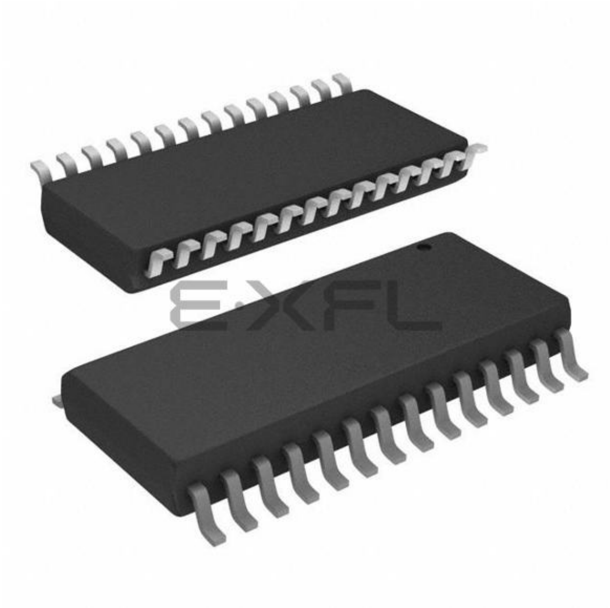

Welcome to E-XFL.COM

## 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 | 24 |
| Program Memory Size | 4KB (4K 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 | 0°C ~ 70°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f0413sj005sg |

