . . . . . . . . 7
Pin Configurations . . . . . . . . . . 7
Signal Descriptions . . . . . . . . . . 9
Pin Characteristics . . . . . . . . . . 10
**Address Space . . . . . . . . . . 13**
Register File . . . . . . . . . . 13
Program Memory . . . . . . . . . . 13
Data Memory . . . . . . . . . . 15
Flash Information Area . . . . . . . . . . 15
**Register Map . . . . . . . . . . 17**
**Reset and Stop Mode Recovery . . . . . . . . . . 21**
Reset Types . . . . . . . . . . 21
Reset Sources . . . . . . . . . . 22
    Power-On Reset . . . . . . . . . . 23
    Voltage Brownout Reset . . . . . . . . . . 24
    Watchdog Timer Reset . . . . . . . . . . 25
    External Reset Input . . . . . . . . . . 25
    External Reset Indicator . . . . . . . . . . 26

**Table 3. Signal Descriptions (Continued)**

| Signal Mnemonic | I/O | Description |
|---|---|---|
| **Analog** | | |
| ANA[7:0] | I | Analog port. These signals are used as inputs to the ADC. The ANA0, ANA1, and ANA2 pins can also access the inputs and output of the integrated transimpedance amplifier. |
| VREF | I/O | Analog-to-Digital Converter reference voltage input. |
| **Clock Input** | | |
| CLKIN | I | Clock Input Signal. This pin can be used to input a TTL-level signal to be used as the system clock. |
| **LED Drivers** | | |
| LED | O | Direct LED drive capability. All port C pins have the capability to drive an LED without any other external components. These pins have programmable drive strengths set by the GPIO block. |
| **On-Chip Debugger** | | |
| DBG | I/O | Debug. This signal is the control and data input and output to and from the OCD.<br>**!** **Caution:** *The DBG pin is open-drain and requires an external pull-up resistor to ensure proper operation.* |
| **Reset** | | |
| $\overline{\text{RESET}}$ | I/O | RESET. Generates a reset when asserted (driven Low). Also serves as a reset indicator; the Z8 Encore! XP forces this pin Low when in reset. This pin is open-drain and features an enabled internal pull-up resistor. |
| **Power Supply** | | |
| $V_{DD}$ | I | Digital Power Supply. |
| $AV_{DD}$ | I | Analog Power Supply. |
| $V_{SS}$ | I | Digital Ground. |
| $AV_{SS}$ | I | Analog Ground. |

**Note:** The $AV_{DD}$ and $AV_{SS}$ signals are available only in 28-pin packages with ADC. They are replaced by PB6 and PB7 on 28-pin packages without ADC.

## Pin Characteristics

Table 4 provides detailed information about the characteristics for each pin available on Z8 Encore! XP F0823 Series 20- and 28-pin devices. Data in Table 4 is sorted alphabetically by the pin symbol mnemonic.

**Note:** *This register is only reset during a Power-On Reset sequence. Other System Reset events do not affect it.*

**Table 13. Power Control Register 0 (PWRCTL0)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | Reserved | Reserved | | VBO | Reserved | ADC | COMP | Reserved |
| RESET | 1 | 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 |
| ADDR | F80H | | | | | | | |

Reserved—Must be 1

Reserved—Must be 0

VBO—Voltage Brownout Detector Disable
This bit and the VBO_AO Flash option bit must both enable the VBO for the VBO to be active.
0 = VBO enabled
1 = VBO disabled

ADC—Analog-to-Digital Converter Disable
0 = Analog-to-Digital Converter enabled
1 = Analog-to-Digital Converter disabled

COMP—Comparator Disable
0 = Comparator is enabled
1 = Comparator is disabled

Reserved—Must be 0

## Architecture

Figure 7 displays a simplified block diagram of a GPIO port pin. In this figure, the ability to accommodate alternate functions and variable port current drive strength is not displayed.

