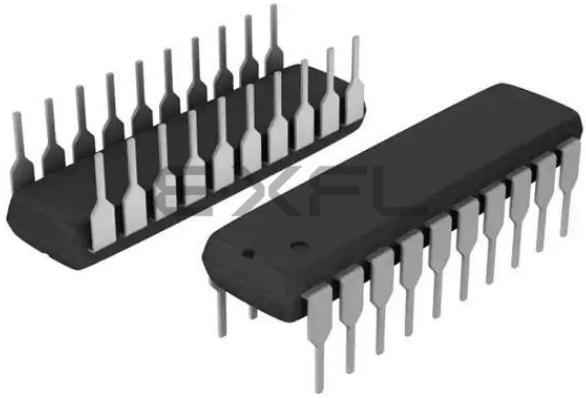

Welcome to E-XFL.COM

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	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	16 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f011aph020eg

CPU and Peripheral Overview

The eZ8 CPU, Zilog's latest 8-bit Central Processing Unit (CPU), meets the continuing demand for faster and more code-efficient microcontrollers. The eZ8 CPU executes a superset of the original Z8 instruction set. The features of eZ8 CPU include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required program memory
- Software stack allows much greater depth in subroutine calls and interrupts than hardware stacks
- Compatible with existing Z8 code
- Expanded internal Register File allows access of up to 4 KB
- New instructions improve execution efficiency for code developed using higher-level programming languages, including C
- Pipelined instruction fetch and execution
- New instructions for improved performance including BIT, BSWAP, BTJ, CPC, LDC, LDCI, LEA, MULT and SRL
- New instructions support 12-bit linear addressing of the Register File
- Up to 10 MIPS operation
- C-Compiler friendly
- 2 to 9 clock cycles per instruction

For more information about eZ8 CPU, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download on www.zilog.com.

10-Bit Analog-to-Digital Converter

The optional analog-to-digital converter (ADC) converts an analog input signal to a 10-bit binary number. The ADC accepts inputs from eight different analog input pins in both single-ended and differential modes. The ADC also features a unity gain buffer when high input impedance is required.

Low-Power Operational Amplifier

The optional low-power operational amplifier (LPO) is a general-purpose amplifier primarily targeted for current sense applications. The LPO output may be routed internally to the ADC or externally to a pin.

Signal Descriptions

Table 2 describes the Z8 Encore! XP F082A Series signals. See the Pin Configurations section on page 8 to determine the signals available for the specific package styles.

Table 2. 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]	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.
PD[0]	I/O	Port D. This pin is used for general-purpose output only.
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.
$\overline{\text{CTS0}}$	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 may 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 outputs from the timers.
$\overline{\text{T0OUT}}$/$\overline{\text{T1OUT}}$	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.
Comparator		
CINP/CINN	I	Comparator Inputs. These signals are the positive and negative inputs to the comparator.
COUT	O	Comparator Output.

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.

Port A–C Input Data Registers

Reading from the Port A–C Input Data registers, shown in Table 29, return the sampled values from the corresponding port pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8- and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

Table 29. Port A–C Input Data Registers (PxIN)

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	FD2H, FD6H, FDAH							

X = Undefined.

Bit	Description
[7:0] PxIN	**Port Input Data** Sampled data from the corresponding port pin input. 0 = Input data is logical 0 (Low). 1 = Input data is logical 1 (High).

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

Port A–D Output Data Register

The Port A–D Output Data Register, shown in Table 30, controls the output data to the pins.

Table 30. Port A–D Output Data Register (P*x*OUT)

Bit	7	6	5	4	3	2	1	0
Field	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
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	FD3H, FD7H, FDBH, FDFH							

Bit	Description
[7:0] PxOUT	**Port Output Data** These bits contain the data to be driven to the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation. 0 = Drive a logical 0 (Low). 1 = Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding Port Output Control Register bit to 1.

