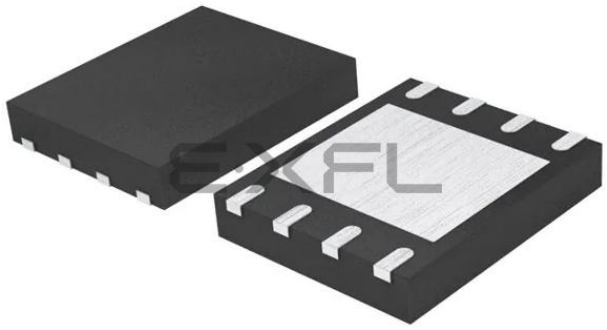


**Welcome to E-XFL.COM**

## 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, Temp Sensor, WDT |
| Number of I/O | 6 |
| Program Memory Size | 2KB (2K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 64 x 8 |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | A/D 4x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 8-VDFN Exposed Pad |
| Supplier Device Package | 8-QFN (5x6) |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f022aqb020ec |

## Universal Asynchronous Receiver/Transmitter

The full-duplex universal asynchronous receiver/transmitter (UART) is included in all Z8 Encore! XP package types. The UART supports 8- and 9-bit data modes and selectable parity. The UART also supports multi-drop address processing in hardware. The UART baud rate generator (BRG) can be configured and used as a basic 16-bit timer.

## Timers

Two enhanced 16-bit reloadable timers can be used for timing/counting events or for motor control operations. These timers provide a 16-bit programmable reload counter and operate in ONE-SHOT, CONTINUOUS, GATED, CAPTURE, CAPTURE RESTART, COMPARE, CAPTURE and COMPARE, PWM SINGLE OUTPUT and PWM DUAL OUTPUT modes.

## General-Purpose Input/Output

The Z8 Encore! XP F082A Series features 6 to 25 port pins (Ports A–D) for general- purpose input/output (GPIO). The number of GPIO pins available is a function of package, and each pin is individually programmable. 5 V tolerant input pins are available on all I/Os on 8-pin devices and most I/Os on other package types.

## Direct LED Drive

The 20- and 28-pin devices support controlled current sinking output pins capable of driving LEDs without the need for a current limiting resistor. These LED drivers are independently programmable to four different intensity levels.

## Flash Controller

The Flash Controller programs and erases Flash memory. The Flash Controller supports several protection mechanisms against accidental program and erasure, as well as factory serialization and read protection.

## Non-Volatile Data Storage

The non-volatile data storage (NVDS) uses a hybrid hardware/software scheme to implement a byte programmable data memory and is capable of over 100,000 write cycles.

**Note:** *Devices with 8 KB Flash memory do not include the NVDS feature.*

# Address Space

The eZ8 CPU can access the following three distinct address spaces:

1. The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral, and general-purpose I/O port control registers.
2. The Program Memory contains addresses for all memory locations having executable code and/or data.
3. The Data Memory contains addresses for all memory locations that contain data only.

These three address spaces are covered briefly in the following subsections. For more information on eZ8 CPU and its address space, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com.

## Register File

The Register File address space in the Z8 Encore!® MCU is 4 KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as sources are read, and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4 KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256 B control register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The Z8 Encore! XP® F082A Series devices contain 256 B to 1 KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

## Program Memory

The eZ8 CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP F082A Series devices contain 1 KB to 8 KB of on-chip Flash memory in the Program Memory address space, depending on the device. Reading from Program Memory

