



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

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

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

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

Details

| | |
|----------------------------|---|
| Product Status | 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 | 24 |
| Program Memory Size | 8KB (8K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f0813hj005eg |

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

| Date | Revision Level | Chapter/Section | Description | Page No. |
|----------|----------------|---|---|--|
| Sep | 15 | LED Drive Enable Register | Clarified statement surrounding the Alternate Function Register as it relates to the LED function; revised Flash Sector Protect Register description; revised Packaging chapter. | <u>51</u> , <u>144</u> , <u>210</u> |
| Mar 2008 | 14 | n/a | Changed branding to <i>Z8 Encore! XP F0823 Series</i> where appropriate. | All |
| Dec 2007 | 13 | Pin Description, General-Purpose Input/Output, Interrupt Controller, Watchdog Timer, Electrical Characteristics, and Ordering Information | Updated title from <i>Z8 Encore! 8K and 4K Series</i> to <i>Z8 Encore! XP Z8F0823 Series</i> . Updated Figure 3, Table 15, Table 35, Tables 59 through 61, Table 119 and Part Number Suffix Designations section. | <u>8</u> , <u>36</u> , <u>60</u> , <u>95</u> , <u>199</u> , and <u>220</u> |
| Aug 2007 | 12 | Part Selection Guide, External Clock Setup, and Program Memory | Updated Table 1, Table 16, and Program Memory section. | <u>2</u> , <u>35</u> , and <u>13</u> |
| Jun 2007 | 11 | n/a | Updated to combine Z8 Encore! 8K and Z8 Encore! 4K Series. | All |
| Dec 2006 | 10 | Ordering Information | Updated Ordering Information chapter. | <u>211</u> |

Table of Contents

| | |
|---|-----|
| Revision History | iii |
| List of Figures | x |
| List of Tables | xii |
| Overview | 1 |
| Features | 1 |
| Part Selection Guide | 2 |
| Block Diagram | 3 |
| CPU and Peripheral Overview | 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 | 5 |
| On-Chip Debugger | 6 |
| Pin Description | 7 |
| Available Packages | 7 |
| Pin Configurations | 7 |
| Signal Descriptions | 9 |
| Pin Characteristics | 11 |
| Address Space | 13 |
| Register File | 13 |
| Program Memory | 13 |
| Data Memory | 15 |
| Flash Information Area | 15 |
| Register Map | 16 |
| Reset and Stop Mode Recovery | 21 |
| Reset Types | 21 |
| Reset Sources | 23 |
| Power-On Reset | 23 |
| Voltage Brown-Out Reset | 24 |
| Watchdog Timer Reset | 25 |

► **Note:** *Analog input alternate functions (ANA) are not available on Z8F0x13 devices.

Signal Descriptions

Table 3 lists the Z8 Encore! XP F0823 Series signals. To determine the signals available for the specific package styles, see the [Pin Configurations](#) section on page 7.

Table 3. Signal Descriptions

| Signal Mnemonic | I/O | Description |
|--------------------------------------|-----|---|
| General-Purpose I/O Ports A–D | | |
| PA[7:0] | I/O | Port A. These pins are used for general-purpose I/O. |
| PB[7:0] ¹ | I/O | Port B. These pins are used for general-purpose I/O. PB6 and PB7 are available only in those devices without an ADC. |
| PC[7:0] | I/O | Port C. These pins are used for general-purpose I/O. |
| UART Controllers | | |
| TXD0 | O | Transmit Data. This signal is the transmit output from the UART and IrDA. |
| RXD0 | I | Receive Data. This signal is the receive input for the UART and IrDA. |
| CTS0 | I | Clear To Send. This signal is the flow control input for the UART. |
| DE | O | Driver Enable. This signal allows automatic control of external RS-485 drivers. This signal is approximately the inverse of the TXE (Transmit Empty) bit in the UART Status 0 Register. The DE signal can be used to ensure the external RS-485 driver is enabled when data is transmitted by the UART. |
| Timers | | |
| T0OUT/T1OUT | O | Timer Output 0–1. These signals are output from the timers. |
| T0OUT/T1OUT | O | Timer Complement Output 0–1. These signals are output from the timers in PWM DUAL OUTPUT Mode. |
| T0IN/T1IN | I | Timer Input 0–1. These signals are used as the capture, gating and counter inputs. The T0IN signal is multiplexed T0OUT signals. |
| Comparator | | |
| CINP/CINN | I | Comparator Inputs. These signals are the positive and negative inputs to the comparator. |

Notes:

1. PB6 and PB7 are only available in 28-pin packages without ADC. In 28-pin packages with ADC, they are replaced by AV_{DD} and AV_{SS}.
2. 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.