## Block Diagram

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

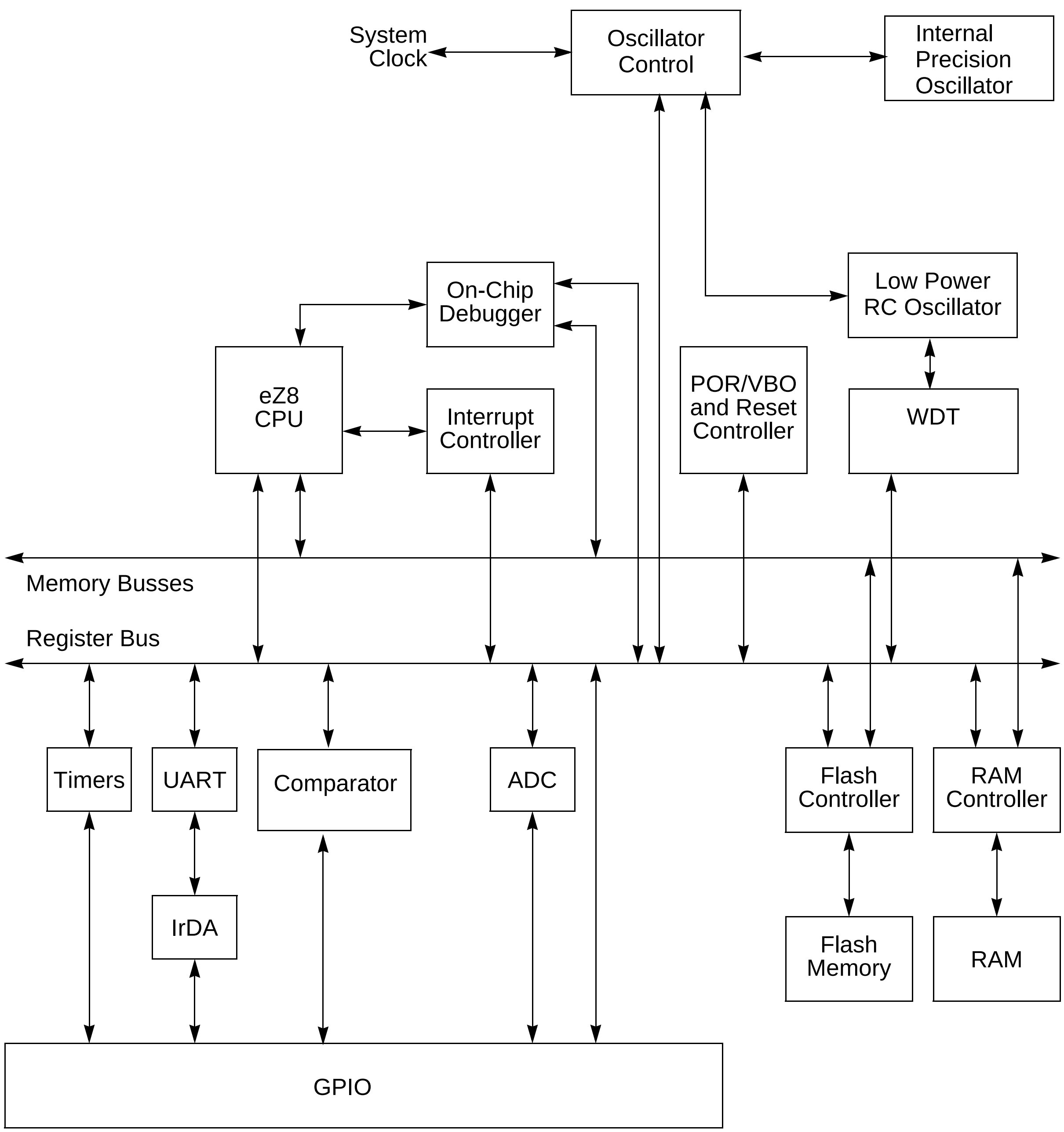


**Figure 1. Z8 Encore! XP F0823 Series Block Diagram**

| Bit | Description (Continued) |
|---|---|
| [4]<br>EXT | **External Reset Indicator**<br>If this bit is set to 1, a Reset initiated by the external $\overline{\text{RESET}}$ pin occurred. A Power-On Reset or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit. For POR/Stop Mode Recover event values, please see Table 13. |
| [3:0] | **Reserved**<br>These bits are reserved and must be programmed to 0000 when read. |

**Table 13. POR Indicator Values**

| Reset or Stop Mode Recovery Event | POR | STOP | WDT | EXT |
|---|---|---|---|---|
| Power-On Reset | 1 | 0 | 0 | 0 |
| Reset using $\overline{\text{RESET}}$ pin assertion | 0 | 0 | 0 | 1 |
| Reset using WDT time-out | 0 | 0 | 1 | 0 |
| Reset using the OCD (OCTCTL[1] set to 1) | 1 | 0 | 0 | 0 |
| Reset from STOP Mode using DBG Pin driven Low | 1 | 0 | 0 | 0 |
| Stop Mode Recovery using GPIO pin transition | 0 | 1 | 0 | 0 |
| Stop Mode Recovery using WDT time-out | 0 | 1 | 1 | 0 |

PA0 and PA6 contain two different timer functions, a timer input and a complementary timer output. Both of these functions require the same GPIO configuration, the selection between the two is based on the timer mode. For more details, see the Timers chapter on page 69.

! **Caution:** For pins with multiple alternate functions, Zilog recommends writing to the AFS1 and AFS2 subregisters before enabling the alternate function via the AF Subregister. This prevents spurious transitions through unwanted alternate function modes.

