**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 | 20MHz |
| Connectivity | I²C, IrDA, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 19 |
| 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 | A/D 5x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°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/z8f0822sj020ec |

# Introduction

Zilog's Z8 Encore! XP® MCU product family is a line of Zilog microcontrollers based on the 8-bit eZ8 CPU. Z8 Encore! XP® F0822 Series, hereafter referred as Z8 Encore! XP or the 8K Series adds Flash memory to Zilog's extensive line of 8-bit microcontrollers. The Flash in-circuit programming allows faster development time and program changes in the field. The new eZ8 CPU is upward-compatible with the existing Z8® instructions. The rich peripheral set of Z8 Encore! XP makes it suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

## Features

The features of Z8 Encore! XP MCU product family include:

- 20 MHz eZ8 CPU core
- Up to 8 KB Flash with in-circuit programming capability
- 1 KB Register RAM
- Optional 2- to 5-channel, 10-bit Analog-to-Digital Converter (ADC)
- Full-duplex 9-bit Universal Asynchronous Receiver/Transmitter (UART) with bus transceiver Driver Enable Control
- Inter-Integrated Circuit ($I^2C$)
- Serial Peripheral Interface (SPI)
- Infrared Data Association (IrDA)-compliant infrared encoder/decoders
- Two 16-bit timers with Capture, Compare, and PWM capability
- Watchdog Timer (WDT) with internal RC oscillator
- 11 to 19 Input/Output pins depending upon package
- Up to 19 interrupts with configurable priority
- On-Chip Debugger (OCD)
- Voltage Brownout (VBO) protection
- Power-On Reset (POR)
- Crystal oscillator with three power settings and RC oscillator option

# Signal and Pin Descriptions

Z8 Encore! XP® F0822 Series products are available in a variety of packages, styles, and pin configurations. This chapter describes the signals and available pin configurations for each of the package styles. For information regarding the physical package specifications, see Packaging on page 233.

## Available Packages

Table 2 identifies the package styles available for each device within Z8 Encore! XP F0822 Series product line.

**Table 2. Z8 Encore! XP F0822 Series Package Options**

| Part Number | 10-Bit ADC | 20-Pin SSOP and PDIP | 28-Pin SOIC and PDIP |
|---|---|---|---|
| Z8F0822 | Yes | | X |
| Z8F0821 | Yes | X | |
| Z8F0812 | No | | X |
| Z8F0811 | No | X | |
| Z8F0422 | Yes | | X |
| Z8F0421 | Yes | X | |
| Z8F0412 | No | | X |
| Z8F0411 | No | X | |

## Pin Configurations

Figure 2 through Figure 5 display the pin configurations for all of the packages available in Z8 Encore! XP F0822 Series. See Table 4 for a description of the signals.

**Note:** *The analog input alternate functions (ANAx) are not available on Z8 Encore! XP® F0822 Series devices.*

## Pin Characteristics

Table 4 provides detailed information on the characteristics for each pin available on Z8 Encore! XP® F0822 Series products. Table 4 data is sorted alphabetically by the pin symbol mnemonic.

**Table 4. Pin Characteristics**

| Symbol Mnemonic | Direction | Reset Direction | Active Low or Active High | Tri-State Output | Internal Pull-up or Pull-down | Schmitt-Trigger Input | Open Drain Output |
|---|---|---|---|---|---|---|---|
| $AV_{DD}$ | N/A | N/A | N/A | N/A | No | No | N/A |
| $AV_{SS}$ | N/A | N/A | N/A | N/A | No | No | N/A |
| DBG | I/O | I | N/A | Yes | No | Yes | Yes |
| PA[7:0] | I/O | I | N/A | Yes | Programmable Pull-up | Yes | Yes, Programmable |
| PB[4:0] | I/O | I | N/A | Yes | Programmable Pull-up | Yes | Yes, Programmable |
| PC[5:0] | I/O | I | N/A | Yes | Programmable Pull-up | Yes | Yes, Programmable |
| RESET | I | I | Low | N/A | Pull-up | Yes | N/A |
| $V_{DD}$ | N/A | N/A | N/A | N/A | No | No | N/A |
| VREF | Analog | N/A | N/A | N/A | No | No | N/A |
| $V_{SS}$ | N/A | N/A | N/A | N/A | No | No | N/A |
| XIN | I | I | N/A | N/A | No | No | N/A |
| XOUT | O | O | N/A | No | No | No | No |

### System Reset

During a System Reset, a Z8 Encore! XP® F0822 Series device is held in Reset for 66 cycles of the WDT oscillator followed by 16 cycles of the system clock. At the beginning of Reset, all GPIO pins are configured as inputs. All GPIO programmable pull-ups are disabled.