Table 3. Signal Descriptions (Continued)

| Signal Mnemonic | I/O | Description |
|-------------------------------|-----|---|
| COUT | O | Comparator Output. This is the output of the comparator. |
| 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. Caution: The DBG pin is open-drain and requires an external pull-up resistor to ensure proper operation. |
| Reset | | |
| 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} ² | I | Analog Power Supply. |
| V _{SS} | I | Digital Ground. |
| AV _{SS} | I | Analog Ground. |

Notes:

1. PB6 and PB7 are only available in 28-pin packages without ADC. In 28-pin packages with ADC, they are replaced by AV_{DD} and AV_{SS}.
2. 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.

Reset and Stop Mode Recovery

The Reset Controller within the Z8 Encore! XP F0823 Series controls Reset and Stop Mode Recovery operation and provides indication of low supply voltage conditions. In typical operation, the following events cause a Reset:

- Power-On Reset (POR)
- Voltage Brown-Out (VBO)
- Watchdog Timer time-out (when configured by the WDT_RES Flash Option Bit to initiate a reset)
- External $\overline{\text{RESET}}$ pin assertion (when the alternate RESET function is enabled by the GPIO register)
- On-chip Debugger initiated Reset (OCDCTL[0] set to 1)

When the device is in STOP Mode, a Stop Mode Recovery is initiated by either of the following:

- Watchdog Timer time-out
- GPIO port input pin transition on an enabled Stop Mode Recovery source

The VBO circuitry on the device performs the following function:

- Generates the VBO reset when the supply voltage drops below a minimum safe level

Reset Types

F0823 Series MCUs provide several different types of Reset operations. Stop Mode Recovery is considered a form of Reset. Table 9 lists the types of Reset and their operating characteristics. The duration of a System Reset is longer if the external crystal oscillator is enabled by the Flash option bits; this configuration allows additional time for oscillator startup.

Table 9. Reset and Stop Mode Recovery Characteristics and Latency

| Reset Characteristics and Latency | | | |
|-----------------------------------|--|---------|--|
| Reset Type | Control Registers | eZ8 CPU | Reset Latency (Delay) |
| System Reset | Reset (as applicable) | Reset | 66 Internal Precision Oscillator Cycles |
| Stop Mode Recovery | Unaffected, except WDT_CTL and OSC_CTL registers | Reset | 66 Internal Precision Oscillator Cycles + IPO startup time |

During a System Reset or Stop Mode Recovery, the IPO requires 4 μ s to start up. Then the Z8 Encore! XP F0823 Series device is held in Reset for 66 cycles of the Internal Precision Oscillator. If the crystal oscillator is enabled in the Flash option bits, this reset period is increased to 5000 IPO cycles. When a reset occurs because of a low voltage condition or Power-On Reset, this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the reset period, the device remains in reset until the pin is deasserted.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address.

When the control registers are re-initialized by a system reset, the system clock after reset is always the IPO. The software must reconfigure the oscillator control block, such that the correct system clock source is enabled and selected.

clock and reset signals, the required reset duration can be as short as three clock periods and as long as four. A reset pulse three clock cycles in duration might trigger a reset; a pulse four cycles in duration always triggers a reset.

While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z8 Encore! XP F0823 Series devices remain in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the System Reset time-out, the device exits the Reset state on the system clock rising edge following $\overline{\text{RESET}}$ pin deassertion. Following a System Reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the WDT Control (WDTCTL) register is set to 1.

External Reset Indicator

During System Reset or when enabled by the GPIO logic (see [the Port A–C Control Registers section on page 42](#)), the $\overline{\text{RESET}}$ pin functions as an open-drain (active Low) reset mode indicator in addition to the input functionality. This reset output feature allows an Z8 Encore! XP F0823 Series device to reset other components to which it is connected, even if that reset is caused by internal sources such as POR, VBO, or WDT events.

After an internal reset event occurs, the internal circuitry begins driving the $\overline{\text{RESET}}$ pin Low. The $\overline{\text{RESET}}$ pin is held Low by the internal circuitry until the appropriate delay listed in Table 9 has elapsed.