**Table 16. Port Alternate Function Mapping (8-Pin Parts)**

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Select Register AFS1 | Alternate Function Select Register AFS2 |
|---|---|---|---|---|---|
| **Port A** | PA0 | T0IN | Timer 0 Input | AFS1[0]: 0 | AFS2[0]: 0 |
| | | Reserved | | AFS1[0]: 0 | AFS2[0]: 1 |
| | | Reserved | | AFS1[0]: 1 | AFS2[0]: 0 |
| | | T0OUT | Timer 0 Output Complement | AFS1[0]: 1 | AFS2[0]: 1 |
| | PA1 | T0OUT | Timer 0 Output | AFS1[1]: 0 | AFS2[1]: 0 |
| | | Reserved | | AFS1[1]: 0 | AFS2[1]: 1 |
| | | CLKIN | External Clock Input | AFS1[1]: 1 | AFS2[1]: 0 |
| | | Analog Functions* | ADC Analog Input/$V_{REF}$ | AFS1[1]: 1 | AFS2[1]: 1 |
| | PA2 | DE0 | UART 0 Driver Enable | AFS1[2]: 0 | AFS2[2]: 0 |
| | | RESET | External Reset | AFS1[2]: 0 | AFS2[2]: 1 |
| | | T1OUT | Timer 1 Output | AFS1[2]: 1 | AFS2[2]: 0 |
| | | Reserved | | AFS1[2]: 1 | AFS2[2]: 1 |
| | PA3 | CTS0 | UART 0 Clear to Send | AFS1[3]: 0 | AFS2[3]: 0 |
| | | COUT | Comparator Output | AFS1[3]: 0 | AFS2[3]: 1 |
| | | T1IN | Timer 1 Input | AFS1[3]: 1 | AFS2[3]: 0 |
| | | Analog Functions* | ADC Analog Input | AFS1[3]: 1 | AFS2[3]: 1 |
| | PA4 | RXD0 | UART 0 Receive Data | AFS1[4]: 0 | AFS2[4]: 0 |
| | | Reserved | | AFS1[4]: 0 | AFS2[4]: 1 |
| | | Reserved | | AFS1[4]: 1 | AFS2[4]: 0 |
| | | Analog Functions* | ADC/Comparator Input (N) | AFS1[4]: 1 | AFS2[4]: 1 |
| | PA5 | TXD0 | UART 0 Transmit Data | AFS1[5]: 0 | AFS2[5]: 0 |
| | | T1OUT | Timer 1 Output Complement | AFS1[5]: 0 | AFS2[5]: 1 |
| | | Reserved | | AFS1[5]: 1 | AFS2[5]: 0 |
| | | Analog Functions* | ADC/Comparator Input (P) | AFS1[5]: 1 | AFS2[5]: 1 |

Note: *Analog Functions include ADC inputs, ADC reference and comparator inputs. Also, alternate function selection as described in the Port A–C Alternate Function Subregisters section on page 43 must be enabled.

**Table 17. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)**

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Set Register AFS1 |
|---|---|---|---|---|
| **Port B[3]** | PB03 | 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 |
| | | $V_{REF}$[4] | ADC Voltage Reference | AFS1[5]: 1 |
| | PB6 | Reserved | | AFS1[6]: 0 |
| | | Reserved | | AFS1[6]: 1 |
| | PB7 | Reserved | | AFS1[7]: 0 |
| | | Reserved | | AFS1[7]: 1 |

Notes:
1. Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in the Port A–C Alternate Function Subregisters section on page 43 automatically enables the associated alternate function.
2. Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in the Timer Pin Signal Operation section on page 83.
3. 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 the Port A–C Alternate Function Subregisters section on page 43 must also be enabled.
4. $V_{REF}$ is available on PB5 in 28-pin products only.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, Alternate Function selection as described in the Port A–C Alternate Function Subregisters section on page 43 must also be enabled.
6. $V_{REF}$ is available on PC2 in 20-pin parts only.

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:

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

## COMPARE Mode

In COMPARE Mode, the timer counts up to the 16-bit maximum compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the compare value, the timer generates an interrupt and counting continues (the timer value is not reset to `0001H`). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) upon compare.

If the Timer reaches `FFFFH`, the timer rolls over to `0000H` and continue counting. Observe the following steps to configure a timer for COMPARE Mode and to initiate the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for COMPARE Mode
   - Set the prescale value
   - Set the initial logic level (High or Low) for the Timer Output alternate function, if appropriate
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the compare value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers.

**Table 53. Timer 0–1 Reload High Byte Register (TxRH)**

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

**Table 54. Timer 0–1 Reload Low Byte Register (TxRL)**

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

| Bit | Description |
|---|---|
| [7] | |
| [6] | |
| [5] | |
| [4] | |
| [3] | |
| [2] | |
| [1] | |
| [0] | |

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

## Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers (Table 55 and Table 56) control pulse-width modulator (PWM) operations. These registers also store the capture values for the CAPTURE and CAPTURE/COMPARE modes.