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

LED Drive Enable Register

The LED Drive Enable Register, shown in Table 31, activates the controlled current drive. The Alternate Function Register has no control over the LED function; therefore, setting the Alternate Function Register to select the LED function is not required. LEDEN bits [7:0] correspond to Port C bits [7:0], respectively.

Table 31. LED Drive Enable (LEDEN)

Bit	7	6	5	4	3	2	1	0
Field	LEDEN[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F82H							

Bit	Description
[7:0] LEDENx	**LED Drive Enable** These bits determine which Port C pins are connected to an internal current sink. 0 = Tristate the Port C pin. 1 = Enable controlled current sink on the Port C pin. **Note:** *x* indicates the specific GPIO port pin number (7–0).

LED Drive Level High Register

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

Table 32. LED Drive Level High Register (LEDLVLH)

Bit	7	6	5	4	3	2	1	0
Field	LEDLVLH[7:0]							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F83H							

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

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

GPIO Mode Interrupt Controller

The interrupt controller on the Z8 Encore! XP F082A Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 possible interrupt sources with 18 unique interrupt vectors:
  - Twelve GPIO port pin interrupt sources (two interrupt vectors are shared)
  - Eight on-chip peripheral interrupt sources (two interrupt vectors are shared)
- Flexible GPIO interrupts:
  - Eight selectable rising and falling edge GPIO interrupts
  - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Watchdog Timer and LVD can be configured to generate an interrupt
- Supports vectored and polled interrupts

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt controller has no effect on operation. For more information about interrupt servicing by the eZ8 CPU, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download on www.zilog.com.

Interrupt Vector Listing

Table 34 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

Note: Some port interrupts are not available on the 8- and 20-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

! **Caution:** To avoid retriggerings of the Watchdog Timer interrupt after exiting the associated interrupt service routine, Zilog recommends that the service routine continues to read from the RSTSTAT Register until the WDT bit is cleared as shown in the following example.

```
CLEARWDT:
 LDX r0, RSTSTAT ; read reset status register to clear wdt bit
 BTJNZ 5, r0, CLEARWDT   ; loop until bit is cleared
```

Interrupt Control Register Definitions

For all interrupts other than the Watchdog Timer interrupt, the Primary Oscillator Fail Trap and the Watchdog Oscillator Fail Trap, the interrupt control registers enable individual interrupts, set interrupt priorities and indicate interrupt requests.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register, shown in Table 35, stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 0 Register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 0 Register (IRQ0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1I	T0I	U0RXI	U0TXI	Reserved	Reserved	ADCI
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	FC0H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1I	**Timer 1 Interrupt Request** 0 = No interrupt request is pending for Timer 1. 1 = An interrupt request from Timer 1 is awaiting service.
[5] T0I	**Timer 0 Interrupt Request** 0 = No interrupt request is pending for Timer 0. 1 = An interrupt request from Timer 0 is awaiting service.

Bit	Description (Continued)
[5:3] PRES	**Prescale value** The timer input clock is divided by 2^{PRES}, where PRES can be set from 0 to 7. The prescaler is reset each time the Timer is disabled. This reset ensures proper clock division each time the Timer is restarted. 000 = Divide by 1. 001 = Divide by 2. 010 = Divide by 4. 011 = Divide by 8. 100 = Divide by 16. 101 = Divide by 32. 110 = Divide by 64. 111 = Divide by 128.
[2:0] TMODE	**Timer Mode** This field, along with the TMODEHI bit in the TxCTL0 Register, determines the operating mode of the timer. TMODEHI is the most significant bit of the Timer mode selection value. The entire operating mode bits are expressed as {TMODEHI, TMODE[2:0]}. The TMODEHI is bit 7 of the TxCTL0 Register while TMODE[2:0] is the lower 3 bits of the TxCTL1 Register. 0000 = ONE-SHOT Mode. 0001 = CONTINUOUS Mode. 0010 = COUNTER Mode. 0011 = PWM SINGLE OUTPUT Mode. 0100 = CAPTURE Mode. 0101 = COMPARE Mode. 0110 = GATED Mode. 0111 = CAPTURE/COMPARE Mode. 1000 = PWM DUAL OUTPUT Mode. 1001 = CAPTURE RESTART Mode. 1010 = COMPARATOR COUNTER Mode.