On-Chip Debugger Initiated Reset

A POR is initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The OCD block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the System Reset. Following the System Reset, the POR bit in the Reset Status (RSTSTAT) Register is set.

Stop Mode Recovery

The device enters into STOP Mode when eZ8 CPU executes a STOP instruction. For more details about STOP Mode, see [the Low-Power Modes section on page 30](#). During Stop Mode Recovery, the CPU is held in reset for 66 IPO cycles if the crystal oscillator is disabled or 5000 cycles if it is enabled. The SMR delay also included the time required to start up the IPO.

Stop Mode Recovery does not affect on-chip registers other than the Watchdog Timer Control Register (WDTCTL) and the Oscillator Control Register (OSCCTL). After any Stop Mode Recovery, the IPO is enabled and selected as the system clock. If another system clock source is required or IPO disabling is required, the Stop Mode Recovery code must reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

LED Drive Level Low Register

The LED Drive Level registers contain two control bits for each Port C pin (Table 34). These two bits select between four programmable drive levels. Each pin is individually programmable.

Table 34. LED Drive Level Low Register (LEDLVLL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------------|-----|-----|-----|-----|-----|-----|-----|
| Field | LEDLVLL[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 |
| Address | F84H | | | | | | | |

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

- Execution of an Return from Interrupt (IRET) instruction
- Writing a 1 to the IRQE bit in the Interrupt Control Register

Interrupts are globally disabled by any of the following actions:

- Execution of a Disable Interrupt (DI) instruction
- eZ8 CPU acknowledgement of an interrupt service request from the interrupt controller
- Writing a 0 to the IRQE bit in the Interrupt Control Register
- Reset
- Execution of a Trap instruction
- Illegal Instruction Trap
- Primary Oscillator Fail Trap
- Watchdog Timer Oscillator Fail Trap

Interrupt Vectors and Priority

The interrupt controller supports three levels of interrupt priority. Level 3 is the highest priority, Level 2 is the second highest priority, and Level 1 is the lowest priority. If all interrupts are enabled with identical interrupt priority (for example, all as Level 2 interrupts), the interrupt priority is assigned from highest to lowest as specified in [Table 35](#) on page 55. Level 3 interrupts are always assigned higher priority than Level 2 interrupts which, in turn, always are assigned higher priority than Level 1 interrupts. Within each interrupt priority level (Level 1, Level 2 or Level 3), priority is assigned as specified in [Table 35](#). Reset, Watchdog Timer interrupt (if enabled), Primary Oscillator Fail Trap, Watchdog Timer Oscillator Fail Trap, and Illegal Instruction Trap always have highest (Level 3) priority.

Interrupt Assertion

Interrupt sources assert their interrupt requests for only a single system clock period (single pulse). When the interrupt request is acknowledged by the eZ8 CPU, the corresponding bit in the Interrupt Request register is cleared until the next interrupt occurs. Writing a 0 to the corresponding bit in the Interrupt Request register likewise clears the interrupt request.

! Caution: Zilog recommends not using a coding style that clears bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 1, which follows.

two bits to configure timer interrupt definition, and a status bit to identify if the most recent timer interrupt is caused by an input capture event.

Table 57. Timer 0–1 Control Register 0 (TxCTL0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|----------|-----|-----|----------|------|-----|--------|
| Field | TMODEHI | TICONFIG | | | Reserved | PWMD | | INPCAP |
| 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 | F06H, F0EH | | | | | | | |

| Bit | Description |
|-------------------|---|
| [7] TMODEHI | Timer Mode High Bit This bit along with the TMODE field in TxCTL1 Register determines the operating mode of the timer. This is the most-significant bit of the Timer mode selection value. |
| [6:5] TICONFIG | Timer Interrupt Configuration This field configures timer interrupt definition. 0x = Timer Interrupt occurs on all defined reload, compare and input events. 10 = Timer Interrupt only on defined input capture/deassertion events. 11 = Timer Interrupt only on defined reload/compare events. |
| [4] | Reserved This bit is reserved and must be programmed to 0. |
| [3:1] PWMD | PWMD—PWM Delay value This field is a programmable delay to control the number of system clock cycles delay before the Timer Output and the Timer Output Complement are forced to their active state. 000 = No delay. 001 = 2 cycles delay. 010 = 4 cycles delay. 011 = 8 cycles delay. 100 = 16 cycles delay. 101 = 32 cycles delay. 110 = 64 cycles delay. 111 = 128 cycles delay. |
| [0] INPCAP | Input Capture Event This bit indicates if the most recent timer interrupt is caused by a Timer Input capture event. 0 = Previous timer interrupt is not a result of Timer Input capture event. 1 = Previous timer interrupt is a result of Timer Input capture event. |

Timer 0–1 Control Register 1

The Timer 0–1 Control (TxCTL1) registers enable/disable the timers, set the prescaler value, and determine the timer operating mode.