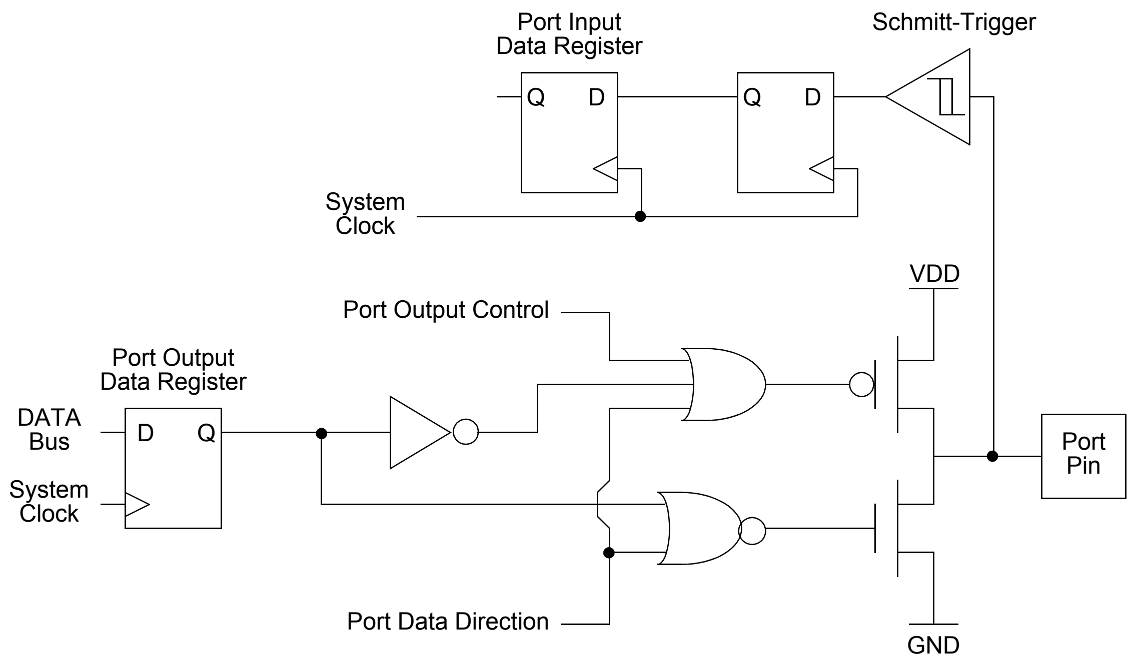


**Figure 7. GPIO Port Pin Block Diagram**

## GPIO Alternate Functions

Many of the GPIO port pins are used for general-purpose I/O and access to on-chip peripheral functions such as the timers and serial communication devices. The port A–D Alternate Function sub-registers configure these pins for either GPIO or alternate function operation. When a pin is configured for alternate function, control of the port pin direction (input/output) is passed from the Port A–D Data Direction registers to the alternate function assigned to this pin. Table 15 on page 39 lists the alternate functions possible with each port pin. The alternate function associated at a pin is defined through Alternate Function Sets sub-registers AFS1 and AFS2.

The crystal oscillator functionality is not controlled by the GPIO block. When the crystal oscillator is enabled in the oscillator control block, the GPIO functionality of PA0 and PA1 is overridden. In that case, those pins function as input and output for the crystal oscillator.

**Table 15. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)**

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Set Register AFS1 |
|---|---|---|---|---|
| **Port B** | PB0 | Reserved | | AFS1[0]: 0 |
| | | ANA0 | ADC Analog Input | AFS1[0]: 1 |
| | PB1 | Reserved | | AFS1[1]: 0 |
| | | ANA1 | ADC Analog Input | AFS1[1]: 1 |
| | PB2 | Reserved | | AFS1[2]: 0 |
| | | ANA2 | ADC Analog Input | AFS1[2]: 1 |
| | PB3 | CLKIN | External Clock Input | AFS1[3]: 0 |
| | | ANA3 | ADC Analog Input | AFS1[3]: 1 |
| | PB4 | Reserved | | AFS1[4]: 0 |
| | | ANA7 | ADC Analog Input | AFS1[4]: 1 |
| | PB5 | Reserved | | AFS1[5]: 0 |
| | | VREF* | ADC Voltage Reference | AFS1[5]: 1 |
| | PB6 | Reserved | | AFS1[6]: 0 |
| | | Reserved | | AFS1[6]: 1 |
| | PB7 | Reserved | | AFS1[7]: 0 |
| | | Reserved | | AFS1[7]: 1 |

**Note:** Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, alternate function selection as described in Port A–C Alternate Function Sub-Registers must also be enabled.
* VREF is available on PB5 in 28-pin products only.