**Table 55. Timer 0–1 PWM High Byte Register (TxPWMH)**

| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| **Field** | PWMH | | | | | | | |
| **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** | F04H, F0CH | | | | | | | |

**Table 56. Timer 0–1 PWM Low Byte Register (TxPWML)**

| **Bit** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|
| **Field** | PWML | | | | | | | |
| **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** | F05H, F0DH | | | | | | | |

| **Bit** | **Description** |
|---|---|
| [7:0]<br>PWMH,<br>PWML | **Pulse-Width Modulator High and Low Bytes**<br>These two bytes, {PWMH[7:0], PWML[7:0]}, form a 16-bit value that is compared to the current 16-bit timer count. When a match occurs, the PWM output changes state. The PWM output value is set by the TPOL bit in the Timer Control Register (TxCTL1) register.<br><br>These TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in CAPTURE or CAPTURE/COMPARE modes. |

## Timer 0–1 Control Registers

The Timer Control registers are 8-bit read/write registers that control the operation of their associated counter/timers.

### Timer 0–1 Control Register 0

The Timer Control Register 0 (TxCTL0) and Timer Control Register 1 (TxCTL1) determine the timer operating mode. It also includes a programmable PWM deadband delay, 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]<br>TMODEHI | **Timer Mode High Bit**<br>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]<br>TICONFIG | **Timer Interrupt Configuration**<br>This field configures timer interrupt definition.<br>0x = Timer Interrupt occurs on all defined reload, compare and input events.<br>10 = Timer Interrupt only on defined input capture/deassertion events.<br>11 = Timer Interrupt only on defined reload/compare events. |
| [4] | **Reserved**<br>This bit is reserved and must be programmed to 0. |
| [3:1]<br>PWMD | **PWMD—PWM Delay value**<br>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.<br>000 = No delay.<br>001 = 2 cycles delay.<br>010 = 4 cycles delay.<br>011 = 8 cycles delay.<br>100 = 16 cycles delay.<br>101 = 32 cycles delay.<br>110 = 64 cycles delay.<br>111 = 128 cycles delay. |
| [0]<br>INPCAP | **Input Capture Event**<br>This bit indicates if the most recent timer interrupt is caused by a Timer Input capture event.<br>0 = Previous timer interrupt is not a result of Timer Input capture event.<br>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.

occurred, this byte cannot contain valid data and must be ignored. The BRKD bit indicates if the overrun was caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data Register must be read again to clear the error bits is the UART Status 0 Register. Updates to the Receive Data Register occur only when the next data word is received.

## UART Data and Error Handling Procedure

Figure 15 displays the recommended procedure for use in UART receiver interrupt service routines.