Timer 0–1 High and Low Byte Registers

The Timer 0–1 High and Low Byte (TxH and TxL) registers, shown in Tables 52 and 53, contain the current 16-bit timer count value. When the timer is enabled, a read from TxH causes the value in TxL to be stored in a temporary holding register. A read from TxL always returns this temporary register when the timers are enabled. When the timer is disabled, reads from TxL read the register directly.

Writing to the Timer High and Low Byte registers while the timer is enabled is not recommended. There are no temporary holding registers available for write operations, so simultaneous 16-bit writes are not possible. If either the Timer High or Low Byte registers are written during counting, the 8-bit written value is placed in the counter (High or Low Byte) at the next clock edge. The counter continues counting from the new value.

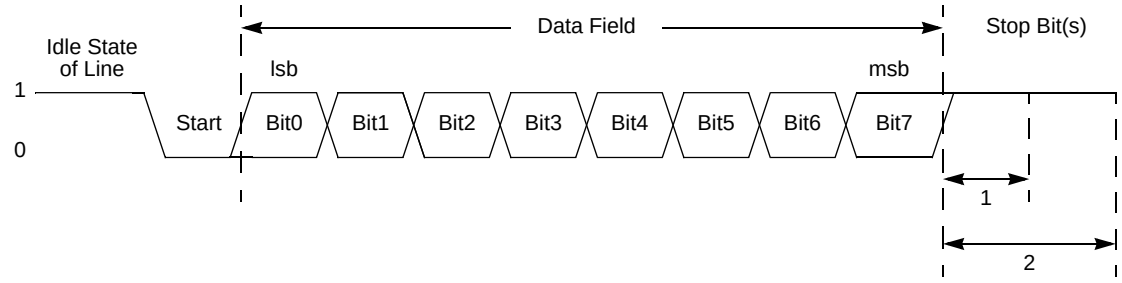

Figure 11. UART Asynchronous Data Format without Parity

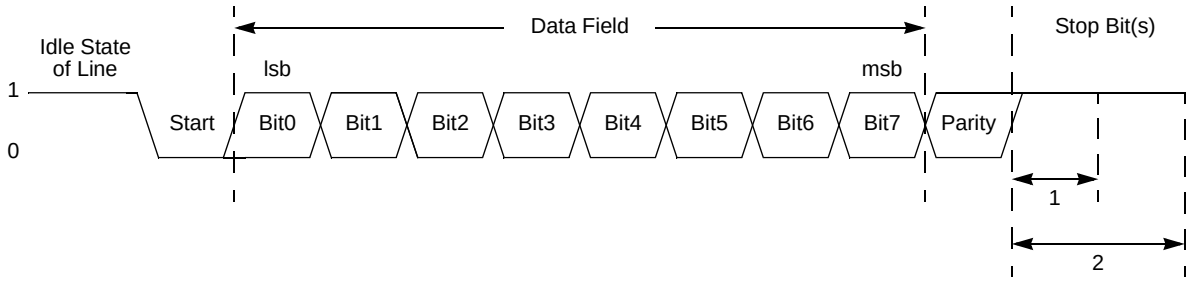

Figure 12. UART Asynchronous Data Format with Parity

Transmitting Data using the Polled Method

Observe the following steps to transmit data using the polled method of operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the required baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Write to the UART Control 1 Register, if MULTIPROCESSOR Mode is appropriate, to enable MULTIPROCESSOR (9-bit) Mode functions.
4. Set the Multiprocessor Mode Select (MPEN) bit to enable MULTIPROCESSOR Mode.
5. Write to the UART Control 0 Register to:
   - Set the transmit enable bit (TEN) to enable the UART for data transmission
   - Set the parity enable bit (PEN), if parity is appropriate and MULTIPROCESSOR Mode is not enabled and select either even or odd parity (PSEL)

UART Interrupts

The UART features separate interrupts for the transmitter and the receiver. In addition, when the UART primary functionality is disabled, the Baud Rate Generator can also function as a basic timer with interrupt capability.

