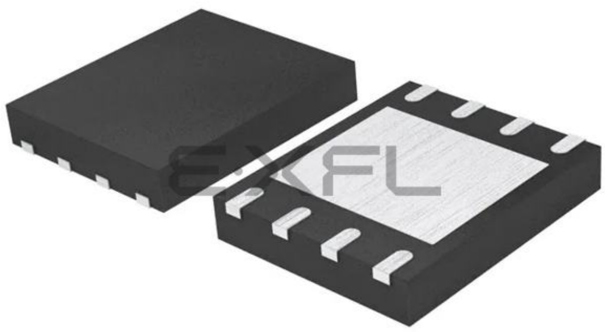

Welcome to E-XFL.COM

## 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 | Active |
| 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/z8f022aqb020eg |

# List of Figures

**Table 8. Reset and Stop Mode Recovery Characteristics and Latency**

| | Reset Characteristics and Latency | | |
|---|---|---|---|
| **Reset Type** | **Control Registers** | **eZ8 CPU** | **Reset Latency (Delay)** |
| System Reset | Reset (as applicable) | Reset | 66 Internal Precision Oscillator Cycles |
| System Reset with Crystal Oscillator Enabled | Reset (as applicable) | Reset | 5000 Internal Precision Oscillator Cycles |
| Stop Mode Recovery | Unaffected, except WDT_CTL and OSC_CTL registers | Reset | 66 Internal Precision Oscillator Cycles + IPO startup time |
| Stop Mode Recovery with Crystal Oscillator Enabled | Unaffected, except WDT_CTL and OSC_CTL registers | Reset | 5000 Internal Precision Oscillator Cycles |

During a System Reset or Stop Mode Recovery, the Internal Precision Oscillator requires 4 µs to start up. Then the Z8 Encore! XP F082A Series device is held in Reset for 66 cycles of the Internal Precision Oscillator. If the crystal oscillator is enabled in the Flash option bits, this reset period is increased to 5000 IPO cycles. When a reset occurs because of a low voltage condition or Power-On Reset (POR), this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the reset period, the device remains in reset until the pin is deasserted.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled, except PD0 (or PA2 on 8-pin devices) which is shared with the reset pin. On reset, the PD0 is configured as a bidirectional open-drain reset. The pin is internally driven low during port reset, after which the user code may reconfigure this pin as a general purpose output.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run.

Upon Reset, control registers within the Register File that 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 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.

As the control registers are reinitialized by a system reset, the system clock after reset is always the IPO. The software must reconfigure the oscillator control block, such that the correct system clock source is enabled and selected.

PA0 and PA6 contain two different timer functions, a timer input and a complementary timer output. Both of these functions require the same GPIO configuration, the selection between the two is based on the timer mode. See **the** Timers **chapter on page 70** for more details.

---

**!** **Caution:** For pins with multiple alternate functions, Zilog recommends writing to the AFS1 and AFS2 subregisters before enabling the alternate function via the AF subregister. As a result, spurious transitions through unwanted alternate function modes will be prevented.

---

## Direct LED Drive

The Port C pins provide a current sinked output capable of driving an LED without requiring an external resistor. The output sinks current at programmable levels of 3 mA, 7 mA, 13 mA and 20 mA. This mode is enabled through the LED control registers. The LED Drive Enable (LEDEN) Register turns on the drivers. The LED Drive Level (LEDLVLH and LEDLVLL) registers select the sink current.

For correct function, the LED anode must be connected to $V_{DD}$ and the cathode to the GPIO pin. Using all Port C pins in LED drive mode with maximum current may result in excessive total current. See **the** Electrical Characteristics chapter on page 226 for the maximum total current for the applicable package.

## Shared Reset Pin

On the 20- and 28-pin devices, the PD0 pin shares function with a bidirectional reset pin. Unlike all other I/O pins, this pin does not default to GPIO function on power-up. This pin acts as a bidirectional input/open-drain output reset until the software reconfigures it. The PD0 pin is an output-only open drain when in GPIO mode. There are no pull-up, High Drive, or Stop Mode Recovery source features associated with the PD0 pin.