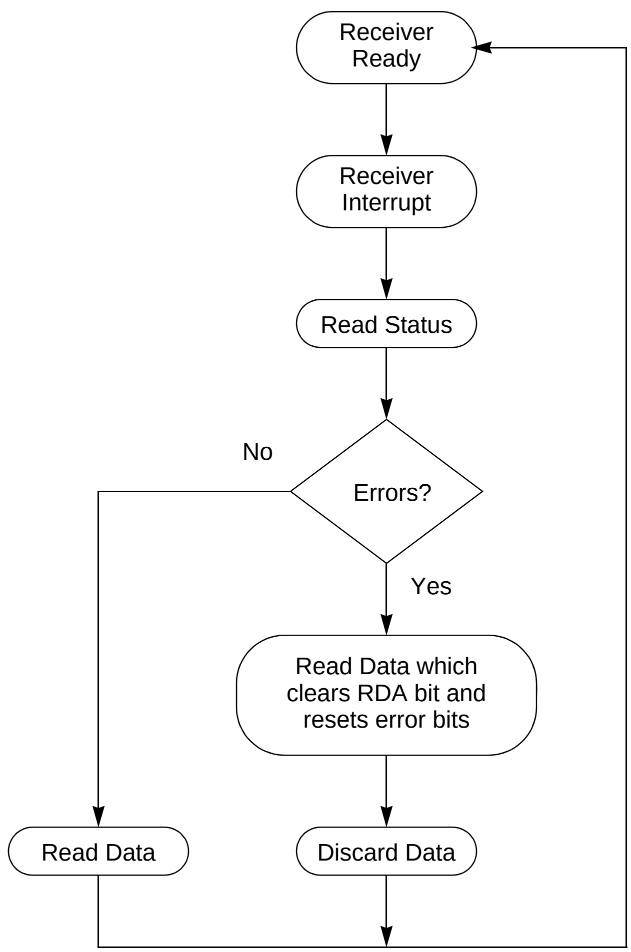


**Figure 15. UART Receiver Interrupt Service Routine Flow**

  - `CEN` resets to 0 to indicate the conversion is complete

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

## Continuous Conversion

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

---

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

---

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

1. Enable the acceptable analog input by configuring the general-purpose I/O pins for alternate function. This action disables the digital input and output driver.
2. Write the ADC Control/Status Register 1 to configure the ADC:
   - Write the `REFSELH` bit of the pair {`REFSELH`, `REFSELL`} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in the ADC Control/Status Register 1.

3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control Register can be written simultaneously:
   - Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
   - Set `CONT` to 1 to select continuous conversion.
   - If the internal VREF must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
   - Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in ADC Control Register 0.
   - Set `CEN` to 1 to start the conversions.

## 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 `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 84. Flash Sector Protect Register (FPROT)**

| **Bit** | **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 |
| **Address** | FF9H | | | | | | | |

| **Bit** | **Description** |
|---|---|
| [7]<br>SPROT*n* | **Sector Protection**<br>Each bit corresponds to a 1024-byte Flash sector on devices in the 8K range, while the remaining devices correspond to a 512-byte Flash sector. To determine the appropriate Flash memory sector address range and sector number for your Z8F0823 Series product, please refer to Table 79 on page 134 and to Figure 20, which follows the table.<br>• For Z8F08x3 and Z8F04x3 devices, all bits are used.<br>• For Z8F02x3 devices, the upper 4 bits are unused.<br>• For Z8F01x3 devices, the upper 6 bits are unused. |

Note: *n* indicates the specific Flash sector (7–0).

## 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:

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

## Serialization Data

**Table 96. Serial Number at 001C–001F (S_NUM)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | S_NUM | | | | | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Information Page Memory 001C–001F | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>S_NUM | **Serial Number Byte**<br>The serial number is a unique four-byte binary value; see Table 97. |

**Table 97. Serialization Data Locations**

| Info Page Address | Memory Address | Usage |
|---|---|---|
| 1C | FE1C | Serial Number Byte 3 (most significant). |
| 1D | FE1D | Serial Number Byte 2. |
| 1E | FE1E | Serial Number Byte 1. |
| 1F | FE1F | Serial Number Byte 0 (least significant). |

## Randomized Lot Identifier

**Table 98. Lot Identification Number (RAND_LOT)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | RAND_LOT | | | | | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Interspersed throughout Information Page Memory | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|---|---|
| [7]<br>RAND_LOT | **Randomized Lot ID**<br>The randomized lot ID is a 32-byte binary value that changes for each production lot; see Table 99. |

- If the PA2/$\overline{\text{RESET}}$ pin is held Low while a 32-bit key sequence is issued to the PA0/DBG pin, the DBG feature is unlocked. After releasing PA2/$\overline{\text{RESET}}$, it is pulled high. At this point, the PA0/DBG pin can be used to autobaud and cause the device to enter DEBUG Mode. For more details, see the OCD Unlock Sequence (8-Pin Devices Only) section on page 161.

### Exiting DEBUG Mode

The device exits DEBUG Mode following any of these operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset
- 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 as displayed in Figure 25.