During Reset, the eZ8 CPU and the on-chip peripherals are idle; however, the on-chip crystal oscillator and WDT oscillator continue to run. The system clock begins operating following the WDT oscillator cycle count. The eZ8 CPU and on-chip peripherals remain idle through all the 16 cycles of the system clock.

Upon Reset, control registers within the Register File which 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 the 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.

## Reset Sources

Table 9 lists the reset sources as a function of the operating mode. The text following provides more detailed information on the individual reset sources.

**Note:** *A POR/VBO event always has priority over all other possible reset sources to insure a full system reset occurs.*

**Table 9. Reset Sources and Resulting Reset Type**

| Operating Mode | Reset Source | Reset Type |
|---|---|---|
| NORMAL or HALT modes | POR/VBO | System Reset |
| | WDT time-out when configured for Reset | System Reset |
| | $\overline{\text{RESET}}$ pin assertion | System Reset |
| | OCD initiated Reset (OCDCTL[0] set to 1) | System Reset except the OCD is unaffected by the reset |
| STOP mode | POR/ VBO | System Reset |
| | $\overline{\text{RESET}}$ pin assertion | System Reset |
| | DBG pin driven Low | System Reset |

### Stop Mode Recovery Using WDT Time-Out

If the WDT times out during STOP mode, the device undergoes a Stop Mode Recovery sequence. In the WDT Control Register, the WDT and STOP bits are set to 1. If the WDT is configured to generate an interrupt upon time-out and the Z8 Encore! XP® F0822 Series device is configured to respond to interrupts, the eZ8 CPU services the WDT interrupt request following the normal Stop Mode Recovery sequence.

### Stop Mode Recovery Using a GPIO Port Pin Transition

Each of the GPIO Port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a STOP Mode Recover source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. The GPIO Stop Mode Recovery signals are filtered to reject pulses less than 10 ns (typical) in duration. In the WDT Control Register, the STOP bit is set to 1.

**Caution:** *In STOP mode, the GPIO Port Input Data Registers (PxIN) are disabled. The Port Input Data Registers record the Port transition only if the signal stays on the Port pin through the end of the Stop Mode Recovery delay. Therefore, short pulses on the Port pin initiates Stop Mode Recovery without being written to the Port Input Data Register or without initiating an interrupt (if enabled for that pin).*

**Table 24. Interrupt Vectors in Order of Priority (Continued)**

| Priority | Program Memory Vector Address | Interrupt Source |
|---|---|---|
| | 0008H | Reserved |
| | 000AH | Timer 1 |
| | 000CH | Timer 0 |
| | 000EH | UART 0 receiver |
| | 0010H | UART 0 transmitter |
| | 0012H | $I^2C$ |
| | 0014H | SPI |
| | 0016H | ADC |
| | 0018H | Port A7, rising or falling input edge |
| | 001AH | Port A6, rising or falling input edge |
| | 001CH | Port A5, rising or falling input edge |
| | 001EH | Port A4, rising or falling input edge |
| | 0020H | Port A3, rising or falling input edge |
| | 0022H | Port A2, rising or falling input edge |
| | 0024H | Port A1, rising or falling input edge |
| | 0026H | Port A0, rising or falling input edge |
| | 0028H | Reserved |
| | 002AH | Reserved |
| | 002CH | Reserved |
| | 002EH | Reserved |
| | 0030H | Port C3, both input edges |
| | 0032H | Port C2, both input edges |
| | 0034H | Port C1, both input edges |
| **Lowest** | 0036H | Port C0, both input edges |

3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the Timer Reload High and Low Byte Registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.
5. If required, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the Timer Output alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

The PWM period is given by the following equation.