## GPIO Interrupts

Many of the GPIO port pins are used as interrupt sources. Some port pins are configured to generate an interrupt request on either the rising edge or falling edge of the pin input signal. Other port pin interrupt sources generate an interrupt when any edge occurs (both rising and falling). For more information about interrupts using the GPIO pins, see Interrupt Controller on page 53.

## GPIO Control Register Definitions

Four registers for each Port provide access to GPIO control, input data, and output data. Table 17 lists these Port registers. Use the Port A–D Address and Control registers together to provide access to sub-registers for Port configuration and control.

**Table 17. GPIO Port Registers and Sub-Registers**

| Port Register Mnemonic | Port Register Name |
|---|---|
| P*x*ADDR | Port A–C Address Register (Selects sub-registers) |
| P*x*CTL | Port A–C Control Register (Provides access to sub-registers) |
| P*x*IN | Port A–C Input Data Register |
| P*x*OUT | Port A–C Output Data Register |
| **Port Sub-Register Mnemonic** | **Port Register Name** |
| P*x*DD | Data Direction |
| P*x*AF | Alternate Function |
| P*x*OC | Output Control (Open-Drain) |
| P*x*HDE | High Drive Enable |
| P*x*SMRE | Stop Mode Recovery Source Enable |
| P*x*PUE | Pull-up Enable |
| PxAFS1 | Alternate Function Set 1 |
| PxAFS2 | Alternate Function Set 2 |

**Table 30. LED Drive Enable (LEDEN)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | LEDEN[7:0] | | | | | | | |
| 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 |
| ADDR | F82H | | | | | | | |

LEDEN[7:0]—LED Drive Enable
These bits determine which Port C pins are connected to an internal current sink.
0 = Tristate the Port C pin.
1= Connect controlled current sink to the Port C pin.

## LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin (Table 31). These two bits select between four programmable drive levels. Each pin is individually programmable.

**Table 31. LED Drive Level High Register (LEDLVLH)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | LEDLVLH[7:0] | | | | | | | |
| 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 |
| ADDR | F83H | | | | | | | |

LEDLVLH[7:0]—LED Level High Bit
{LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin.
00 = 3 mA
01= 7 mA
10= 13 mA
11= 20 mA

## LED Drive Level Low Register

The LED Drive Level registers contain two control bits for each Port C pin (Table 32). These two bits select between four programmable drive levels. Each pin is individually programmable.

**Table 43. IRQ2 Enable and Priority Encoding (Continued)**

| IRQ2ENH[x] | IRQ2ENL[x] | Priority | Description |
|---|---|---|---|
| 1 | 1 | Level 3 | High |

where *x* indicates the register bits from 0–7.

**Table 44. IRQ2 Enable High Bit Register (IRQ2ENH)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | Reserved | | | | C3ENH | C2ENH | C1ENH | C0ENH |
| 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 |
| ADDR | FC7H | | | | | | | |

Reserved—Must be 0

C3ENH—Port C3 Interrupt Request Enable High Bit
C2ENH—Port C2 Interrupt Request Enable High Bit
C1ENH—Port C1 Interrupt Request Enable High Bit
C0ENH—Port C0 Interrupt Request Enable High Bit

**Table 45. IRQ2 Enable Low Bit Register (IRQ2ENL)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | Reserved | | | | C3ENL | C2ENL | C1ENL | C0ENL |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| ADDR | FC8H | | | | | | | |

Reserved—Must be 0

C3ENL—Port C3 Interrupt Request Enable Low Bit
C2ENL—Port C2 Interrupt Request Enable Low Bit
C1ENL—Port C1 Interrupt Request Enable Low Bit
C0ENL—Port C0 Interrupt Request Enable Low Bit

## Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register (Table 46) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A or Port D input pin.

of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the timer continues counting. The `INPCAP` bit in TxCTL1 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 TxCTL1 register clears indicating the timer interrupt is not because of an input capture event.