Transmitter Interrupts

The transmitter generates a single interrupt when the Transmit Data Register Empty bit (TDRE) is set to 1. This indicates that the transmitter is ready to accept new data for transmission. The TDRE interrupt occurs after the Transmit Shift Register has shifted the first bit of data out. The Transmit Data Register can now be written with the next character to send. This action provides 7 bit periods of latency to load the Transmit Data Register before the Transmit Shift Register completes shifting the current character. Writing to the UART Transmit Data Register clears the TDRE bit to 0.

Receiver Interrupts

The receiver generates an interrupt when any of the following actions occur:

- A data byte is received and is available in the UART Receive Data Register. This interrupt can be disabled independently of the other receiver interrupt sources. The received data interrupt occurs after the receive character has been received and placed in the Receive Data Register. To avoid an overrun error, software must respond to this received data available condition before the next character is completely received.

Note: 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 after the valid data has been 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 may not 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

Table 69. UART Address Compare Register (U0ADDR)

Bit	7	6	5	4	3	2	1	0
Field	COMP_ADDR							
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	F45H							

Bit	Description
[7:0] COMP_ADDR	**Compare Address** This 8-bit value is compared to incoming address bytes.

UART Baud Rate High and Low Byte Registers

The UART Baud Rate High (UxBRH) and Low Byte (UxBRL) registers, shown in Tables 70 and 71, combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART.

Table 70. UART Baud Rate High Byte Register (U0BRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
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	F46H							

Bit	Description
[7:0] BRH	**UART Baud Rate High Byte**

Table 71. UART Baud Rate Low Byte Register (U0BRL)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
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	F47H							

Bit	Description
[7:0] BRL	**UART Baud Rate Low Byte**

Table 75. ADC Data High Byte Register (ADCD_H)

Bit	7	6	5	4	3	2	1	0
Field	ADCDH							
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	F72H							
X = Undefined.								

Bit	Description
[7:0] ADCDH	**ADC Data High Byte** This byte contains the upper eight bits of the ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the most recent conversion output is held in this register. These bits are undefined after a Reset.

ADC Data Low Byte Register

The ADC Data Low Byte (ADCD_L) Register contains the lower bits of the ADC output plus an overflow status bit. The output is a 13-bit two's complement value. During a single-shot conversion, this value is invalid. Access to the ADC Data Low Byte Register is read-only. Reading the ADC Data High Byte Register latches data in the ADC Low Bits Register.

Table 76. ADC Data Low Byte Register (ADCD_L)

Bit	7	6	5	4	3	2	1	0
Field	ADCDL					Reserved		OVF
RESET	X	X	X	X	X	X	X	X
R/W	R	R	R	R	R	R	R	R
Address	F73H							
X = Undefined.								

Bit	Description
[7:3] ADCDL	**ADC Data Low Bits** These bits are the least significant five bits of the 13-bits of the ADC output. These bits are undefined after a Reset.

Comparator

The Z8 Encore! XP F082A Series devices feature a general purpose comparator that compares two analog input signals. These analog signals may be external stimulus from a pin (CINP and/or CINN) or internally generated signals. Both a programmable voltage reference and the temperature sensor output voltage are available internally. The output is available as an interrupt source or can be routed to an external pin.