Watchdog Timer Reload High Byte Register (WDTH): see page 95

Watchdog Timer Reload Low Byte Register (WDTL): see page 95

Watchdog Timer Control Register

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

This register address is shared with the read-only Reset Status Register.

Table 60. 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] | Watchdog Timer Unlock |
| WDTUNLK | The software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer reload registers. |

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTH, WDTL) registers, shown in Tables 61 through 63, form the 24-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes. The 24-bit reload value ranges across bits [23:0] to encompass the three bytes {WDTU[7:0], WDTH[7:0], WDTL[7:0]}. Writing to these registers sets the appropriate Reload Value. Reading from these registers returns the current Watchdog Timer count value.

! **Caution:** The 24-bit WDT Reload Value must not be set to a value less than 000004H.

The UART data rate is calculated using the following equation:

$$\text{UART Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

For a given UART data rate, calculate the integer baud rate divisor value using the following equation:

$$\text{UART Baud Rate Divisor Value (BRG)} = \text{Round}\left(\frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Data Rate (bits/s)}}\right)$$

The baud rate error relative to the acceptable baud rate is calculated using the following equation:

$$\text{UART Baud Rate Error (\%)} = 100 \times \left(\frac{\text{Actual Data Rate} - \text{Desired Data Rate}}{\text{Desired Data Rate}} \right)$$

For reliable communication, the UART baud rate error must never exceed five percent. Table 73 provides information about data rate errors for a 5.5296MHz System Clock.

Table 73. UART Baud Rates

| 5.5296MHz System Clock | | | |
|-------------------------------|------------------------------|--------------------------|------------------|
| Acceptable Rate (kHz) | BRG Divisor (Decimal) | Actual Rate (kHz) | Error (%) |
| 1250.0 | N/A | N/A | N/A |
| 625.0 | N/A | N/A | N/A |
| 250.0 | 1 | 345.6 | 38.24 |
| 115.2 | 3 | 115.2 | 0.00 |
| 57.6 | 6 | 57.6 | 0.00 |
| 38.4 | 9 | 38.4 | 0.00 |
| 19.2 | 18 | 19.2 | 0.00 |
| 9.60 | 36 | 9.60 | 0.00 |
| 4.80 | 72 | 4.80 | 0.00 |
| 2.40 | 144 | 2.40 | 0.00 |
| 1.20 | 288 | 1.20 | 0.00 |
| 0.60 | 576 | 0.60 | 0.00 |
| 0.30 | 1152 | 0.30 | 0.00 |

The window remains open until the count again reaches 8 (that is, 24 baud clock periods since the previous pulse was detected), giving the endec a sampling window of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset, resynchronizing the endec to the incoming signal, allowing the endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All infrared endec configuration and status information is set by the UART control registers as defined in the Universal Asynchronous Receiver/Transmitter chapter on page 97.

! **Caution:** To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the endec before enabling the GPIO port alternate function for the corresponding pin.

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

Table 83. Flash Page Select Register (FPS)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---------|------|-----|-----|-----|-----|-----|-----|
| Field | INFO_EN | PAGE | | | | | | |
| 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 | FF9H | | | | | | | |

| Bit | Description |
|----------------|---|
| [7] INFO_EN | Information Area Enable 0 = Information Area is not selected. 1 = Information Area is selected. The Information Area is mapped into the Program Memory address space at addresses FE00H through FFFFH. |
| [6:0] PAGE | Page Select This 7-bit field identifies the Flash memory page for Page Erase and page unlocking. <ul style="list-style-type: none"> • Program Memory Address[15:9] = PAGE[6:0]. • For Z8F04x3 devices, the upper 4 bits must always be 0. • For Z8F02x3 devices, the upper 5 bits must always be 0. • For Z8F01x3 devices, the upper 6 bits must always be 0. |

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.