On the 8-pin product versions, the reset pin is shared with PA2, but the pin is not limited to output-only when in GPIO mode.

---

**!** **Caution:** If PA2 on the 8-pin product is reconfigured as an input, ensure that no external stimulus drives the pin low during any reset sequence. Since PA2 returns to its $\overline{RESET}$ alternate function during system resets, driving it Low holds the chip in a reset state until the pin is released.

---

## Port A–D Control Registers

The Port A–D Control registers set the GPIO port operation. The value in the corresponding Port A–D Address Register determines which subregister is read from or written to by a Port A–D Control Register transaction; see Table 20.

**Table 20. Port A–D Control Registers (PxCTL)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | PCTL | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FD1H, FD5H, FD9H, FDDH | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>PCTLx | **Port Control**<br>The Port Control Register provides access to all subregisters that configure the GPIO port operation. |

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

## Port A–D Data Direction Subregisters

The Port A–D Data Direction subregister is accessed through the Port A–D Control Register by writing `01H` to the Port A–D Address Register; see Table 21.

**Table 21. Port A–D Data Direction Subregisters (PxDD)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | DD7 | DD6 | DD5 | DD4 | DD3 | DD2 | DD1 | DD0 |
| 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 | If 01H in Port A–D Address Register, accessible through the Port A–D Control Register | | | | | | | |

| Bit | Description |
|---|---|
| [7:0]<br>DDx | **Data Direction**<br>These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction Register setting.<br>0 = Output. Data in the Port A–D Output Data Register is driven onto the port pin.<br>1 = Input. The port pin is sampled and the value written into the Port A–D Input Data Register. The output driver is tristated. |

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

   – 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.

Observe the following steps for configuring a timer for GATED Mode and initiating the count:

1. Write to the Timer Control Register to:
   – Disable the timer

| Bit | Description (Continued) |
|---|---|
| [6]<br>TPOL<br>(cont'd) | **GATED Mode**<br>0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input.<br>1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input.<br>**CAPTURE/COMPARE Mode**<br>0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal.<br>1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.<br>**PWM DUAL OUTPUT Mode**<br>0 = Timer Output is forced Low (0) and Timer Output Complement is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon reload. When enabled, the Timer Output Complement is forced Low (0) upon PWM count match and forced High (1) upon reload. The PWMD field in TxCTL0 Register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to High (1).<br>1 = Timer Output is forced High (1) and Timer Output Complement is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon reload.When enabled, the Timer Output Complement is forced High (1) upon PWM count match and forced Low (0) upon reload. The PWMD field in TxCTL0 Register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to Low (0).<br>**CAPTURE RESTART Mode**<br>0 = Count is captured on the rising edge of the Timer Input signal.<br>1 = Count is captured on the falling edge of the Timer Input signal.<br>**COMPARATOR COUNTER Mode**<br>When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer Reload. Also:<br>0 = Count is captured on the rising edge of the comparator output.<br>1 = Count is captured on the falling edge of the comparator output.<br><br>**Caution:** When the Timer Output alternate function TxOUT on a GPIO port pin is enabled, TxOUT changes to whatever state the TPOL bit is in.The timer does not need to be enabled for that to happen. Also, the Port Data Direction Subregister is not required to be set to output on TxOUT. Changing the TPOL bit with the timer enabled and running does not immediately change the TxOUT. |

# Universal Asynchronous Receiver/ Transmitter

The universal asynchronous receiver/transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The UART uses a single 8-bit data mode with selectable parity. Features of the UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- Option of one or two STOP bits
- Separate transmit and receive interrupts
- Framing, parity, overrun and break detection
- Separate transmit and receive enables
- 16-bit baud rate generator (BRG)
- Selectable MULTIPROCESSOR (9-bit) Mode with three configurable interrupt schemes
- Baud rate generator (BRG) can be configured and used as a basic 16-bit timer
- Driver enable (DE) output for external bus transceivers

## Architecture

The UART consists of three primary functional blocks: transmitter, receiver and baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. Figure 10 displays the UART architecture.