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

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

If TPOL is set to 0, the ratio of the PWM output High time to the total period is given by

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

If TPOL is set to 1, the ratio of the PWM output High time to the total period is given by

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

### CAPTURE Mode

In CAPTURE mode, the current timer count value is recorded when the desired external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the timer continues counting.

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.

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

### CAPTURE/COMPARE Mode

In CAPTURE/COMPARE mode, the timer begins counting on the *first* external Timer Input transition. The required transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the system clock.

Every subsequent desired transition (after the first) of the Timer Input signal captures the current count value. The Capture value is written to the Timer PWM High and Low Byte Registers. When the Capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte Registers is reset to 0001H and counting resumes.

If no Capture event occurs, the timer counts up to the 16-bit Compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the Compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte Registers is reset to 0001H and counting resumes.

Follow the steps below for configuring a timer for CAPTURE/COMPARE mode and initiating the count:

1. Write to the Timer Control Register to:
   – Disable the timer
   – Configure the timer for CAPTURE/COMPARE 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 Compare value
4. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers
5. Configure the associated GPIO port pin for the Timer Input alternate function
6. Write to the Timer Control Register to enable the timer
7. Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge

In CAPTURE/COMPARE mode, the elapsed time from timer start to Capture event is calculated using the following equation:

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

## Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte

### WDT Reset in Normal Operation

If configured to generate a Reset when a time-out occurs, the WDT forces the device into the Reset state. The WDT status bit in the WDT Control Register is set to 1. For more information on Reset, see Reset and Stop Mode Recovery on page 39.

### WDT Reset in STOP Mode

If enabled in STOP mode and configured to generate a Reset when a time-out occurs and the device is in STOP mode, the WDT initiates a Stop Mode Recovery. Both the WDT status bit and the STOP bit in the WDT Control Register is set to 1 following WDT time-out in STOP mode. For more information on Reset, see Reset and Stop Mode Recovery on page 39. Default operation is for the WDT and its RC oscillator to be enabled during STOP mode.

### WDT RC Disable in STOP Mode

To minimize power consumption in STOP mode, the WDT and its RC oscillator can be disabled in STOP mode. The following sequence configures the WDT to be disabled when the Z8F082x family device enters STOP mode following execution of a STOP instruction:

1. Write 55H to the Watchdog Timer Control Register (WDTCTL).
2. Write AAH to the Watchdog Timer Control Register (WDTCTL).
3. Write 81H to the Watchdog Timer Control Register (WDTCTL) to configure the WDT and its oscillator to be disabled during STOP mode. Alternatively, write 00H to the Watchdog Timer Control Register (WDTCTL) as the third step in this sequence to reconfigure the WDT and its oscillator to be enabled during STOP mode. This sequence only affects WDT operation in STOP mode.

## Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the WDTCTL 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 address produce no effect on the bits in the WDTCTL. The locking mechanism prevents spurious writes to the Reload Registers. The following sequence is required to unlock the Watchdog Timer Reload Byte Registers (WDTU, WDTH, and WDTL) for write access.

1. Write 55H to the Watchdog Timer Control Register (WDTCTL).
2. Write AAH to the Watchdog Timer Control Register (WDTCTL).
3. Write the Watchdog Timer Reload Upper Byte Register (WDTU).
4. Write the Watchdog Timer Reload High Byte Register (WDTH).
5. Write the Watchdog Timer Reload Low Byte Register (WDTL).

### Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte is received and is available in the UART Receive Data Register. This interrupt can be disabled independent of the other receiver interrupt sources. The received data interrupt occurs once the receive character is received and placed in the Receive Data Register. Software must respond to this received data available condition before the next character is completely received to avoid an overrun error. In MULTIPROCESSOR mode (MPEN = 1), the receive data interrupts are dependent on the multiprocessor configuration and the most recent address byte
- A break is received
- An overrun is detected
- A data framing error is detected

### UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data Register. The break detect and overrun status bits are not displayed until the valid data is read.

