



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

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

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

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

Details

| | |
|----------------------------|---|
| Product Status | Obsolete |
| Core Processor | eZ8 |
| Core Size | 8-Bit |
| Speed | 5MHz |
| Connectivity | IrDA, UART/USART |
| Peripherals | Brown-out Detect/Reset, LED, POR, PWM, WDT |
| Number of I/O | 16 |
| 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 | -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/z8f0413ph005ec |

| | |
|--|------------|
| Flash Information Area | 130 |
| Operation | 131 |
| Flash Operation Timing Using the Flash Frequency Registers | 133 |
| Flash Code Protection Against External Access | 133 |
| Flash Code Protection Against Accidental Program and Erasure | 133 |
| Byte Programming | 135 |
| Page Erase | 135 |
| Mass Erase | 135 |
| Flash Controller Bypass | 136 |
| Flash Controller Behavior in DEBUG Mode | 136 |
| Flash Control Register Definitions | 137 |
| Flash Control Register | 137 |
| Flash Status Register | 137 |
| Flash Page Select Register | 138 |
| Flash Sector Protect Register | 139 |
| Flash Frequency High and Low Byte Registers | 139 |
| Flash Option Bits | 141 |
| Operation | 141 |
| Option Bit Configuration By Reset | 141 |
| Option Bit Types | 141 |
| Reading the Flash Information Page | 143 |
| Flash Option Bit Control Register Definitions | 143 |
| Trim Bit Address Register | 143 |
| Trim Bit Data Register | 144 |
| Flash Option Bit Address Space | 144 |
| Flash Program Memory Address 0000H | 144 |
| Flash Program Memory Address 0001H | 145 |
| Trim Bit Address Space | 146 |
| Trim Bit Address 0000H—Reserved | 146 |
| Trim Bit Address 0001H—Reserved | 146 |
| Trim Bit Address 0002H | 147 |
| Trim Bit Address 0003H—Reserved | 147 |
| Trim Bit Address 0004H—Reserved | 147 |
| Zilog Calibration Data | 147 |
| ADC Calibration Data | 147 |
| Serialization Data | 148 |
| Randomized Lot Identifier | 149 |
| On-Chip Debugger | 151 |
| Architecture | 151 |

Internal Precision Oscillator

The internal precision oscillator (IPO) is a trimmable clock source that requires no external components.

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.

Analog Comparator

The analog comparator compares the signal at an input pin with either an internal programmable voltage reference or a second input pin. The comparator output can be used to drive either an output pin or to generate an interrupt.

Universal Asynchronous Receiver/Transmitter

The UART is full-duplex and capable of handling asynchronous data transfers. 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 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.

Interrupt Controller

Z8 Encore! XP[®] F0823 Series products support up to 20 interrupts. These interrupts consist of eight internal peripheral interrupts and 12 general-purpose I/O pin interrupt sources. The interrupts have three levels of programmable interrupt priority.

The pin configurations listed are preliminary and subject to change based on manufacturing limitations.

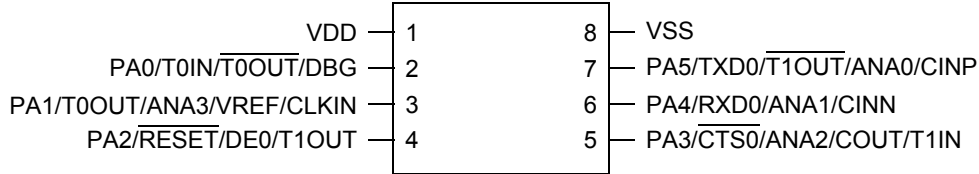


Figure 2. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 8-Pin SOIC, QFN/MLF-S, or PDIP Package*

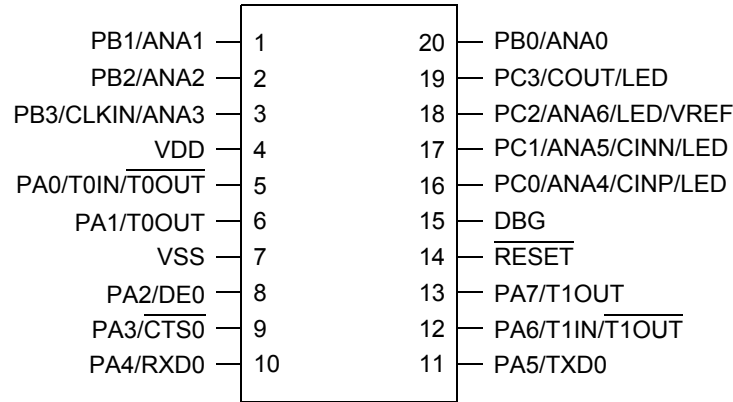


Figure 3. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 20-Pin SOIC, SSOP or PDIP Package*