# Low Power Operational Amplifier

The LPO is a general-purpose low power operational amplifier. Each of the three ports of the amplifier is accessible from the package pins. The LPO contains only one pin configuration: ANA0 is the output/feedback node, ANA1 is the inverting input and ANA2 is the noninverting input.

## Operation

To use the LPO, it must be enabled in the Power Control Register 0 (PWRCTL0). The default state of the LPO is OFF. To use the LPO, the LPO bit must be cleared by turning it ON (for details, see the Power Control Register 0 section on page 33). When making normal ADC measurements on ANA0 (i.e., measurements not involving the LPO output), the LPO bit must be turned OFF. Turning the LPO bit ON interferes with normal ADC measurements.

---

**!** **Caution:** The LPO bit enables the amplifier even in STOP Mode. If the amplifier is not required in STOP Mode, disable it. Failing to perform this results in STOP Mode currents higher than necessary.

---

As with other ADC measurements, any pins used for analog purposes must be configured as such in the GPIO registers. See the Port A–D Alternate Function Subregisters section on page 47 for details.

LPO output measurements are made on ANA0, as selected by the `ANAIN[3:0]` bits of ADC Control Register 0. It is also possible to make single-ended measurements on ANA1 and ANA2 while the amplifier is enabled, which is often useful for determining offset conditions. Differential measurements between ANA0 and ANA2 may be useful for noise cancellation purposes.

If the LPO output is routed to the ADC, then the `BUFFMODE`[2:0] bits of ADC Control/Status Register 1 must also be configured for unity-gain buffered operation. Sampling the LPO in an unbuffered mode is not recommended.

When either input is overdriven, the amplifier output saturates at the positive or negative supply voltage. No instability results.

In the above equation, T is the temperature in °C; V is the sensor output in volts.

Assuming a compensated ADC measurement, the following equation defines the relationship between the ADC reading and the die temperature:

$$T = (25/128) \times (ADC - TSCAL[11:2]) + 30$$

In the above equation, T is the temperature in C; ADC is the 10-bit compensated ADC value; and TSCAL is the temperature sensor calibration value, ignoring the two least significant bits of the 12-bit value.

See the Temperature Sensor Calibration Data section on page 171 for the location of TSCAL.

## Calibration

The temperature sensor undergoes calibration during the manufacturing process and is maximally accurate at 30°C. Accuracy decreases as measured temperatures move further from the calibration point.

**Table 107. NVDS Read Time**

| Operation | Minimum Latency | Maximum Latency |
|---|---|---|
| Read (16 byte array) | 875 | 9961 |
| Read (64 byte array) | 876 | 8952 |
| Read (128 byte array) | 883 | 7609 |
| Write (16 byte array) | 4973 | 5009 |
| Write (64 byte array) | 4971 | 5013 |
| Write (128 byte array) | 4984 | 5023 |
| Illegal Read | 43 | 43 |
| Illegal Write | 31 | 31 |

If NVDS read performance is critical to your software architecture, you can optimize your code for speed. Try the first suggestion below before attempting the second.

1. Periodically refresh all addresses that are used. The optimal use of NVDS in terms of speed is to rotate the writes evenly among all addresses planned to use, bringing all reads closer to the minimum read time. Because the minimum read time is much less than the write time, however, actual speed benefits are not always realized.
2. Use as few unique addresses as possible to optimize the impact of refreshing, plus minimize the requirement for it.

# Crystal Oscillator

The products in the Z8 Encore! XP F082A Series contain an on-chip crystal oscillator for use with external crystals with 32 kHz to 20 MHz frequencies. In addition, the oscillator supports external RC networks with oscillation frequencies up to 4 MHz or ceramic resonators with frequencies up to 8 MHz. The on-chip crystal oscillator can be used to generate the primary system clock for the internal eZ8 CPU and the majority of the on-chip peripherals. Alternatively, the $X_{IN}$ input pin can also accept a CMOS-level clock input signal (32 kHz–20 MHz). If an external clock generator is used, the $X_{OUT}$ pin must be left unconnected. The Z8 Encore! XP F082A Series products do not contain an internal clock divider. The frequency of the signal on the $X_{IN}$ input pin determines the frequency of the system clock.