After the valid data has been read, the UART Status 0 Register is updated to indicate the overrun condition (and Break Detect, if applicable). The RDA bit is set to 1 to indicate that the Receive Data Register contains a data byte. However, because the overrun error occurred, this byte cannot contain valid data and should 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

Figure16 on page 99 displays the recommended procedure for UART receiver ISRs.

### Baud Rate Generator Interrupts

If the BRG interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the BRG to function as an additional counter if the UART functionality is not employed.

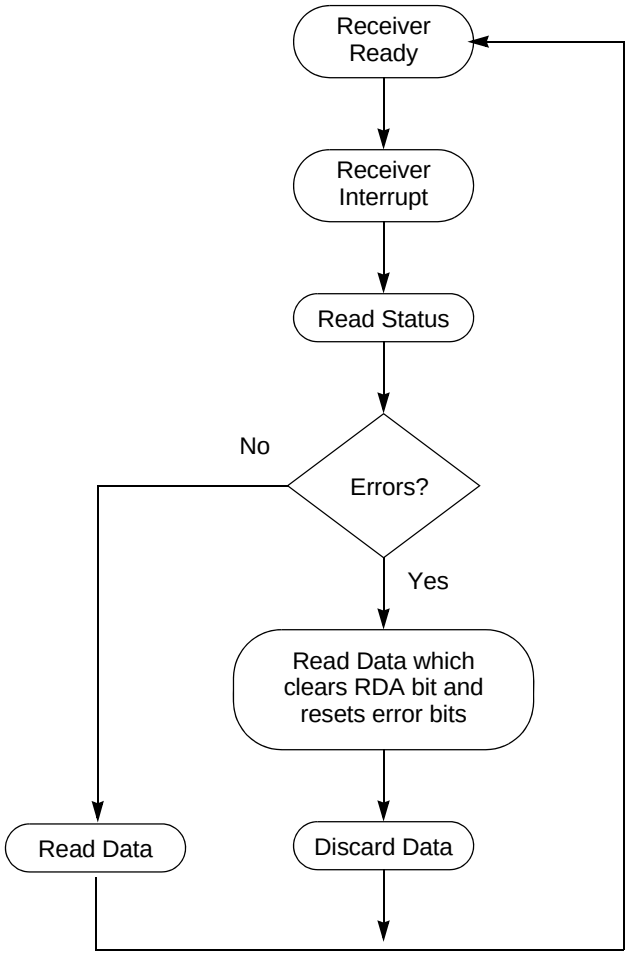



**Figure 16. UART Receiver Interrupt Service Routine Flow**

## UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the BRG is the system clock. The UART Baud Rate High and Low Byte Registers combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. The UART data rate is calculated using the following equation:

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

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. This procedure allows the Endec to tolerate jitter and baud rate errors in the incoming data stream. 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 Endec Control Register Definitions

All Infrared Endec configuration and status information is set by the UART control registers as defined in UART Control Register Definitions on page 100.

**Caution:** *To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 register to 1 to enable the Infrared Endec before enabling the GPIO Port alternate function for the corresponding pin.*