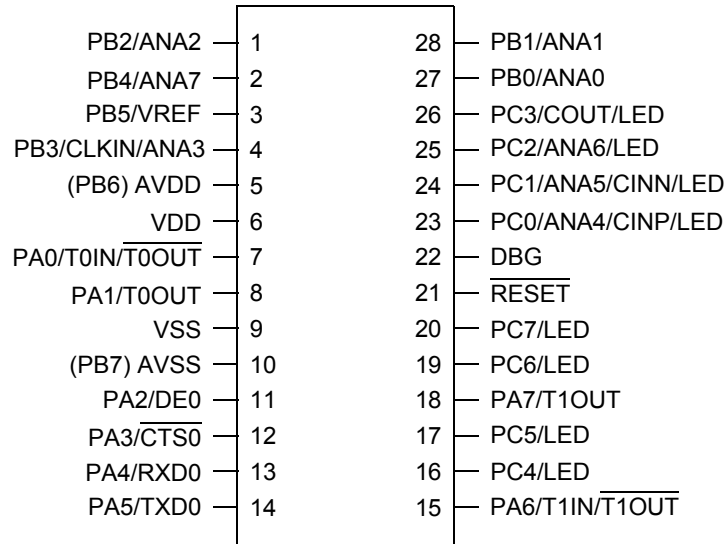


Figure 4. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 28-Pin SOIC, SSOP or PDIP Package*

Register Map

Table 8 lists the address map for the Register File of the Z8 Encore! XP[®] F0823 Series devices. Not all devices and package styles in the Z8 Encore! XP F0823 Series support the ADC, or all GPIO ports. Consider registers for unimplemented peripherals as reserved.

Table 8. Register File Address Map

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No |
|--------------------------------|-----------------------------------|----------|-------------|---------|
| General-Purpose RAM | | | | |
| Z8F0823/Z8F0813 Devices | | | | |
| 000–3FF | General-Purpose Register File RAM | — | XX | |
| 400–EFF | Reserved | — | XX | |
| Z8F0423/Z8F0413 Devices | | | | |
| 000–3FF | General-Purpose Register File RAM | — | XX | |
| 400–EFF | Reserved | — | XX | |
| Z8F0223/Z8F0213 Devices | | | | |
| 000–1FF | General-Purpose Register File RAM | — | XX | |
| 200–EFF | Reserved | — | XX | |
| Z8F0123/Z8F0113 Devices | | | | |
| 000–0FF | General-Purpose Register File RAM | — | XX | |
| 100–EFF | Reserved | — | XX | |
| Timer 0 | | | | |
| F00 | Timer 0 High Byte | T0H | 00 | 80 |
| F01 | Timer 0 Low Byte | T0L | 01 | 80 |
| F02 | Timer 0 Reload High Byte | T0RH | FF | 81 |
| F03 | Timer 0 Reload Low Byte | T0RL | FF | 81 |
| F04 | Timer 0 PWM High Byte | T0PWMH | 00 | 81 |
| F05 | Timer 0 PWM Low Byte | T0PWML | 00 | 82 |
| F06 | Timer 0 Control 0 | T0CTL0 | 00 | 82 |
| F07 | Timer 0 Control 1 | T0CTL1 | 00 | 83 |
| Timer 1 | | | | |
| F08 | Timer 1 High Byte | T1H | 00 | 80 |
| F09 | Timer 1 Low Byte | T1L | 01 | 80 |
| F0A | Timer 1 Reload High Byte | T1RH | FF | 81 |
| F0B | Timer 1 Reload Low Byte | T1RL | FF | 81 |

COMPARATOR COUNTER Mode

In COMPARATOR COUNTER mode, the timer counts input transitions from the analog comparator output. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the comparator output signal. In COMPARATOR COUNTER mode, the prescaler is disabled.

! Caution: *The frequency of the comparator output signal must not exceed one-fourth the system clock frequency.*

After reaching the Reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer Reload.