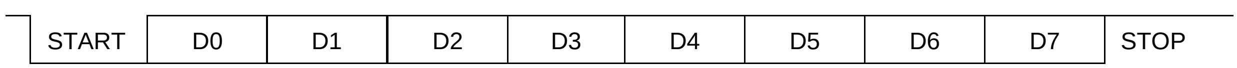


**Figure 25. OCD Data Format**

> **Note:** When responding to a request for data, the OCD may commence transmitting immediately after receiving the stop bit of an incoming frame. Therefore, when sending the stop bit, the host must not actively drive the DBG pin High for more than 0.5 bit times. Zilog recommends that, if possible, the host drives the DBG pin using an open-drain output.

## OCD Autobaud Detector/Generator

To run over a range of baud rates (data bits per second) with various system clock frequencies, the OCD contains an auto-baud 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

In the following list of OCD Commands, data and commands sent from the host to the OCD are identified by `'DBG ← 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 OCD. 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 up 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 and the only method of returning the device to normal operating mode is to reset the device.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

**Read OCD Control Register (05H).** The Read OCD Control Register command reads the value of the OCDCTL register.

```
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]
```

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

| Assembly Mnemonic | Symbolic Operation | Address Mode dst | Address Mode src | Opcode(s) (Hex) | Flags C | Flags Z | Flags S | Flags V | Flags D | Flags H | Fetch Cycles | Instr. Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BTJNZ bit, src, dst | if src[bit] = 1<br>PC ← PC + X | | r | F6 | – | – | – | – | – | – | 3 | 3 |
| | | | Ir | F7 | | | | | | | 3 | 4 |
| BTJZ bit, src, dst | if src[bit] = 0<br>PC ← PC + X | | r | F6 | – | – | – | – | – | – | 3 | 3 |
| | | | Ir | F7 | | | | | | | 3 | 4 |
| CALL dst | SP ← SP –2<br>@SP ← PC<br>PC ← dst | IRR | | D4 | – | – | – | – | – | – | 2 | 6 |
| | | DA | | D6 | | | | | | | 3 | 3 |
| CCF | C ← ~C | | | EF | * | – | – | – | – | –- | 1 | 2 |
| CLR dst | dst ← 00H | R | | B0 | – | – | – | – | – | – | 2 | 2 |
| | | IR | | B1 | | | | | | | 2 | 3 |
| COM dst | dst ← ~dst | R | | 60 | – | * | * | 0 | – | – | 2 | 2 |
| | | IR | | 61 | | | | | | | 2 | 3 |
| CP dst, src | dst - src | r | r | A2 | * | * | * | * | – | – | 2 | 3 |
| | | r | Ir | A3 | | | | | | | 2 | 4 |
| | | R | R | A4 | | | | | | | 3 | 3 |
| | | R | IR | A5 | | | | | | | 3 | 4 |
| | | R | IM | A6 | | | | | | | 3 | 3 |
| | | IR | IM | A7 | | | | | | | 3 | 4 |
| CPC dst, src | dst - src - C | r | r | 1F A2 | * | * | * | * | – | – | 3 | 3 |
| | | r | Ir | 1F A3 | | | | | | | 3 | 4 |
| | | R | R | 1F A4 | | | | | | | 4 | 3 |
| | | R | IR | 1F A5 | | | | | | | 4 | 4 |
| | | R | IM | 1F A6 | | | | | | | 4 | 3 |
| | | IR | IM | 1F A7 | | | | | | | 4 | 4 |
| CPCX dst, src | dst - src - C | ER | ER | 1F A8 | * | * | * | * | – | – | 5 | 3 |
| | | ER | IM | 1F A9 | | | | | | | 5 | 3 |
| CPX dst, src | dst - src | ER | ER | A8 | * | * | * | * | – | – | 4 | 3 |
| | | ER | IM | A9 | | | | | | | 4 | 3 |