**Caution:** *To avoid re-triggerings of the Watchdog Timer interrupt after exiting the associated interrupt service routine, it is recommended that the service routine continues to read from the RSTSTAT register until the WDT bit is cleared as given in the following coding sample:*

```
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 (Table 33) 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 33. Interrupt Request 0 Register (IRQ0)**

| BITS | 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 |
| ADDR | FC0H | | | | | | | |

Reserved—Must be 0.

T1I—Timer 1 Interrupt Request
0 = No interrupt request is pending for Timer 1.
1 = An interrupt request from Timer 1 is awaiting service.

T0I—Timer 0 Interrupt Request
0 = No interrupt request is pending for Timer 0.
1 = An interrupt request from Timer 0 is awaiting service.

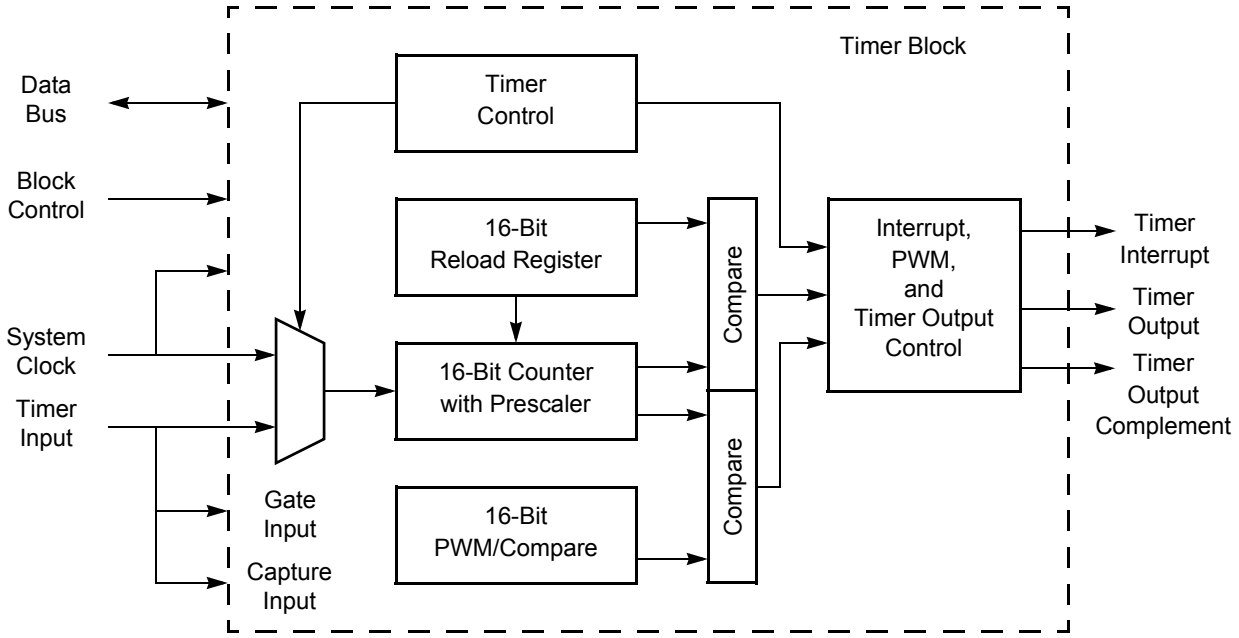



**Figure 9. Timer Block Diagram**

# Operation

The timers are 16-bit up-counters. Minimum time-out delay is set by loading the value 0001H into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value 0000H into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the Timer reaches FFFFH, the timer rolls over to 0000H and continues counting.

## Timer Operating Modes

The timers can be configured to operate in the following modes:

### ONE-SHOT Mode

In ONE-SHOT mode, the timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low byte registers. The timer input is the system clock. Upon reaching the Reload value, the timer generates an interrupt and the count value in the Timer High and Low Byte registers is reset to 0001H. The timer is automatically disabled and stops counting.

Follow the steps below for configuring a timer for COMPARATOR COUNTER mode and initiating the count:

1. Write to the Timer Control register to:
   - Disable the timer.
   - Configure the timer for COMPARATOR COUNTER mode.
   - Select either the rising edge or falling edge of the comparator output signal for the count. This also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function is not required to be enabled.
2. Write to the Timer High and Low Byte registers to set the starting count value. This action only affects the first pass in COMPARATOR COUNTER mode. After the first timer Reload in COMPARATOR COUNTER mode, counting always begins at the reset value of 0001H. Generally, in COMPARATOR COUNTER mode the Timer High and Low Byte registers must be written with the value 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
6. Write to the Timer Control register to enable the timer.

In COMPARATOR COUNTER mode, the number of comparator output transitions since the timer start is given by the following equation:

$$\text{Comparator Output Transitions} = \text{Current Count Value} - \text{Start Value}$$

### PWM SINGLE OUTPUT Mode

In PWM SINGLE OUTPUT mode, the timer outputs a Pulse-Width Modulator (PWM) output signal through a GPIO Port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

If the TPOL bit in the Timer Control register is set to 1, the Timer Output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the Reload value and is reset to 0001H.

duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).

5. 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.
6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
8. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

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

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the ONE-SHOT mode equation determines 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 represented 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 represented 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 appropriate 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 INPCAP bit in TxCTL0 register is set to indicate the timer interrupt is because of an input capture event.

The timer continues counting up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the Reload value, the timer generates an interrupt and continues counting. The INPCAP bit in TxCTL0 register clears indicating the timer interrupt is not because of an input capture event.

4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers.By default, the timer interrupt are 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 TxCTL0 register.
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 can be calculated using the following equation:

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

## 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 register is read, the contents of the Timer Low Byte register are placed in a holding register. A subsequent read from the Timer Low Byte register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte register returns the actual value in the counter.

## Timer Pin Signal Operation

Timer Output is a GPIO Port pin alternate function. The Timer Output is toggled every time the counter is reloaded.

The Timer Input can be used as a selectable counting source. It shares the same pin as the complementary timer output. When selected by the GPIO Alternate Function Registers, this pin functions as a timer input in all modes except for the DUAL PWM OUTPUT mode. For this mode, there is no timer input available.

## UART Address Compare Register

The UART Address Compare (UxADDR) register stores the multi-node network address of the UART (see Table 67). When the MPMD[1] bit of UART Control Register 0 is set, all incoming address bytes are compared to the value stored in the Address Compare register. Receive interrupts and RDA assertions only occur in the event of a match.