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:

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

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. Follow the steps below 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.
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 and initiate counting.

In COMPARE mode, the system clock always provides the timer input. The Compare time can be calculated by the following equation:

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

GATED Mode

In GATED mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is deasserted or a timer Reload occurs. To determine if a Timer Input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

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. When 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 (assuming the Timer Input signal remains asserted). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reset.

- 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. 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 TxCTL1 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.

Watchdog Timer Reload Byte registers (WDTU, WDTL, and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL register address produce no effect on the bits in the WDTCTL register. The locking mechanism prevents spurious writes to the Reload registers.

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

Table 58. Watchdog Timer Control Register (WDTCTL)

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|---------|---|---|---|---|---|---|---|
| FIELD | WDTUNLK | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | W | W | W | W | W | W | W | W |
| ADDR | FF0H | | | | | | | |

WDTUNLK—Watchdog Timer Unlock

The software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer reload registers.

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTL, WDTL) registers (Tables 59 through Table 61) form the 24-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes. The 24-bit reload value is {WDTU[7:0], WDTL[7:0], WDTL[7:0]}. Writing to these registers sets the appropriate Reload Value. Reading from these registers returns the current Watchdog Timer count value.

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

External Driver Enable

The UART provides a Driver Enable (DE) signal for off-chip bus transceivers. This feature reduces the software overhead associated with using a GPIO pin to control the transceiver when communicating on a multi-transceiver bus, such as RS-485.

Driver Enable is an active High signal that envelopes the entire transmitted data frame including parity and Stop bits as displayed in Figure 14. The Driver Enable signal asserts when a byte is written to the UART Transmit Data register. The Driver Enable signal asserts at least one UART bit period and no greater than two UART bit periods before the Start bit is transmitted. This allows a setup time to enable the transceiver. The Driver Enable signal deasserts one system clock period after the final Stop bit is transmitted. This one system clock delay allows both time for data to clear the transceiver before disabling it, as well as the ability to determine if another character follows the current character. In the event of back to back characters (new data must be written to the Transmit Data Register before the previous character is completely transmitted) the DE signal is not deasserted between characters. The `DEPOL` bit in the UART Control Register 1 sets the polarity of the Driver Enable signal.

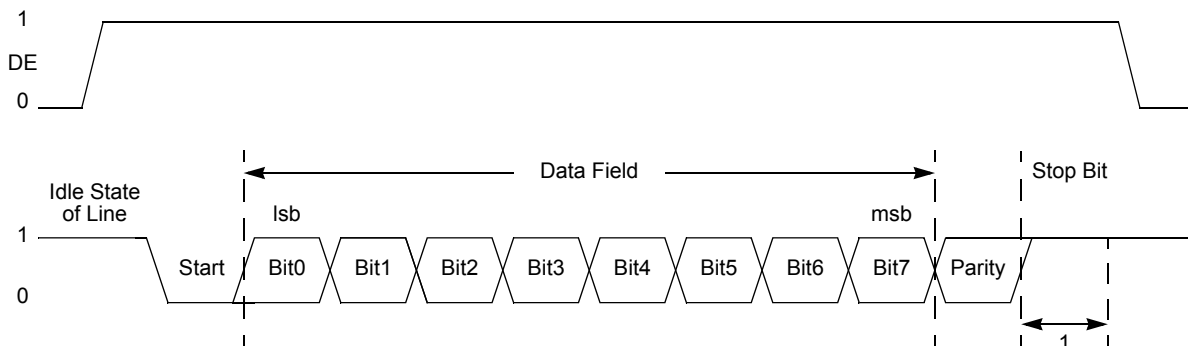


Figure 14. UART Driver Enable Signal Timing (shown with 1 Stop Bit and Parity)

The Driver Enable to Start bit setup time is calculated as follows:

$$\left(\frac{1}{\text{Baud Rate (Hz)}} \right) \leq \text{DE to Start Bit Setup Time (s)} \leq \left(\frac{2}{\text{Baud Rate (Hz)}} \right)$$

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.

Reserved—R/W bits must be 0 during writes; 0 when read.

NEWFRM—Status bit denoting the start of a new frame. Reading the UART Receive Data register resets this bit to 0.

0 = The current byte is not the first data byte of a new frame

1 = The current byte is the first data byte of a new frame

MPRX—Multiprocessor Receive

Returns the value of the most recent multiprocessor bit received. Reading from the UART Receive Data register resets this bit to 0.