Follow the steps below 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 or a Reload event. If the PWM High and Low Byte registers still contain `0000H` after the interrupt, the interrupt was generated by a Reload.
5. 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 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

The UART is now configured for interrupt-driven data transmission. Because the UART Transmit Data register is empty, an interrupt is generated immediately. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Write the UART Control 1 register to select the multiprocessor bit for the byte to be transmitted:

   Set the Multiprocessor Bit Transmitter (`MPBT`) if sending an address byte, clear it if sending a data byte.

2. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Transmit Shift register and transmits the data.
3. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.
4. Execute the `IRET` instruction to return from the interrupt-service routine and wait for the Transmit Data register to again become empty.

## Receiving Data using the Polled Method

Follow the steps below to configure the UART for polled data reception:

1. Write to the UART Baud Rate High and Low Byte registers to set an acceptable baud rate for the incoming data stream.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Write to the UART Control 1 register to enable MULTIPROCESSOR mode functions, if appropriate.
4. Write to the UART Control 0 register to:
   - Set the receive enable bit (`REN`) to enable the UART for data reception
   - Enable parity, if appropriate and if Multiprocessor mode is not enabled, and select either even or odd parity
5. Check the RDA bit in the UART Status 0 register to determine if the Receive Data register contains a valid data byte (indicated by a 1). If `RDA` is set to 1 to indicate available data, continue to step 6. If the Receive Data register is empty (indicated by a 0), continue to monitor the RDA bit awaiting reception of the valid data.
6. Read data from the UART Receive Data register. If operating in MULTIPROCESSOR (9-bit) mode, further actions may be required depending on the MULTIPROCESSOR mode bits MPMD[1:0].
7. Return to step 4 to receive additional data.

## Operation

The Flash Controller programs and erases Flash memory. The Flash Controller provides the proper Flash controls and timing for Byte Programming, Page Erase, and Mass Erase of Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels.

The Flowchart in Figure 21 displays basic Flash Controller operation. The following subsections provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select Page Erase, and Mass Erase) displayed in Figure 21.

## Flash Sector Protect Register

The Flash Sector Protect (FPROT) register is shared with the Flash Page Select Register. When the Flash Control Register is written with 73H followed by 5EH, the next write to this address targets the Flash Sector Protect Register. In all other cases, it targets the Flash Page Select Register.

This register selects one of the 8 available Flash memory sectors to be protected. The reset state of each Sector Protect bit is an unprotected state. After a sector is protected by setting its corresponding register bit, it cannot be unprotected (the register bit cannot be cleared) without powering down the device.

**Table 82. Flash Sector Protect Register (FPROT)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | SPROT7 | SPROT6 | SPROT5 | SPROT4 | SPROT3 | SPROT2 | SPROT1 | SPROT0 |
| 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 |
| ADDR | FF9H | | | | | | | |

SPROT7-SPROT0—Sector Protection
Each bit corresponds to a 512 bytes Flash sector. For the Z8F08x3 devices, the upper 3 bits must be zero. For the Z8F04x3 devices all bits are used. For the Z8F02x3 devices, the upper 4 bits are unused. For the Z8F01x3 devices, the upper 6 bits are unused.