**Table 67. UART Address Compare Register (U0ADDR)**

| BITS | 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 |
| ADDR | F45H | | | | | | | |

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 (Table 68 and Table 69) 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 68. UART Baud Rate High Byte Register (U0BRH)**

| BITS | 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 |
| ADDR | F46H | | | | | | | |

**Table 69. UART Baud Rate Low Byte Register (U0BRL)**

| BITS | 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 |
| ADDR | F47H | | | | | | | |

**Table 73. ADC Data High Byte Register (ADCD_H)**

| BITS | 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 |
| ADDR | F72H | | | | | | | |
| X = Undefined. | | | | | | | | |

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 as well as 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 74. ADC Data Low Byte Register (ADCD_L)**

| BITS | 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 |
| ADDR | F73H | | | | | | | |
| X = Undefined. | | | | | | | | |

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.

Reserved—Must be undefined.

OVF—Overflow Status
0= A hardware overflow did not occur in the ADC for the current sample.
1= A hardware overflow did occur in the ADC for the current sample, therefore the current sample is invalid.

## Trim Bit Address 0001H

**Table 89. Trim Option Bits at 0001H**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | Reserved | | | | | | | |
| 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 |
| ADDR | Information Page Memory 0021H | | | | | | | |

**Note:** U = Unchanged by Reset. R/W = Read/Write.

Reserved—Altering this register may result in incorrect device operation.

## Trim Bit Address 0002H

**Table 90. Trim Option Bits at 0002H (TIPO)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | IPO_TRIM | | | | | | | |
| RESET | U | | | | | | | |
| R/W | R/W | | | | | | | |
| ADDR | Information Page Memory 0022H | | | | | | | |

**Note:** U = Unchanged by Reset. R/W = Read/Write.

IPO_TRIM—Internal Precision Oscillator Trim Byte
Contains trimming bits for Internal Precision Oscillator.

## Trim Bit Address 0003H

**Note:** *The LVD is available on 8-pin devices only.*

**Table 91. Trim Option Bits at Address 0003H (TLVD)**

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| FIELD | Reserved | | | LVD_TRIM | | | | |
| 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 |
| ADDR | Information Page Memory 0023H | | | | | | | |