Assembly Language Source Program Example

```
JP START      ; Everything after the semicolon is a comment.
START:        ; A label called 'START'. The first instruction (JP START) in this
              ; example causes program execution to jump to the point within the
              ; program where the START label occurs.

LD R4, R7     ; A Load (LD) instruction with two operands. The first operand,
              ; Working Register R4, is the destination. The second operand,
              ; Working Register R7, is the source. The contents of R7 is
              ; written into R4.

LD 234H, #01  ; Another Load (LD) instruction with two operands.
              ; The first operand, Extended Mode Register Address 234H,
              ; identifies the destination. The second operand, Immediate Data
              ; value 01H, is the source. The value 01H is written into the
              ; Register at address 234H.
```

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as 'destination, source'. After assembly, the object code usually has the operands in the order 'source, destination', but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. You must follow this binary format if you prefer manual program coding or intend to implement your own assembler.

Example 1

If the contents of registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is shown in Table 106.

Table 106. Assembly Language Syntax Example 1

| | | | | |
|------------------------|-----|------|-----|----------------|
| Assembly Language Code | ADD | 43H, | 08H | (ADD dst, src) |
| Object Code | 04 | 08 | 43 | (OPC src, dst) |

Example 2

In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0–R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is shown in Table 107.

eZ8 CPU Instruction Classes

eZ8 CPU instructions are divided functionally into the following groups:

- Arithmetic
- Bit Manipulation
- Block Transfer
- CPU Control
- Load
- Logical
- Program Control
- Rotate and Shift

Tables 110 through 117 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 instruction 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 110. 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 |
| INCW | dst | Increment Word |

Table 110. Arithmetic Instructions (Continued)

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|---|
| MULT | dst | Multiply |
| SBC | dst, src | Subtract with Carry |
| SBCX | dst, src | Subtract with Carry using Extended Addressing |
| SUB | dst, src | Subtract |
| SUBX | dst, src | Subtract using Extended Addressing |

Table 111. Bit Manipulation Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|--|
| BCLR | bit, dst | Bit Clear |
| BIT | p, bit, dst | Bit Set or Clear |
| BSET | bit, dst | Bit Set |
| BSWAP | dst | Bit Swap |
| CCF | — | Complement Carry Flag |
| RCF | — | Reset Carry Flag |
| SCF | — | Set Carry Flag |
| TCM | dst, src | Test Complement Under Mask |
| TCMX | dst, src | Test Complement Under Mask using Extended Addressing |
| TM | dst, src | Test Under Mask |
| TMX | dst, src | Test Under Mask using Extended Addressing |

Table 112. Block Transfer Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|---|
| LDCI | dst, src | Load Constant to/from Program Memory and Auto-Increment Addresses |
| LDEI | dst, src | Load External Data to/from Data Memory and Auto-Increment Addresses |

Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 26. Figures 27 and 28 provide information about each of the eZ8 CPU instructions. Table 119 lists Opcode Map abbreviations.

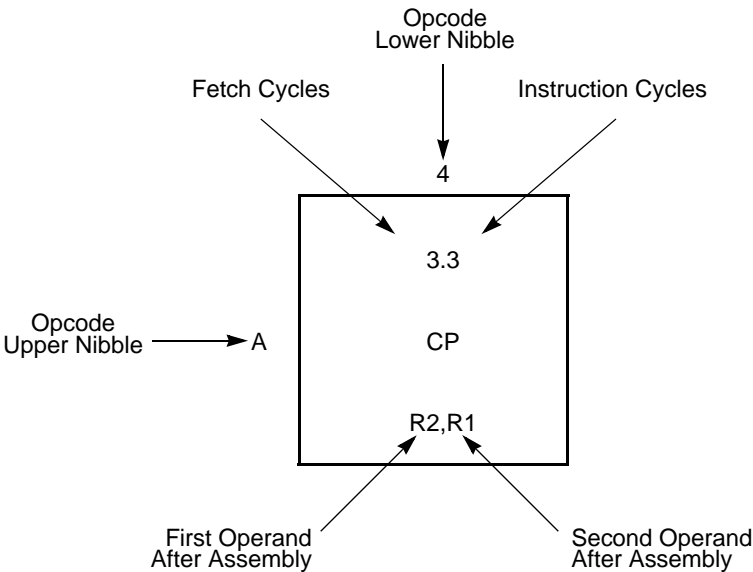


Figure 26. Opcode Map Cell Description