**Note:** Although the $X_{IN}$ pin can be used as an input for an external clock generator, the CLKIN pin is better suited for such use (see the System Clock Selection section on page 193).

## Operating Modes

The Z8 Encore! XP F082A Series products support four oscillator modes:

- Minimum power for use with very low frequency crystals (32 kHz – 1 MHz)
- Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 8 MHz)
- Maximum power for use with high frequency crystals (8 MHz to 20 MHz)
- On-chip oscillator configured for use with external RC networks (<4 MHz)

The oscillator mode is selected via user-programmable Flash option bits. See **the** Flash Option Bits chapter on page 159 for information.

## Crystal Oscillator Operation

The XTLDIS Flash option bit controls whether the crystal oscillator is enabled during reset. The crystal may later be disabled after reset if a new oscillator has been selected as the system clock. If the crystal is manually enabled after reset through the OSCCTL Register, the user code must wait at least 1000 crystal oscillator cycles for the crystal to stabilize. After this, the crystal oscillator may be selected as the system clock.

**Note:** The stabilization time varies depending on the crystal, resonator or feedback network used. See Table 115 for transconductance values to compute oscillator stabilization times.

Figure 27 displays a recommended configuration for connection with an external fundamental-mode, parallel-resonant crystal operating at 20 MHz. Recommended 20 MHz crystal specifications are provided in Table 114. Printed circuit board layouts must add no more than 4 pF of stray capacitance to either the $X_{IN}$ or $X_{OUT}$ pins. If oscillation does not occur, reduce the values of capacitors C1 and C2 to decrease loading.

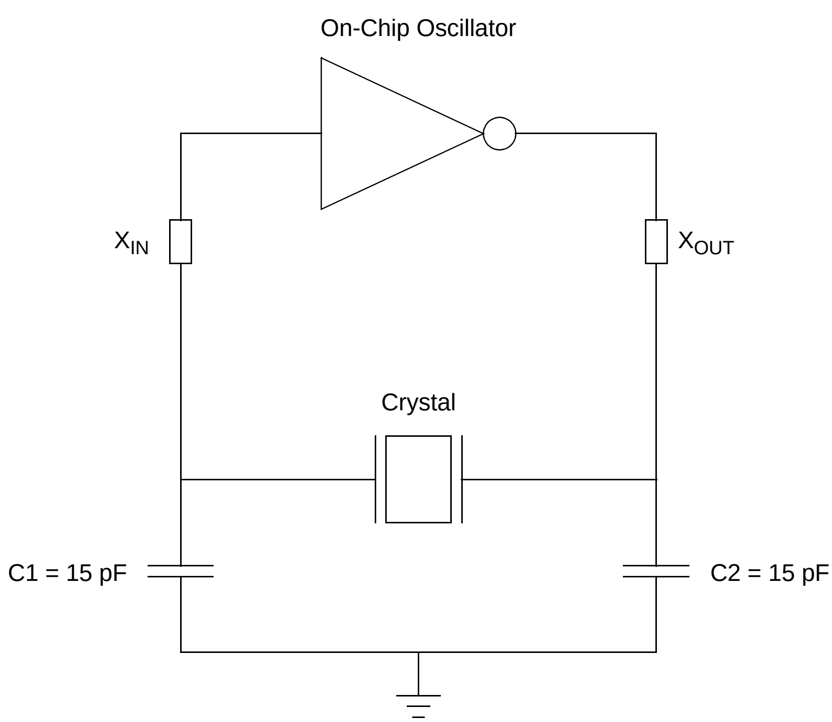


**Figure 27. Recommended 20 MHz Crystal Oscillator Configuration**

### Assembly Language Source Program Example

| | |
|---|---|
| `JP START` | ; Everything after the semicolon is a comment. |
| `START:` | ; A label called 'START'. The first instruction (`JP START`) in this ; example causes program execution to jump to the point within the ; program where the `START` label occurs. |
| `LD R4, R7` | ; A Load (LD) instruction with two operands. The first operand, ; Working Register R4, is the destination. The second operand, ; Working Register R7, is the source. The contents of R7 is ; written into R4. |
| `LD 234H, #%01` | ; Another Load (LD) instruction with two operands. ; The first operand, Extended Mode Register Address `234H`, ; identifies the destination. The second operand, Immediate Data ; value `01H`, is the source. The value `01H` is written into the ; Register at address `234H`. |

## Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as 'destination, source'. After assembly, the object code usually has the operands in the order 'source, destination', but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. This binary format must be followed if manual program coding is preferred or if you intend to implement your own assembler.

**Example 1.** If the contents of registers `43H` and `08H` are added and the result is stored in `43H`, the assembly syntax and resulting object code is:

**Table 116. Assembly Language Syntax Example 1**

| | | | | |
|---|---|---|---|---|
| **Assembly Language Code** | ADD | 43H, | 08H | (ADD dst, src) |
| **Object Code** | 04 | 08 | 43 | (OPC src, dst) |

**Example 2.** In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0–R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in `43H`, the assembly syntax and resulting object code is:

**Table 117. Assembly Language Syntax Example 2**

| | | | | |
|---|---|---|---|---|
| **Assembly Language Code** | ADD | 43H, | R8 | (ADD dst, src) |
| **Object Code** | 04 | E8 | 43 | (OPC src, dst) |

**Table 131. DC Characteristics (Continued)**

| Symbol | Parameter | $T_A$ = –40°C to +105°C (unless otherwise specified) Minimum | Typical | Maximum | Units | Conditions |
|---|---|---|---|---|---|---|
| $V_{OH1}$ | High Level Output Voltage | 2.4 | – | – | V | $I_{OH}$ = -2 mA; $V_{DD}$ = 3.0 V High Output Drive disabled. |
| $V_{OL2}$ | Low Level Output Voltage | – | – | 0.6 | V | $I_{OL}$ = 20 mA; $V_{DD}$ = 3.3V High Output Drive enabled. |
| $V_{OH2}$ | High Level Output Voltage | 2.4 | – | – | V | $I_{OH}$ = -20 mA; $V_{DD}$ = 3.3V High Output Drive enabled. |
| $I_{IH}$ | Input Leakage Current | – | $\pm$0.002 | $\pm$5 | µA | $V_{IN} = V_{DD}$ $V_{DD}$ = 3.3V; |
| $I_{IL}$ | Input Leakage Current | – | $\pm$0.007 | $\pm$5 | µA | $V_{IN} = V_{SS}$ $V_{DD}$ = 3.3V; |
| $I_{TL}$ | Tristate Leakage Current | – | – | $\pm$5 | µA | |
| $I_{LED}$ | Controlled Current Drive | 1.8 | 3 | 4.5 | mA | {AFS2,AFS1} = {0,0} |
| | | 2.8 | 7 | 10.5 | mA | {AFS2,AFS1} = {0,1} |
| | | 7.8 | 13 | 19.5 | mA | {AFS2,AFS1} = {1,0} |
| | | 12 | 20 | 30 | mA | {AFS2,AFS1} = {1,1} |
| $C_{PAD}$ | GPIO Port Pad Capacitance | – | 8.0[2] | – | pF | |
| $C_{XIN}$ | XIN Pad Capacitance | – | 8.0[2] | – | pF | |
| $C_{XOUT}$ | $X_{OUT}$ Pad Capacitance | – | 9.5[2] | – | pF | |
| $I_{PU}$ | Weak Pull-up Current | 30 | 100 | 350 | µA | $V_{DD}$ = 3.0 V–3.6 V |
| $V_{RAM}$ | RAM Data Retention Voltage | TBD | | | V | Voltage at which RAM retains static values; no reading or writing is allowed. |

Notes:
1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.

Figure 38 and Table 147 provide timing information for UART pins for the case where CTS is not used for flow control. DE asserts after the Transmit Data Register has been written. DE remains asserted for multiple characters as long as the Transmit Data Register is written with the next character before the current character has completed.