## Flash Frequency High and Low Byte Registers

The Flash Frequency High (FFREQH) and Low Byte (FFREQL) registers combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation:

$$\text{FFREQ[15:0]} = \{\text{FFREQH[7:0]},\text{FFREQL[7:0]}\} = \frac{\text{System Clock Frequency}}{1000}$$

**! Caution:** *The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device. Also, Flash programming and erasure is not supported for system clock frequencies below 20 kHz or above 20 MHz.*

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

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

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

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

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

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

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

- **Read Register (09H)**—The Read Register command reads data from the Register File. Data can be read 1–256 bytes at a time (256 bytes can be read by setting size to 0). If the device is not in DEBUG mode or if the Flash Read Protect Option bit is enabled, this command returns `FFH` for all the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

- **Write Program Memory (0AH)**—The Write Program Memory command writes data to Program Memory. This command is equivalent to the `LDC` and `LDCI` instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device

## OCD Status Register

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

**Table 100. OCD Status Register (OCDSTAT)**

| BITS | 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 |

DBG—Debug Status
0 = NORMAL mode
1 = DEBUG mode

HALT—HALT Mode
0 = Not in HALT mode
1 = In HALT mode

FRPENB—Flash Read Protect Option Bit Enable
0 = FRP bit enabled, that allows disabling of many OCD commands
1 = FRP bit has no effect

Reserved—0 when read

**Table 105. Notational Shorthand**

| Notation | Description | Operand | Range |
|---|---|---|---|
| b | Bit | b | b represents a value from 0 to 7 (000B to 111B). |
| cc | Condition Code | — | See Condition Codes overview in the eZ8 CPU User Manual. |
| DA | Direct Address | Addrs | Addrs represents a number in the range of 0000H to FFFFH. |
| ER | Extended Addressing Register | Reg | Reg represents a number in the range of 000H to FFFH. |
| IM | Immediate Data | #Data | Data is a number between 00H to FFH. |
| Ir | Indirect Working Register | @Rn | n = 0–15. |
| IR | Indirect Register | @Reg | Reg. represents a number in the range of 00H to FFH. |
| Irr | Indirect Working Register Pair | @RRp | p = 0, 2, 4, 6, 8, 10, 12, or 14. |
| IRR | Indirect Register Pair | @Reg | Reg represents an even number in the range 00H to FEH |
| p | Polarity | p | Polarity is a single bit binary value of either 0B or 1B. |
| r | Working Register | Rn | n = 0–15. |
| R | Register | Reg | Reg. represents a number in the range of 00H to FFH. |
| RA | Relative Address | X | X represents an index in the range of +127 to –128 which is an offset relative to the address of the next instruction |
| rr | Working Register Pair | RRp | p = 0, 2, 4, 6, 8, 10, 12, or 14. |
| RR | Register Pair | Reg | Reg. represents an even number in the range of 00H to FEH. |
| Vector | Vector Address | Vector | Vector represents a number in the range of 00H to FFH. |
| X | Indexed | #Index | The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to -128 range. |

**Table 115. 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| LDX dst, src | dst ← src | r | ER | 84 | – | – | – | – | – | – | 3 | 2 |
| | | Ir | ER | 85 | | | | | | | 3 | 3 |
| | | R | IRR | 86 | | | | | | | 3 | 4 |
| | | IR | IRR | 87 | | | | | | | 3 | 5 |
| | | r | X(rr) | 88 | | | | | | | 3 | 4 |
| | | X(rr) | r | 89 | | | | | | | 3 | 4 |
| | | ER | r | 94 | | | | | | | 3 | 2 |
| | | ER | Ir | 95 | | | | | | | 3 | 3 |
| | | IRR | R | 96 | | | | | | | 3 | 4 |
| | | IRR | IR | 97 | | | | | | | 3 | 5 |
| | | ER | ER | E8 | | | | | | | 4 | 2 |
| | | ER | IM | E9 | | | | | | | 4 | 2 |
| LEA dst, X(src) | dst ← src + X | r | X(r) | 98 | – | – | – | – | – | – | 3 | 3 |
| | | rr | X(rr) | 99 | | | | | | | 3 | 5 |
| MULT dst | dst[15:0] ← dst[15:8] * dst[7:0] | RR | | F4 | – | – | – | – | – | – | 2 | 8 |
| NOP | No operation | | | 0F | – | – | – | – | – | – | 1 | 2 |
| Flags Notation: | * = Value is a function of the result of the operation.<br>– = Unaffected<br>X = Undefined | | | | 0 = Reset to 0<br>1 = Set to 1 | | | | | | | |

# Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 26. Figure 27 and Figure 28 provide information about each of the eZ8 CPU instructions. Table 116 lists Opcode Map abbreviations.