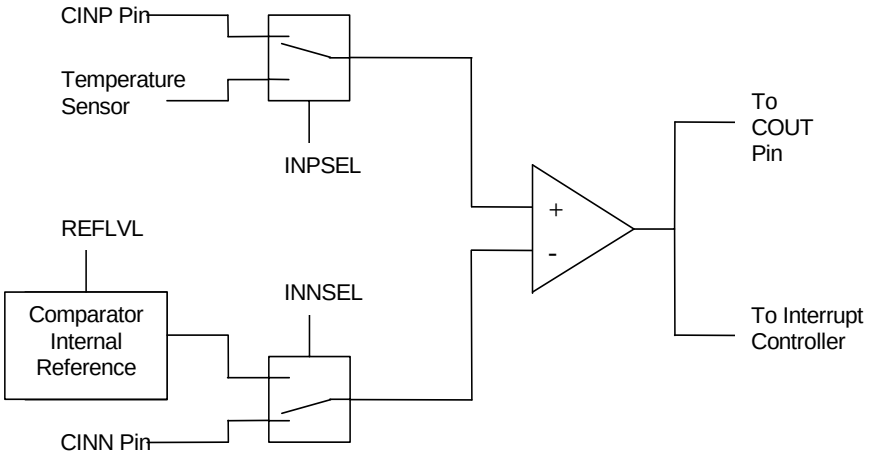

Figure 20. Comparator Block Diagram

Operation

When the positive comparator input exceeds the negative input by more than the specified hysteresis, the output is a logic High. When the negative input exceeds the positive by more than the hysteresis, the output is a logic Low. Otherwise, the comparator output retains its present value. See Table 141 on page 238 for details.

The comparator may be powered down to reduce supply current. See the Power Control Register 0 section on page 33 for details.

! 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.

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

Table 84. Flash Frequency High Byte Register (FFREQH)

Bit	7	6	5	4	3	2	1	0
Field	FFREQH							
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	FFAH							

Bit	Description
[7:0] FFREQH	**Flash Frequency High Byte** High byte of the 16-bit Flash Frequency value.

Table 85. Flash Frequency Low Byte Register (FFREQL)

Bit	7	6	5	4	3	2	1	0
Field	FFREQL							
RESET	0							
R/W	R/W							
Address	FFBH							

Bit	Description
[7:0] FFREQL	**Flash Frequency Low Byte** Low byte of the 16-bit Flash Frequency value.

Flash Option Bits

Programmable Flash option bits allow user configuration of certain aspects of Z8 Encore! XP F082A Series operation. The feature configuration data is stored in Flash program memory and loaded into holding registers during Reset. The features available for control through the Flash option bits include:

- Watchdog Timer time-out response selection–interrupt or system reset
- Watchdog Timer always on (enabled at Reset)
- The ability to prevent unwanted read access to user code in Program Memory
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Program Memory
- Voltage Brown-Out 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
- Factory trimming information for the internal precision oscillator and low voltage detection
- Factory calibration values for ADC, temperature sensor and Watchdog Timer compensation
- Factory serialization and randomized lot identifier (optional)

Operation

This section describes the type and configuration of the programmable Flash option bits.

Option Bit Configuration By Reset

Each time the Flash option bits are programmed or erased, the device must be Reset for the change to take effect. During any reset operation (System Reset, Power-On Reset, or Stop Mode Recovery), the Flash option bits are automatically read from Flash program memory and written to the Option Configuration registers. The Option Configuration registers control the operation of the devices within the Z8 Encore! XP F082A 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.

Table 142. Temperature Sensor Electrical Characteristics

		V_{DD} = 2.7 V to 3.6 V				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
T_{AERR}	Temperature Error		±0.5	±2	°C	Over the range +20°C to +30°C (as measured by ADC).[1]
			±1	±5	°C	Over the range +0°C to +70°C (as measured by ADC).
			±2	±7	°C	Over the range +0°C to +105°C (as measured by ADC).
			±7		°C	Over the range –40°C to +105°C (as measured by ADC).
t_{WAKE}	Wakeup Time		80	100	µs	Time required for Temperature Sensor to stabilize after enabling.

Note: Devices are factory calibrated at for maximal accuracy between +20°C and +30°C, so the sensor is maximally accurate in that range. User recalibration for a different temperature range is possible and increases accuracy near the new calibration point.

J

L

U

V

W

X

Customer Support

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

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

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