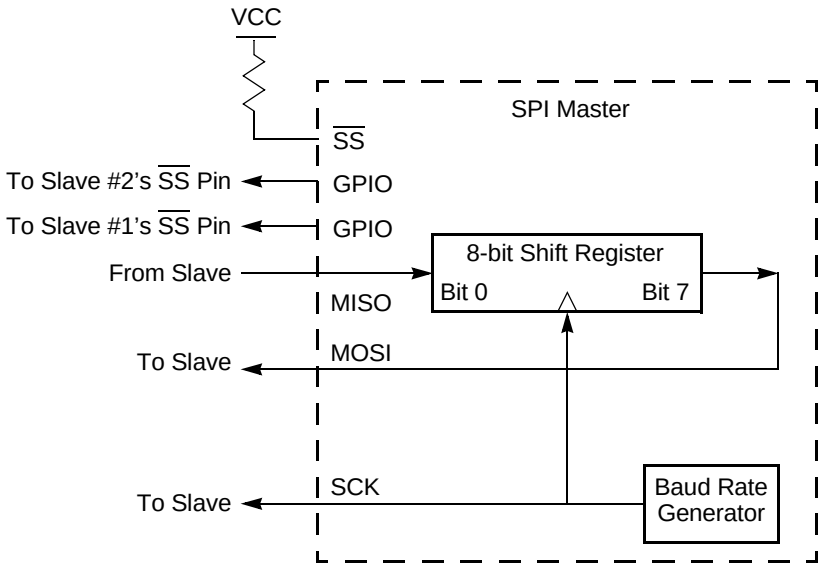



**Figure 21. SPI Configured as a Master in a Single Master, Multiple Slave System**

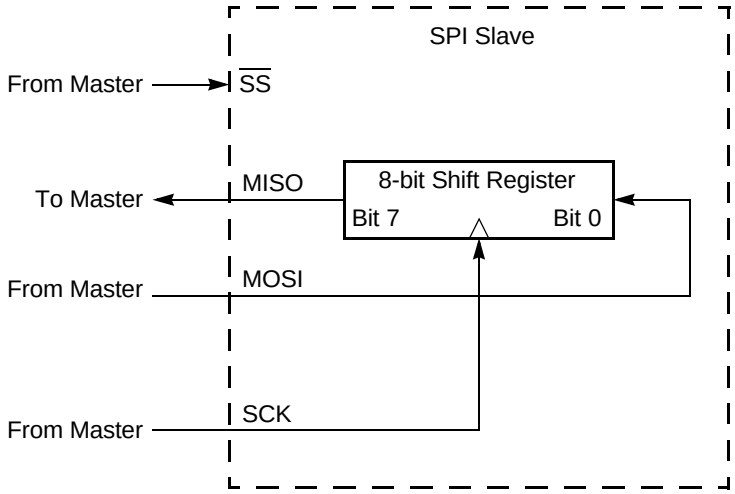



**Figure 22. SPI Configured as a Slave**

## Operation

The SPI is a full-duplex, synchronous, and character-oriented channel that supports a four-wire interface (serial clock, transmit, receive and Slave select). The SPI block consists of a transmit/receive shift register, a Baud Rate (clock) Generator and a control unit.

**INFO_EN—Information Area Enable**
0 = Information Area is not selected.
1 = Information Area is selected. The Information area is mapped into the Flash Memory address space at addresses FE00H through FFFFH.

**PAGE—Page Select**
This 7-bit field selects the Flash memory page for Programming and Page Erase operations. Flash Memory Address[15:9] = PAGE[6:0].

## Flash Sector Protect Register

The Flash Sector Protect Register (Table 86) protects Flash memory sectors from being programmed or erased from user code. The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register can be accessed only after writing the Flash Control Register with 5EH. User code can only write bits in this register to 1 (bits cannot be cleared to 0 by user code).

**Table 86. Flash Sector Protect Register (FPROT)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | SECT7 | SECT6 | SECT5 | SECT4 | SECT3 | SECT2 | SECT1 | SECT0 |
| RESET | 0 | | | | | | | |
| R/W | R/W1 | | | | | | | |
| ADDR | FF9H | | | | | | | |

R/W1 = Register is accessible for Read operations. Register can be written to 1 only (using user code).

**SECT*n*—Sector Protect**
0 = Sector *n* can be programmed or erased from user code.
1 = Sector *n* is protected and cannot be programmed or erased from user code. User code can only write bits from 0 to 1.

## Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte Registers (Table 87 and Table 88) combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit Flash Frequency registers must be written with the system clock frequency in kHz for Program and Erase operations. The Flash Frequency value is calculated using the following equation:

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

**Caution:** *Flash programming and erasure is not supported for system clock frequencies below 20 kHz, above 20 MHz, or outside of the valid operating*

# Option Bits

Option Bits allow user configuration of certain aspects of Z8 Encore! XP® F0822 Series operation. The feature configuration data is stored in Flash Memory and read during Reset. Features available for control through the Option Bits are:

- Watchdog Timer time-out response selection–interrupt or Reset.
- Watchdog Timer enabled at Reset.
- The ability to prevent unwanted read access to user code in Flash Memory.
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Flash Memory.
- Voltage Brownout configuration-always enabled or disabled during STOP mode to reduce STOP mode power consumption.
- Oscillator mode selection-for high, medium, and low power crystal oscillators, or external RC oscillator.