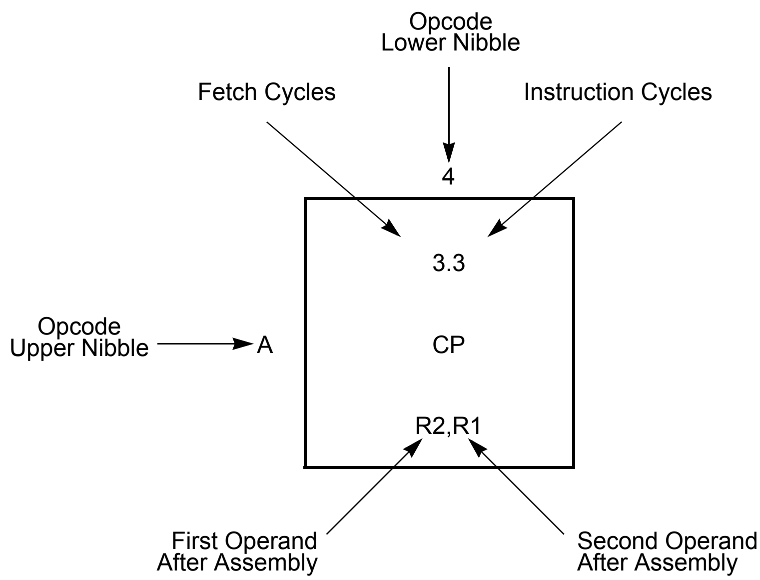


**Figure 26. Opcode Map Cell Description**

# On-Chip Peripheral AC and DC Electrical Characteristics

### Table 122. Power-On Reset and Voltage Brownout Electrical Characteristics and Timing

| Symbol | Parameter | $T_A$ = -40 °C to +105 °C | | | Units | Conditions |
|---|---|---|---|---|---|---|
| | | Minimum | Typical[1] | Maximum | | |
| $V_{POR}$ | Power-On Reset Voltage Threshold | 2.20 | 2.45 | 2.70 | V | $V_{DD} = V_{POR}$ |
| $V_{VBO}$ | Voltage Brownout Reset Voltage Threshold | 2.15 | 2.40 | 2.65 | V | $V_{DD} = V_{VBO}$ |
| | $V_{POR}$ to $V_{VBO}$ hysteresis | | 50 | 75 | mV | |
| | Starting $V_{DD}$ voltage to ensure valid Power-On Reset. | – | $V_{SS}$ | – | V | |
| $T_{ANA}$ | Power-On Reset Analog Delay | – | 70 | – | µs | $V_{DD} > V_{POR}$; $T_{POR}$ Digital Reset delay follows $T_{ANA}$ |
| $T_{POR}$ | Power-On Reset Digital Delay | | 16 | | µs | 66 Internal Precision Oscillator cycles + IPO startup time ($T_{IPOST}$) |
| $T_{SMR}$ | Stop Mode Recovery | | 16 | | µs | 66 Internal Precision Oscillator cycles |
| $T_{VBO}$ | Voltage Brownout Pulse Rejection Period | – | 10 | – | µs | Period of time in which $V_{DD} < V_{VBO}$ without generating a Reset. |
| $T_{RAMP}$ | Time for $V_{DD}$ to transition from $V_{SS}$ to $V_{POR}$ to ensure valid Reset | 0.10 | – | 100 | ms | |
| $T_{SMP}$ | Stop Mode Recovery pin pulse rejection period | | 20 | | ns | For any SMR pin or for the Reset pin when it is asserted in STOP mode. |

[1]Data in the typical column is from characterization at 3.3 V and 30 °C. These values are provided for design guidance only and are not tested in production.