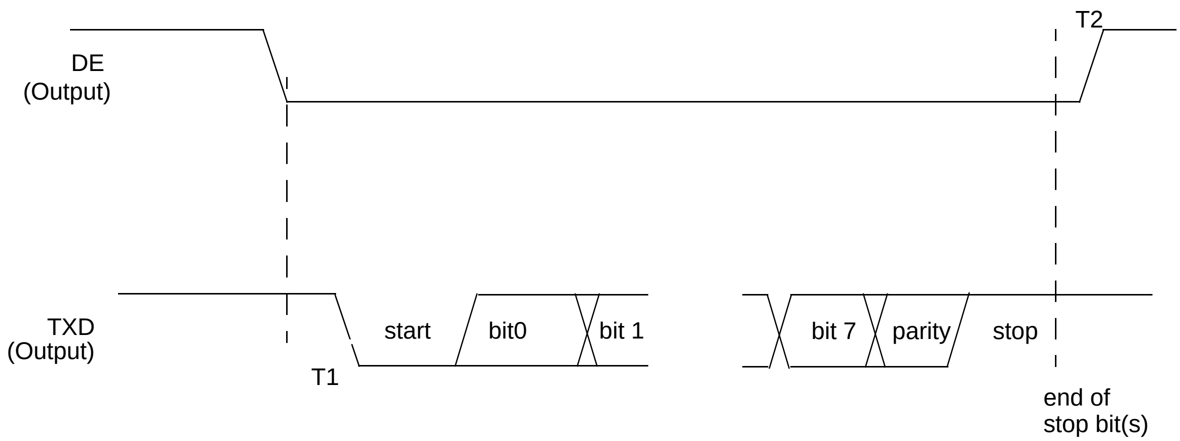


**Figure 38. UART Timing Without CTS**

**Table 147. UART Timing Without CTS**

| | | Delay (ns) | |
|---|---|---|---|
| **Parameter** | **Abbreviation** | **Minimum** | **Maximum** |
| **UART** | | | |
| $T_1$ | DE assertion to TXD falling edge (start bit) delay | 1 * $X_{IN}$ period | 1 bit time |
| $T_2$ | End of Stop Bit(s) to DE deassertion delay (Tx Data Register is empty) | ± 5 | |

**Table 148. Z8 Encore! XP F082A Series Ordering Matrix**

| 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 2 KB Flash, 10-Bit Analog-to-Digital Converter** | | | | | | | | | | | |
| **Standard Temperature: 0°C to 70°C** | | | | | | | | | | | |
| Z8F022APB020SG | 2 KB | 512 B | 64 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | PDIP 8-pin package |
| Z8F022AQB020SG | 2 KB | 512 B | 64 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | QFN 8-pin package |
| Z8F022ASB020SG | 2 KB | 512 B | 64 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | SOIC 8-pin package |
| Z8F022ASH020SG | 2 KB | 512 B | 64 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SOIC 20-pin package |
| Z8F022AHH020SG | 2 KB | 512 B | 64 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SSOP 20-pin package |
| Z8F022APH020SG | 2 KB | 512 B | 64 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | PDIP 20-pin package |
| Z8F022ASJ020SG | 2 KB | 512 B | 64 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SOIC 28-pin package |
| Z8F022AHJ020SG | 2 KB | 512 B | 64 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SSOP 28-pin package |
| Z8F022APJ020SG | 2 KB | 512 B | 64 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | PDIP 28-pin package |
| **Extended Temperature: –40°C to 105°C** | | | | | | | | | | | |
| Z8F022APB020EG | 2 KB | 512 B | 64 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | PDIP 8-pin package |
| Z8F022AQB020EG | 2 KB | 512 B | 64 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | QFN 8-pin package |
| Z8F022ASB020EG | 2 KB | 512 B | 64 B | 6 | 14 | 2 | 4 | 1 | 1 | 1 | SOIC 8-pin package |
| Z8F022ASH020EG | 2 KB | 512 B | 64 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SOIC 20-pin package |
| Z8F022AHH020EG | 2 KB | 512 B | 64 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | SSOP 20-pin package |
| Z8F022APH020EG | 2 KB | 512 B | 64 B | 17 | 20 | 2 | 7 | 1 | 1 | 1 | PDIP 20-pin package |
| Z8F022ASJ020EG | 2 KB | 512 B | 64 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SOIC 28-pin package |
| Z8F022AHJ020EG | 2 KB | 512 B | 64 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | SSOP 28-pin package |
| Z8F022APJ020EG | 2 KB | 512 B | 64 B | 23 | 20 | 2 | 8 | 1 | 1 | 1 | PDIP 28-pin package |