## Operation

### Option Bit Configuration By Reset

During any reset operation (System Reset, Reset, or Stop Mode Recovery), the Option Bits are automatically read from the Flash Memory and written to Option Configuration registers. The Option Configuration registers control operation of the devices within the Z8 Encore! XP F0822 Series. Option Bit control is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access. Each time the Option Bits are programmed or erased, the device must be Reset for the change to take place (Flash version only)

## Option Bit Address Space

The first two bytes of Flash Memory at addresses 0000H (Table 89 on page 164) and 0001H (Table 90 on page 165) are reserved for the user programmable Option Bits. The byte at Program Memory address 0000H configures user options. The byte at Flash Memory address 0001H is reserved for future use and must be left in its unprogrammed state.

## Debug Mode

The operating characteristics of the Z8 Encore! XP® F0822 Series devices in DEBUG mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions.
- The system clock operates unless in STOP mode.
- All enabled on-chip peripherals operate unless in STOP mode.
- Automatically exits HALT mode.
- Constantly refreshes the Watchdog Timer, if enabled.

### Entering Debug Mode

The device enters DEBUG mode following any of the following operations:

- Writing the DBGMODE bit in the OCD Control Register to 1 using the OCD interface.
- eZ8 CPU execution of a BRK (Breakpoint) instruction.
- Match of PC to OCDCNTR register (when enabled)
- OCDCNTR register decrements to 0000H (when enabled)
- If the DBG pin is Low when the device exits Reset, the OCD automatically puts the device into DEBUG mode.

### Exiting Debug Mode

The device exits DEBUG mode following any of the following operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0.
- Power-On Reset
- Voltage Brownout reset
- Asserting the RESET pin Low to initiate a Reset.
- Driving the DBG pin Low while the device is in STOP mode initiates a System Reset.

## OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 Start bit, 8 data bits (least-significant bit first), and 1 STOP bit (see Figure 40).

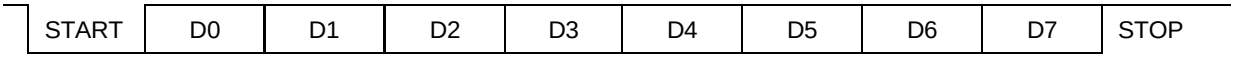



**Figure 40. OCD Data Format**

### OCDCNTR Register

The OCD contains a multipurpose 16-bit Counter Register. It can be used for the following:

- Count system clock cycles between Breakpoints.
- Generate a BRK when it counts down to zero.
- Generate a BRK when its value matches the Program Counter.

When configured as a counter, the OCDCNTR register starts counting when the OCD leaves DEBUG mode and stops counting when it enters DEBUG mode again or when it reaches the maximum count of FFFFH. The OCDCNTR register automatically resets itself to 0000H when the OCD exits DEBUG mode if it is configured to count clock cycles between breakpoints.

**Caution:** *The OCDCNTR register is used by many of the OCD commands. It counts the number of bytes for the register and memory read/write commands. It holds the residual value when generating the CRC. Therefore, if the OCDCNTR is being used to generate a BRK, its value should be written as a last step before leaving DEBUG mode.*

Since this register is overwritten by various OCD commands, it should only be used to generate temporary breakpoints, such as stepping over CALL instructions or running to a specific instruction and stopping.

## On-Chip Debugger Commands

The host communicates to the OCD by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG mode, all OCD commands become available unless the user code and control registers are protected by programming the Read Protect Option Bit (RP). The Read Protect Option Bit prevents the code in memory from being read out of the Z8 Encore! XP F0822 Series products. When this option is enabled, several of the OCD commands are disabled. Table 93 on page 177 contains a summary of the OCD commands. Each OCD command is described further in the bulleted list. It also lists the commands that operate when the device is not in DEBUG mode (normal operation) and those commands that are disabled by programming the Read Protect Option Bit.