**Note:** U = Unchanged by Reset. R/W = Read/Write.

Table 116 through Table 123 lists 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 116. 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 |
| 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 123. Rotate and Shift Instructions (Continued)**

| Mnemonic | Operands | Instruction |
|---|---|---|
| SRA | dst | Shift Right Arithmetic |
| SRL | dst | Shift Right Logical |
| SWAP | dst | Swap Nibbles |

# eZ8 CPU Instruction Summary

Table 124 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 124. 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 |
| ADD dst, src | dst ← dst + src | r | r | 02 | * | * | * | * | 0 | * | 2 | 3 |
| | | r | Ir | 03 | | | | | | | 2 | 4 |
| | | R | R | 04 | | | | | | | 3 | 3 |
| | | R | IR | 05 | | | | | | | 3 | 4 |
| | | R | IM | 06 | | | | | | | 3 | 3 |
| | | IR | IM | 07 | | | | | | | 3 | 4 |
| ADDX dst, src | dst ← dst + src | ER | ER | 08 | * | * | * | * | 0 | * | 4 | 3 |
| | | ER | IM | 09 | | | | | | | 4 | 3 |
| Flags Notation: | * = Value is a function of the result of the operation.<br>– = Unaffected<br>X = Undefined | | | | 0 = Reset to 0<br>1 = Set to 1 | | | | | | | |

**Table 124. eZ8 CPU Instruction Summary (Continued)**

| Assembly Mnemonic | Symbolic Operation | Address Mode dst | Address Mode src | Opcode(s) (Hex) | Flags C | Z | S | V | D | H | Fetch Cycles | Instr. Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| DA dst | dst ← DA(dst) | R | | 40 | * | * | * | X | – | – | 2 | 2 |
| | | IR | | 41 | | | | | | | 2 | 3 |
| DEC dst | dst ← dst - 1 | R | | 30 | – | * | * | * | – | – | 2 | 2 |
| | | IR | | 31 | | | | | | | 2 | 3 |
| DECW dst | dst ← dst - 1 | RR | | 80 | – | * | * | * | – | – | 2 | 5 |
| | | IRR | | 81 | | | | | | | 2 | 6 |
| DI | IRQCTL[7] ← 0 | | | 8F | – | – | – | – | – | – | 1 | 2 |
| DJNZ dst, RA | dst ← dst – 1<br>if dst ≠ 0<br>PC ← PC + X | r | | 0A-FA | – | – | – | – | – | – | 2 | 3 |
| EI | IRQCTL[7] ← 1 | | | 9F | – | – | – | – | – | – | 1 | 2 |
| Flags Notation: | * = Value is a function of the result of the operation.<br>– = Unaffected<br>X = Undefined | | | | 0 = Reset to 0<br>1 = Set to 1 | | | | | | | |

**Table 135. Analog-to-Digital Converter Electrical Characteristics and Timing (Continued)**

| | | $V_{DD}$ = 3.0 V to 3.6 V<br>$T_A$ = 0 °C to +70 °C<br>(unless otherwise stated) | | | | |
|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Minimum** | **Typical** | **Maximum** | **Units** | **Conditions** |
| | 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 |
| | Signal Input Bandwidth | – | 10 | | kHz | As defined by -3 dB point |
| $R_S$ | Analog Source Impedance[4] | – | – | 10 | kΩ | In unbuffered mode |
| | | | | 500 | kΩ | In buffered modes |
| Zin | Input Impedance | – | 150 | | kΩ | In unbuffered mode at 20 MHz[5] |