UART Control 0 and Control 1 Registers

The UART Control 0 and Control 1 registers (Table 66 and Table 67) configure the properties of the UART's transmit and receive operations. The UART Control registers must not be written while the UART is enabled.

Table 66. UART Control 0 Register (U0CTL0)

| BITS | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|-----|------|-----|------|------|------|------|
| FIELD | TEN | REN | CTSE | PEN | PSEL | SBRK | STOP | LBEN |
| 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 | F42H | | | | | | | |

TEN—Transmit Enable

This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is low and the CTSE bit is 1, the transmitter is enabled.

0 = Transmitter disabled

1 = Transmitter enabled

REN—Receive Enable

This bit enables or disables the receiver.

0 = Receiver disabled

1 = Receiver enabled

CTSE—CTS Enable

0 = The $\overline{\text{CTS}}$ signal has no effect on the transmitter

1 = The UART recognizes the $\overline{\text{CTS}}$ signal as an enable control from the transmitter

PEN—Parity Enable

This bit enables or disables parity. Even or odd is determined by the PSEL bit.

0 = Parity is disabled

1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit

5. When the conversion is complete, the ADC control logic performs the following operations:
 - 11-bit two's-complement result written to {ADCD_H[7:0], ADCD_L[7:5]}.
 - 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.*

Follow the steps below 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.

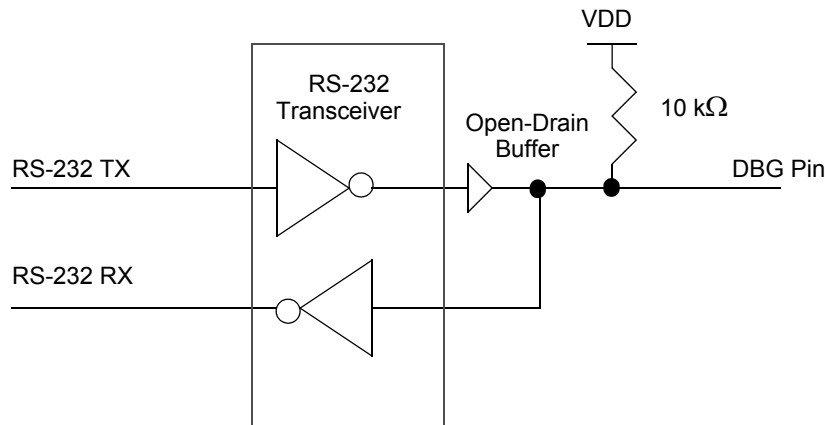


Figure 24. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (2)

DEBUG Mode

The operating characteristics of the 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 the operations below:

- The device enters DEBUG mode after the eZ8 CPU executes a BRK (breakpoint) instruction
- If the DBG pin is held Low during the most recent clock cycle of System Reset, the part enters DEBUG mode upon exiting System Reset

► **Note:** Holding the DBG pin Low for an additional 5000 (minimum) clock cycles after reset (making sure to account for any specified frequency error if using an internal oscillator) prevents a false interpretation of an Autobaud sequence (see OCD Auto-Baud Detector/Generator on page 154).

- 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

datastreams, the maximum recommended baud rate is the system clock frequency divided by eight. The maximum possible baud rate for asynchronous datastreams is the system clock frequency divided by four, but this theoretical maximum is possible only for low noise designs with clean signals. Table 98 lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 98. OCD Baud-Rate Limits

| System Clock Frequency (MHz) | Recommended Maximum Baud Rate (kbps) | Recommended Standard PC Baud Rate (bps) | Minimum Baud Rate (kbps) |
|------------------------------|--------------------------------------|---|--------------------------|
| 5.5296 | 1382.4 | 691,200 | 1.08 |
| 0.032768 (32 kHz) | 4.096 | 2400 | 0.064 |

If the OCD receives a Serial Break (nine or more continuous bits Low) the auto-baud detector/generator resets. Reconfigure the auto-baud detector/generator by sending 80H.