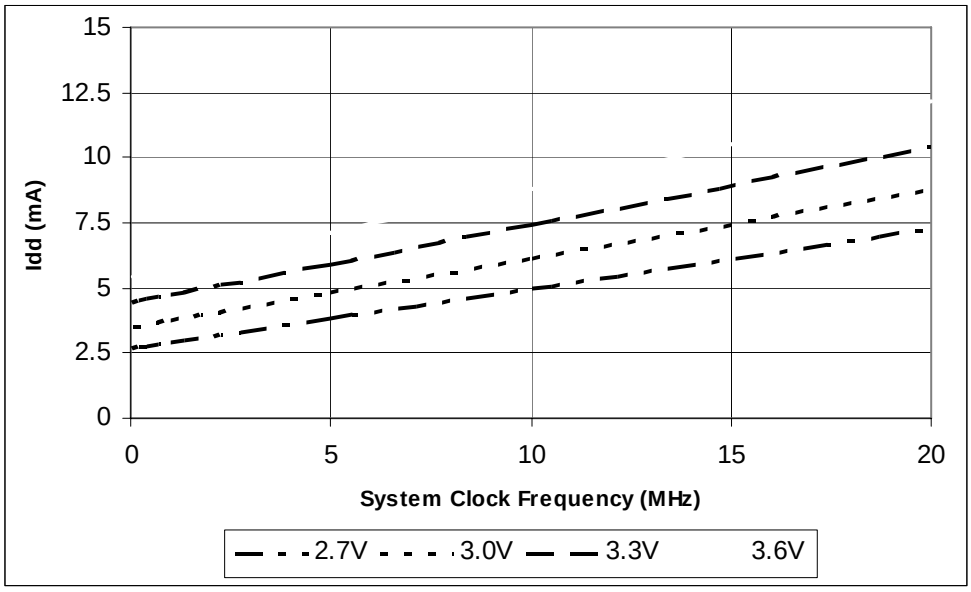



**Figure 41. Typical Active Mode $I_{DD}$ Versus System Clock Frequency**

Figure 42 displays the maximum active mode current consumption across the full operating temperature range of the device and versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

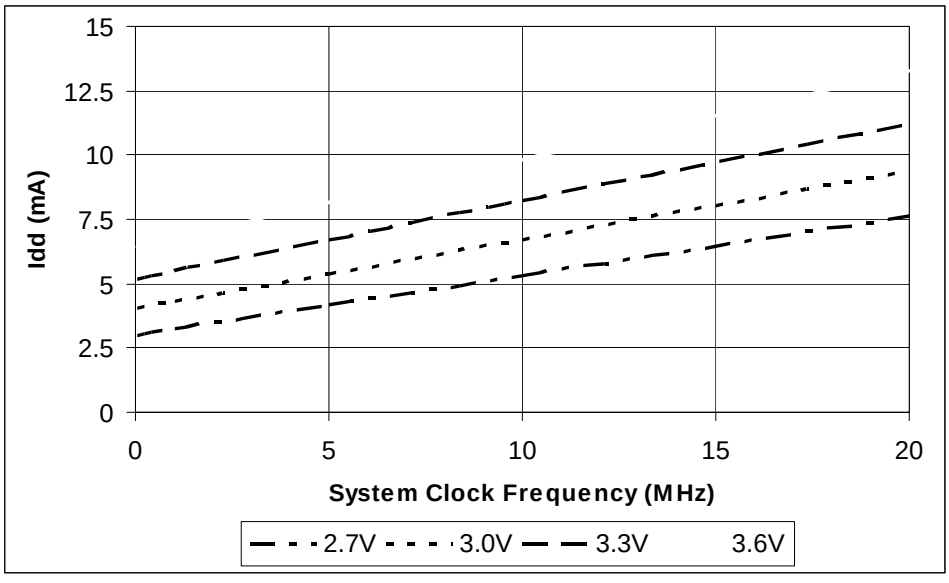



**Figure 42. Maximum Active Mode $I_{DD}$ Versus System Clock Frequency**

**Table 125. Rotate and Shift Instructions**

| Mnemonic | Operands | Instruction |
|---|---|---|
| BSWAP | dst | Bit Swap |
| RL | dst | Rotate Left |
| RLC | dst | Rotate Left through Carry |
| RR | dst | Rotate Right |
| RRC | dst | Rotate Right through Carry |
| SRA | dst | Shift Right Arithmetic |
| SRL | dst | Shift Right Logical |
| SWAP | dst | Swap Nibbles |