**Table 148. Z8 Encore! XP F082A Series Ordering Matrix**

| 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 2 KB Flash** | | | | | | | | | | | |
| **Standard Temperature: 0°C to 70°C** | | | | | | | | | | | |
| Z8F021APB020SG | 2 KB | 512 B | 64 B | 6 | 13 | 2 | 0 | 1 | 1 | 0 | PDIP 8-pin package |
| Z8F021AQB020SG | 2 KB | 512 B | 64 B | 6 | 13 | 2 | 0 | 1 | 1 | 0 | QFN 8-pin package |
| Z8F021ASB020SG | 2 KB | 512 B | 64 B | 6 | 13 | 2 | 0 | 1 | 1 | 0 | SOIC 8-pin package |
| Z8F021ASH020SG | 2 KB | 512 B | 64 B | 17 | 19 | 2 | 0 | 1 | 1 | 0 | SOIC 20-pin package |
| Z8F021AHH020SG | 2 KB | 512 B | 64 B | 17 | 19 | 2 | 0 | 1 | 1 | 0 | SSOP 20-pin package |
| Z8F021APH020SG | 2 KB | 512 B | 64 B | 17 | 19 | 2 | 0 | 1 | 1 | 0 | PDIP 20-pin package |
| Z8F021ASJ020SG | 2 KB | 512 B | 64 B | 25 | 19 | 2 | 0 | 1 | 1 | 0 | SOIC 28-pin package |
| Z8F021AHJ020SG | 2 KB | 512 B | 64 B | 25 | 19 | 2 | 0 | 1 | 1 | 0 | SSOP 28-pin package |
| Z8F021APJ020SG | 2 KB | 512 B | 64 B | 25 | 19 | 2 | 0 | 1 | 1 | 0 | PDIP 28-pin package |
| **Extended Temperature: –40°C to 105°C** | | | | | | | | | | | |
| Z8F021APB020EG | 2 KB | 512 B | 64 B | 6 | 13 | 2 | 0 | 1 | 1 | 0 | PDIP 8-pin package |
| Z8F021AQB020EG | 2 KB | 512 B | 64 B | 6 | 13 | 2 | 0 | 1 | 1 | 0 | QFN 8-pin package |
| Z8F021ASB020EG | 2 KB | 512 B | 64 B | 6 | 13 | 2 | 0 | 1 | 1 | 0 | SOIC 8-pin package |
| Z8F021ASH020EG | 2 KB | 512 B | 64 B | 17 | 19 | 2 | 0 | 1 | 1 | 0 | SOIC 20-pin package |
| Z8F021AHH020EG | 2 KB | 512 B | 64 B | 17 | 19 | 2 | 0 | 1 | 1 | 0 | SSOP 20-pin package |
| Z8F021APH020EG | 2 KB | 512 B | 64 B | 17 | 19 | 2 | 0 | 1 | 1 | 0 | PDIP 20-pin package |
| Z8F021ASJ020EG | 2 KB | 512 B | 64 B | 25 | 19 | 2 | 0 | 1 | 1 | 0 | SOIC 28-pin package |
| Z8F021AHJ020EG | 2 KB | 512 B | 64 B | 25 | 19 | 2 | 0 | 1 | 1 | 0 | SSOP 28-pin package |
| Z8F021APJ020EG | 2 KB | 512 B | 64 B | 25 | 19 | 2 | 0 | 1 | 1 | 0 | PDIP 28-pin package |

## P



## R