OCD Serial Errors

The OCD detects any of the following error conditions on the DBG pin:

- Serial Break (a minimum of nine continuous bits Low)
- Framing Error (received Stop bit is Low)
- Transmit Collision (OCD and host simultaneous transmission detected by the OCD)

When the OCD detects one of these errors, it aborts any command currently in progress, transmits a four character long Serial Break back to the host, and resets the auto-baud detector/generator. A Framing Error or Transmit Collision may be caused by the host sending a Serial Break to the OCD. Because of the open-drain nature of the interface, returning a Serial Break back to the host only extends the length of the Serial Break if the host releases the Serial Break early.

The host transmits a Serial Break on the DBG pin when first connecting to the Z8 Encore! XP F0823 Series devices or when recovering from an error. A Serial Break from the host resets the auto-baud generator/detector but does not reset the OCD Control register. A Serial Break leaves the device in DEBUG mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns High. Because of the open-drain nature of the DBG pin, the host sends a Serial Break to the OCD even if the OCD is transmitting a character.

Table 121. Internal Precision Oscillator Electrical Characteristics

| Symbol | Parameter | $V_{DD} = 2.7\text{ V to }3.6\text{ V}$ $T_A = -40\text{ }^{\circ}\text{C to }+105\text{ }^{\circ}\text{C}$ (unless otherwise stated) | | | Units | Conditions |
|-------------|--|---|---------|---------|---------------|---|
| | | Minimum | Typical | Maximum | | |
| F_{IPO} | Internal Precision Oscillator Frequency (High Speed) | | 5.53 | | MHz | $V_{DD} = 3.3\text{ V}$ $T_A = 30\text{ }^{\circ}\text{C}$ |
| F_{IPO} | Internal Precision Oscillator Frequency (Low Speed) | | 32.7 | | kHz | $V_{DD} = 3.3\text{ V}$ $T_A = 30\text{ }^{\circ}\text{C}$ |
| F_{IPO} | Internal Precision Oscillator Error | | ± 1 | ± 4 | % | |
| T_{IPOST} | Internal Precision Oscillator Startup Time | | 3 | | μs | |

UART Timing

Figure 32 and Table 130 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.

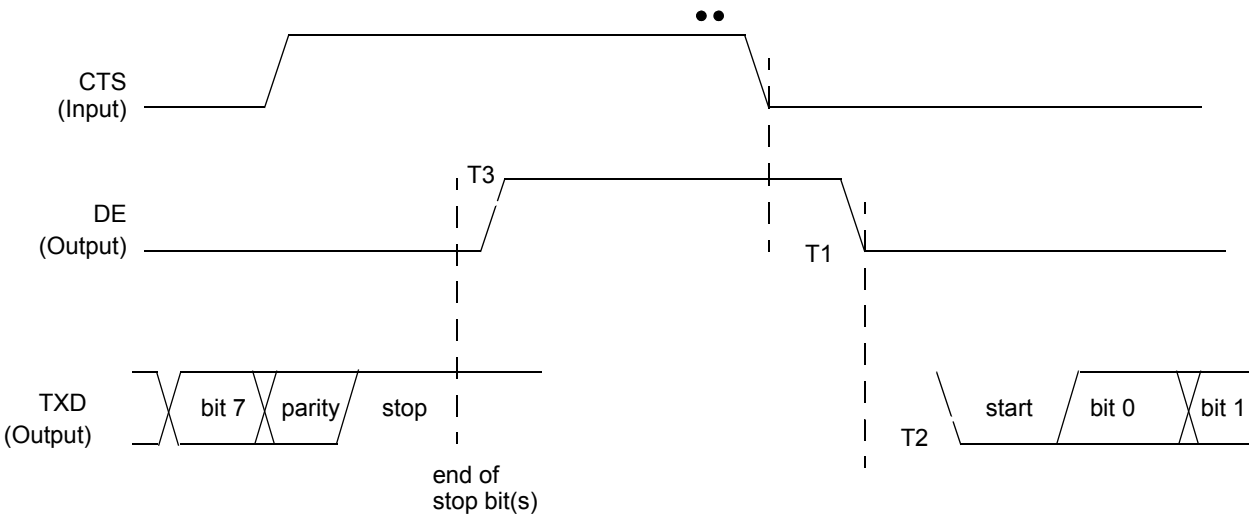


Figure 32. UART Timing With CTS

Table 130. UART Timing With CTS

| Parameter | Abbreviation | Delay (ns) | |
|----------------|--|----------------|-----------------------------|
| | | 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 | |