| | | 10 | – | | MΩ | In buffered modes |
| Vin | Input Voltage Range | 0 | | $V_{DD}$ | V | Unbuffered Mode |
| | | 0.3 | | $V_{DD}$-1.1 | V | Buffered Modes |
| | | | | | **Note:** | *These values define the range over which the ADC performs within spec; exceeding these values does not cause damage or instability;* see DC Characteristics on page 222 *for absolute pin voltage limits* |

**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.

## UART Timing

Figure 37 and Table 142 provide timing information for UART pins for the case where CTS is used for flow control. The CTS to DE assertion delay (T1) assumes the transmit data register has been loaded with data prior to CTS assertion.

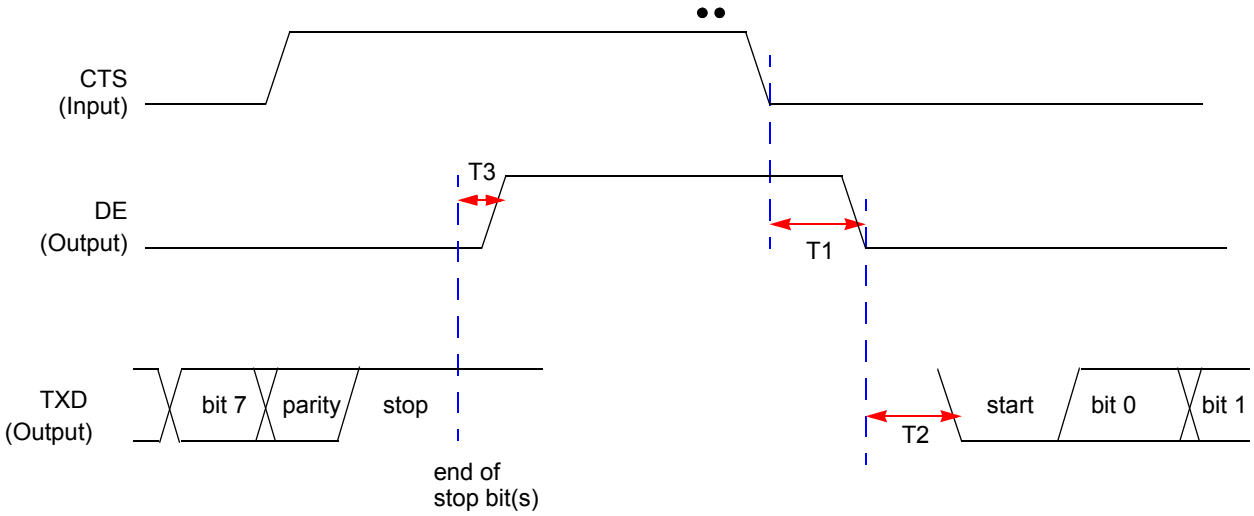
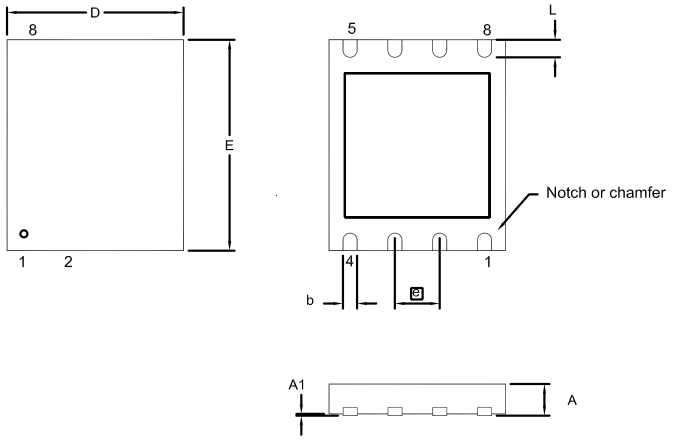



**Figure 37. UART Timing With CTS**

**Table 142. UART Timing With CTS**

| | | Delay (ns) | |
|---|---|---|---|
| **Parameter** | **Abbreviation** | **Minimum** | **Maximum** |
| **UART** | | | |
| $T_1$ | CTS Fall to DE output delay | 2 * XIN period | 2 * XIN period + 1 bit time |
| $T_2$ | DE assertion to TXD falling edge (start bit) delay | ± 5 | |
| $T_3$ | End of Stop Bit(s) to DE deassertion delay | ± 5 | |

Figure 41 displays the 8-pin Quad Flat No-Lead package (QFN)/MLF-S available for the Z8 Encore! XP F082A Series devices. This package has a footprint identical to that of the 8-pin SOIC, but with a lower profile.

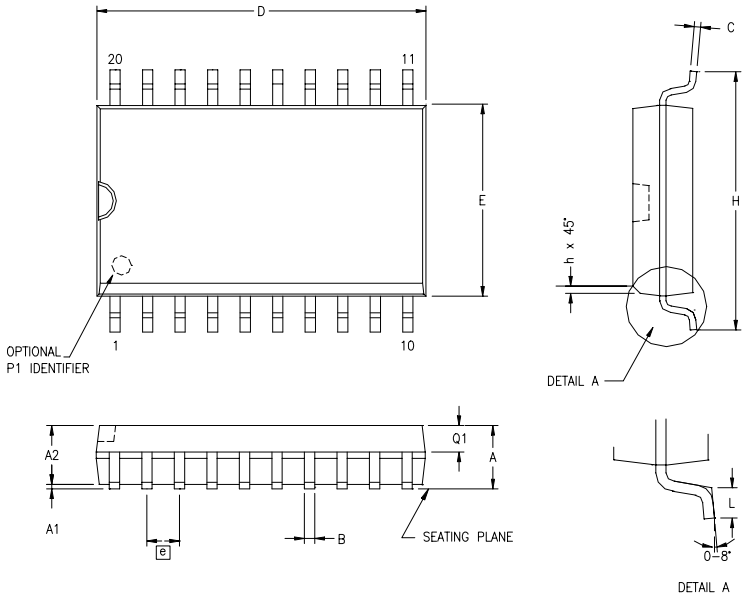



| SYMBOL | MILLIMETER | | INCH | |
|---|---|---|---|---|
| | MIN | MAX | MIN | MAX |
| A | 0.80 | 1.00 | 0.032 | 0.039 |
| A1 | 0 | 0.05 | 0 | 0.002 |
| b | 0.35 | 0.45 | 0.014 | 0.018 |
| D | 4.95 | 5.05 | 0.195 | 0.199 |
| E | 5.95 | 6.05 | 0.234 | 0.238 |
| e | 1.27 BSC | | 0.050 BSC | |
| L | 0.50 | 0.70 | 0.020 | 0.028 |

1. CONTROLLING DIMENSIONS : mm
2. MAX. COPLANARITY : 0.08 mm / 0.003"

**Figure 41. 8-Pin Quad Flat No-Lead Package (QFN)/MLF-S**

Figure 43 displays the 20-pin Small Outline Integrated Circuit Package (SOIC) available for the Z8 Encore! XP F082A Series devices.

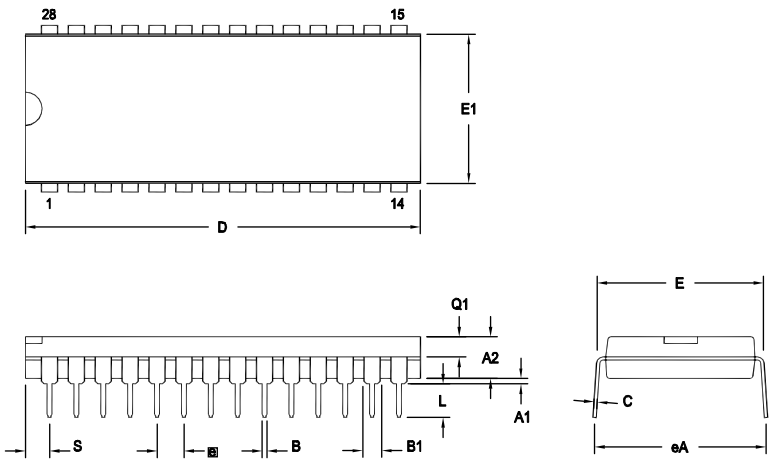



| SYMBOL | MILLIMETER | | INCH | |
|---|---|---|---|---|
| | MIN | MAX | MIN | MAX |
| A | 2.40 | 2.65 | .094 | .104 |
| A1 | 0.10 | 0.30 | .004 | .012 |
| A2 | 2.24 | 2.44 | .088 | .096 |
| B | 0.36 | 0.46 | .014 | .018 |
| C | 0.23 | 0.30 | .009 | .012 |
| D | 12.60 | 12.95 | **.496** | .510 |
| E | 7.40 | 7.60 | .291 | .299 |
| e | 1.27 BSC | | .050 BSC | |
| H | 10.00 | 10.65 | .394 | .419 |
| h | 0.30 | 0.40 | .012 | .016 |
| L | 0.60 | 1.00 | .024 | .039 |
| Q1 | 0.97 | 1.07 | .038 | .042 |

CONTROLLING DIMENSIONS : MM
LEADS ARE COPLANAR WITHIN .004 INCH.

**Figure 43. 20-Pin Small Outline Integrated Circuit Package (SOIC)**

Figure 45 displays the 28-pin Plastic Dual Inline Package (PDIP) available for the Z8 Encore! XP F082A 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 45. 28-Pin Plastic Dual Inline Package (PDIP)**

| Part Number | Flash | RAM | NVDS | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Comparator | Temperature Sensor | Description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Z8 Encore! XP® F082A Series with 4 KB Flash, 10-Bit Analog-to-Digital Converter** | | | | | | | | | | | |
| **Standard Temperature: 0 °C to 70 °C** | | | | | | | | | | | |
| Z8F042APB020SC | 4 KB | 1 KB | 128 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | PDIP 8-pin package |
| Z8F042AQB020SC | 4 KB | 1 KB | 128 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | QFN 8-pin package |
| Z8F042ASB020SC | 4 KB | 1 KB | 128 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | SOIC 8-pin package |
| Z8F042ASH020SC | 4 KB | 1 KB | 128 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SOIC 20-pin package |
| Z8F042AHH020SC | 4 KB | 1 KB | 128 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SSOP 20-pin package |
| Z8F042APH020SC | 4 KB | 1 KB | 128 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | PDIP 20-pin package |
| Z8F042ASJ020SC | 4 KB | 1 KB | 128 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SOIC 28-pin package |
| Z8F042AHJ020SC | 4 KB | 1 KB | 128 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SSOP 28-pin package |
| Z8F042APJ020SC | 4 KB | 1 KB | 128 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | PDIP 28-pin package |
| **Extended Temperature: -40 °C to 105 °C** | | | | | | | | | | | |
| Z8F042APB020EC | 4 KB | 1 KB | 128 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | PDIP 8-pin package |
| Z8F042AQB020EC | 4 KB | 1 KB | 128 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | QFN 8-pin package |
| Z8F042ASB020EC | 4 KB | 1 KB | 128 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | SOIC 8-pin package |
| Z8F042ASH020EC | 4 KB | 1 KB | 128 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SOIC 20-pin package |
| Z8F042AHH020EC | 4 KB | 1 KB | 128 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SSOP 20-pin package |
| Z8F042APH020EC | 4 KB | 1 KB | 128 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | PDIP 20-pin package |
| Z8F042ASJ020EC | 4 KB | 1 KB | 128 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SOIC 28-pin package |
| Z8F042AHJ020EC | 4 KB | 1 KB | 128 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SSOP 28-pin package |
| Z8F042APJ020EC | 4 KB | 1 KB | 128 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | PDIP 28-pin package |
| Replace C with G for Lead-Free Packaging | | | | | | | | | | | |