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

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

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

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

# 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.

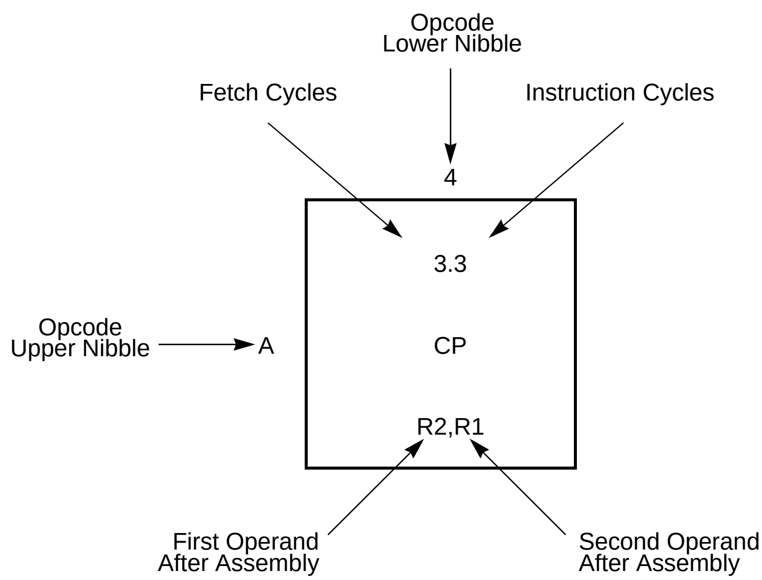


**Figure 26. Opcode Map Cell Description**

**Table 128. Analog-to-Digital Converter Electrical Characteristics and Timing**

| | | $V_{DD}$ = 3.0V to 3.6V<br>$T_A$ = 0°C to +70°C<br>(unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Minimum** | **Typical** | **Maximum** | **Units** | **Conditions** |
| | Resolution | 10 | | – | bits | |
| | Differential Nonlinearity (DNL) | –1.0 | – | 1.0 | LSB[3] | External $V_{REF}$ = 2.0V; $R_S$ ← 3.0 kΩ |
| | Integral Nonlinearity (INL) | –3.0 | – | 3.0 | LSB[3] | External $V_{REF}$ = 2.0V; $R_S$ ← 3.0 kΩ |
| | Offset Error with Calibration | | ±1 | | LSB[3] | |
| | Absolute Accuracy with Calibration | | ±3 | | LSB[3] | |
| $V_{REF}$ | Internal Reference Voltage | 1.0<br>2.0 | 1.1<br>2.2 | 1.2<br>2.4 | V | REFSEL=01<br>REFSEL=10 |
| $V_{REF}$ | Internal Reference Variation with Temperature | | ±1.0 | | % | Temperature variation with $V_{DD}$ = 3.0 |
| $V_{REF}$ | Internal Reference Voltage Variation with $V_{DD}$ | | ±0.5 | | % | Supply voltage variation with $T_A$ = 30°C |
| $R_{REFOUT}$ | Reference Buffer Output Impedance | | 850 | | W | When the internal reference is buffered and driven out to the VREF pin (REFOUT = 1) |
| | Single-Shot Conversion Time | – | 5129 | – | System clock cycles | All measurements but temperature sensor |
| | | | 10258 | | | Temperature sensor measurement |
| | Continuous Conversion Time | – | 256 | – | System clock cycles | All measurements but temperature sensor |
| | | | 512 | | | Temperature sensor measurement |

Notes:
1. Analog source impedance affects the ADC offset voltage (because of pin leakage) and input settling time.
2. Devices are factory calibrated at $V_{DD}$ = 3.3 V and $T_A$ = +30°C, so the ADC is maximally accurate under these conditions.
3. LSBs are defined assuming 10-bit resolution.
4. This is the maximum recommended resistance seen by the ADC input pin.
5. The input impedance is inversely proportional to the system clock frequency.

available to the eZ8 CPU on the second rising clock edge following the change of the port value.