| 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 Development Kit | | | | | | | | |
| Z8F08A28100KITG | | | | | | | | Z8 Encore! XP F082A Series Development Kit (20- and 28-Pin) |
| Z8F04A28100KITG | | | | | | | | Z8 Encore! XP F042A Series Development Kit (20- and 28-Pin) |
| Z8F04A08100KITG | | | | | | | | Z8 Encore! XP F042A Series Development Kit (8-Pin) |
| ZUSBSC00100ZACG | | | | | | | | USB Smart Cable Accessory Kit |
| ZUSBOPTSC01ZACG | | | | | | | | Opto-Isolated USB Smart Cable Accessory Kit |
| ZENETSC0100ZACG | | | | | | | | Ethernet Smart Cable Accessory Kit |

Index

Symbols

174
 % 174
 @ 174

Numerics

10-bit ADC 4
 40-lead plastic dual-inline package 214, 215

A

absolute maximum ratings 193
 AC characteristics 197
 ADC 175
 architecture 117
 automatic power-down 118
 block diagram 118
 continuous conversion 120
 control register 122, 124
 control register definitions 122
 data high byte register 124
 data low bits register 125
 electrical characteristics and timing 201
 operation 118
 single-shot conversion 119
 ADCCTL register 122, 124
 ADCDH register 124
 ADCDL register 125
 ADCX 175
 ADD 175
 add - extended addressing 175
 add with carry 175
 add with carry - extended addressing 175
 additional symbols 174
 address space 13
 ADDX 175
 analog signals 10
 analog-to-digital converter (ADC) 117
 AND 177

ANDX 177
 arithmetic instructions 175
 assembly language programming 171
 assembly language syntax 172

B

B 174
 b 173
 baud rate generator, UART 103
 BCLR 176
 binary number suffix 174
 BIT 176
 bit 173
 clear 176
 manipulation instructions 176
 set 176
 set or clear 176
 swap 176
 test and jump 178
 test and jump if non-zero 178
 test and jump if zero 178
 bit jump and test if non-zero 178
 bit swap 178
 block diagram 3
 block transfer instructions 176
 BRK 178
 BSET 176
 BSWAP 176, 178
 BTJ 178
 BTJNZ 178
 BTJZ 178

C

CALL procedure 178
 CAPTURE mode 84, 85
 CAPTURE/COMPARE mode 85
 cc 173
 CCF 176
 characteristics, electrical 193
 clear 177
 CLR 177
 COM 177

COMPARE 84
 compare - extended addressing 175
 COMPARE mode 84
 compare with carry 175
 compare with carry - extended addressing 175
 complement 177
 complement carry flag 176
 condition code 173
 continuous conversion (ADC) 120
 CONTINUOUS mode 84
 control register definition, UART 104
 Control Registers 13, 17
 COUNTER modes 84
 CP 175
 CPC 175
 CPCX 175
 CPU and peripheral overview 4
 CPU control instructions 176
 CPX 175
 Customer Support 237

D

DA 173, 175
 data memory 15
 DC characteristics 194
 debugger, on-chip 151
 DEC 175
 decimal adjust 175
 decrement 175
 decrement and jump non-zero 178
 decrement word 175
 DECW 175
 destination operand 174
 device, port availability 35
 DI 176
 direct address 173
 disable interrupts 176
 DJNZ 178
 dst 174

E

EI 176

electrical characteristics 193
 ADC 201
 flash memory and timing 200
 GPIO input data sample timing 202
 Watchdog Timer 200, 202
 enable interrupt 176
 ER 173
 extended addressing register 173
 external pin reset 25
 eZ8 CPU features 4
 eZ8 CPU instruction classes 174
 eZ8 CPU instruction notation 172
 eZ8 CPU instruction set 171
 eZ8 CPU instruction summary 179

F

FCTL register 137, 143, 144
 features, Z8 Encore! 1
 first opcode map 190
 FLAGS 174
 flags register 174
 flash
 controller 4
 option bit address space 144
 option bit configuration - reset 141
 program memory address 0000H 144
 program memory address 0001H 145
 flash memory 129
 arrangement 130
 byte programming 135
 code protection 133
 configurations 129
 control register definitions 137, 143
 controller bypass 136
 electrical characteristics and timing 200
 flash control register 137, 143, 144
 flash option bits 134
 flash status register 137
 flow chart 132
 frequency high and low byte registers 139
 mass erase 135
 operation 131
 operation timing 133