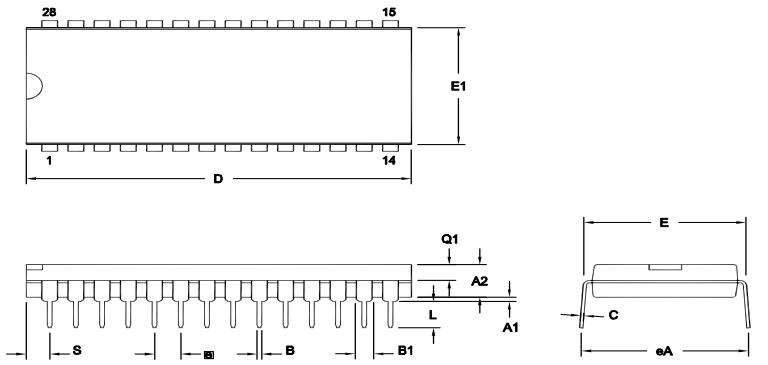
## eZ8 CPU Instruction Summary

Table 126 summarizes the eZ8 CPU instructions. The table identifies the addressing modes employed by the instruction, the effect upon the Flags register, the number of CPU clock cycles required for the instruction fetch, and the number of CPU clock cycles required for the instruction execution.

**Table 126. eZ8 CPU Instruction Summary**

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | dst | src | | C | Z | S | V | D | H | | |
| ADC dst, src | dst ← dst + src + C | r | r | 12 | * | * | * | * | 0 | * | 2 | 3 |
| | | r | Ir | 13 | | | | | | | 2 | 4 |
| | | R | R | 14 | | | | | | | 3 | 3 |
| | | R | IR | 15 | | | | | | | 3 | 4 |
| | | R | IM | 16 | | | | | | | 3 | 3 |
| | | IR | IM | 17 | | | | | | | 3 | 4 |
| ADCX dst, src | dst ← dst + src + C | ER | ER | 18 | * | * | * | * | 0 | * | 4 | 3 |
| | | ER | IM | 19 | | | | | | | 4 | 3 |

Figure 63 displays the 28-pin PDIP package available for Z8 Encore! XP F0822 Series devices.




| SYMBOL | OPT # | MILLIMETER | | INCH | |
|---|---|---|---|---|---|
| | | MIN | MAX | MIN | MAX |
| A1 | | 0.38 | 1.02 | .015 | .040 |
| A2 | | 3.18 | 4.19 | .125 | .165 |
| B | | 0.38 | 0.53 | .015 | .021 |
| B1 | 01 | 1.40 | 1.65 | .055 | .065 |
| | 02 | 1.14 | 1.40 | .045 | .055 |
| C | | 0.23 | 0.38 | .009 | .015 |
| D | 01 | 36.58 | 37.34 | 1.440 | 1.470 |
| | 02 | 35.31 | 35.94 | 1.390 | 1.415 |
| E | | 15.24 | 15.75 | .600 | .620 |
| E1 | 01 | 13.59 | 14.10 | .535 | .555 |
| | 02 | 12.83 | 13.08 | .505 | .515 |
| e | | 2.54 TYP | | .100 BSC | |
| eA | | 15.49 | 16.76 | .610 | .660 |
| L | | 3.05 | 3.81 | .120 | .150 |
| Q1 | 01 | 1.40 | 1.91 | .055 | .075 |
| | 02 | 1.40 | 1.78 | .055 | .070 |
| S | 01 | 1.52 | 2.29 | .060 | .090 |
| | 02 | 1.02 | 1.52 | .040 | .060 |

CONTROLLING DIMENSIONS : INCH

| OPTION TABLE | |
|---|---|
| OPTION # | PACKAGE |
| 01 | STANDARD |
| 02 | IDF |

Note: ZILOG supplies both options for production. Component layout PCB design should cover bigger option 01.

**Figure 63. 28-Pin Plastic Dual-Inline Package (PDIP)**