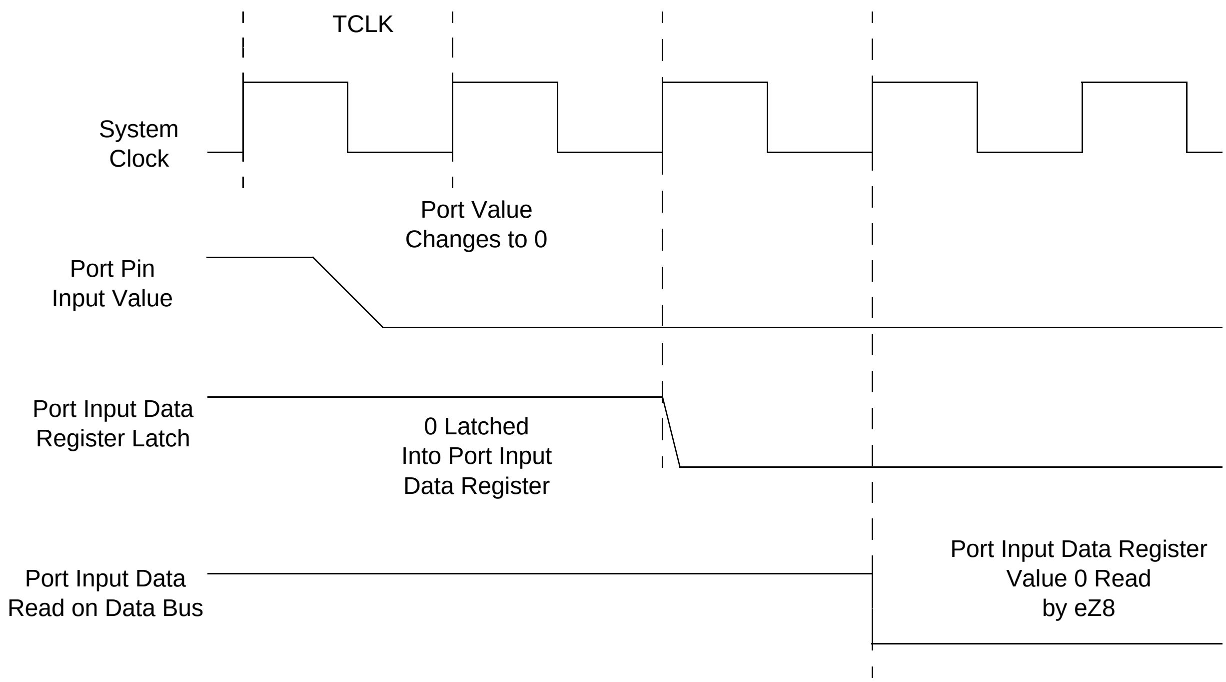


**Figure 29. Port Input Sample Timing**

**Table 130. GPIO Port Input Timing**

| | | Delay (ns) | |
|---|---|---|---|
| **Parameter** | **Abbreviation** | **Minimum** | **Maximum** |
| $T_{S\_PORT}$ | Port Input Transition to $X_{IN}$ Rise Setup Time (Not pictured) | 5 | – |
| $T_{H\_PORT}$ | $X_{IN}$ Rise to Port Input Transition Hold Time (Not pictured) | 0 | – |
| $T_{SMR}$ | GPIO Port Pin Pulse Width to ensure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources) | 1 µs | |

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

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|---|---|---|---|---|---|---|---|
| **Z8 Encore! XP F0823 Series with 4 KB Flash, 10-Bit Analog-to-Digital Converter** | | | | | | | | |
| **Standard Temperature: 0°C to 70°C** | | | | | | | | |
| Z8F0423PB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0423QB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0423SB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0423SH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0423HH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0423PH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0423SJ005SG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0423HJ005SG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0423PJ005SG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |
| **Extended Temperature: –40°C to 105°C** | | | | | | | | |
| Z8F0423PB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0423QB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0423SB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0423SH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0423HH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0423PH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0423SJ005EG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0423HJ005EG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0423PJ005EG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |