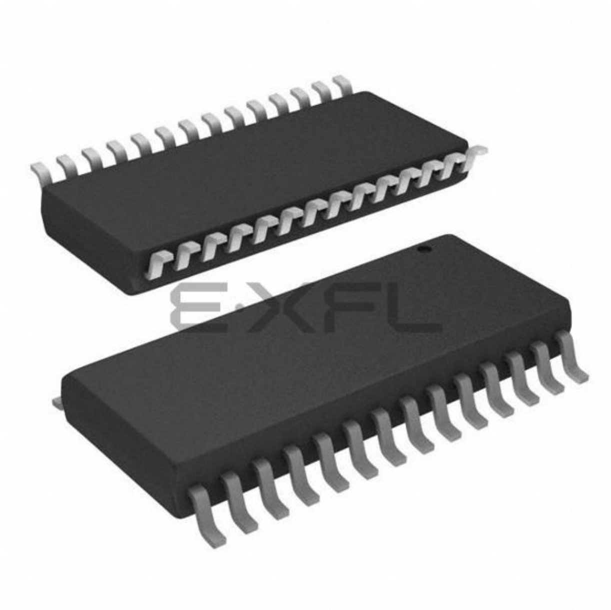

Welcome to E-XFL.COM

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | eZ8 |
| Core Size | 8-Bit |
| Speed | 20MHz |
| Connectivity | IrDA, UART/USART |
| Peripherals | Brown-out Detect/Reset, LED, LVD, POR, PWM, WDT |
| Number of I/O | 25 |
| Program Memory Size | 4KB (4K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 128 x 8 |
| 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 | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z8f041asj020eg |

LED Drive Level Low Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

GPIO Mode Interrupt Controller . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
Interrupt Vector Listing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
Operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
Master Interrupt Enable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
Interrupt Vectors and Priority . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
Interrupt Assertion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
Software Interrupt Assertion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
Watchdog Timer Interrupt Assertion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
Interrupt Control Register Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
Interrupt Request 0 Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
Interrupt Request 1 Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
Interrupt Request 2 Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
IRQ0 Enable High and Low Bit Registers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
IRQ1 Enable High and Low Bit Registers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
IRQ2 Enable High and Low Bit Registers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
Interrupt Edge Select Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
Shared Interrupt Select Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
Interrupt Control Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

Timers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
Operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
Timer Operating Modes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
Reading the Timer Count Values . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
Timer Pin Signal Operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
Timer Control Register Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
Timer 0–1 Control Registers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
Timer 0–1 High and Low Byte Registers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
Timer Reload High and Low Byte Registers . . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
Timer 0–1 PWM High and Low Byte Registers . . . . . . . . . . . . . . . . . . . . . . . . . . 92

Watchdog Timer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
Operation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
Watchdog Timer Refresh . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
Watchdog Timer Time-Out Response . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
Watchdog Timer Reload Unlock Sequence . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
Watchdog Timer Calibration . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95
Watchdog Timer Control Register Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
Watchdog Timer Control Register . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
Watchdog Timer Reload Upper, High and Low Byte Registers . . . . . . . . . . . . . . . 97

# Low-Power Modes

The Z8 Encore! XP F082A Series products contain power-saving features. The highest level of power reduction is provided by the STOP Mode, in which nearly all device functions are powered down. The next lower level of power reduction is provided by the HALT Mode, in which the CPU is powered down.

Further power savings can be implemented by disabling individual peripheral blocks while in Active mode (defined as being in neither STOP nor HALT Mode).

## STOP Mode

Executing the eZ8 CPU's STOP instruction places the device into STOP Mode, powering down all peripherals except the Voltage Brown-Out detector, the Low-power Operational Amplifier and the Watchdog Timer. These three blocks may also be disabled for additional power savings. Specifically, the operating characteristics are:

- Primary crystal oscillator and internal precision oscillator are stopped; $X_{IN}$ and $X_{OUT}$ (if previously enabled) are disabled and PA0/PA1 revert to the states programmed by the GPIO registers
- System clock is stopped
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- Watchdog Timer's internal RC oscillator continues to operate if enabled by the Oscillator Control Register
- If enabled, the Watchdog Timer logic continues to operate
- If enabled for operation in STOP Mode by the associated Flash option bit, the Voltage Brown-Out protection circuit continues to operate
- Low-power operational amplifier continues to operate if enabled by the Power Control Register
- All other on-chip peripherals are idle

To minimize current in STOP Mode, all GPIO pins that are configured as digital inputs must be driven to one of the supply rails ($V_{CC}$ or GND). Additionally, any GPIOs configured as outputs must also be driven to one of the supply rails. The device can be brought out of STOP Mode using Stop Mode Recovery. For more information about Stop Mode Recovery, see the Reset, Stop Mode Recovery and Low Voltage Detection chapter on page 22.

---

**Note:** Asserting any power control bit disables the targeted block regardless of any enable bits contained in the target block's control registers.

---

# Timers

These Z8 Encore! XP F082A Series products contain two 16-bit reloadable timers that can be used for timing, event counting, or generation of pulse-width modulated (PWM) signals. The timers' feature include:

- 16-bit reload counter
- Programmable prescaler with prescale values from 1 to 128
- PWM output generation
- Capture and compare capability
- External input pin for timer input, clock gating, or capture signal. External input pin signal frequency is limited to a maximum of one-fourth the system clock frequency
- Timer output pin
- Timer interrupt

In addition to the timers described in this chapter, the Baud Rate Generator of the UART (if unused) may also provide basic timing functionality. For information about using the Baud Rate Generator as an additional timer, see the Universal Asynchronous Receiver/Transmitter chapter on page 99.

## Architecture

Figure 9 displays the architecture of the timers.

it is appropriate to have the Timer Output make a state change at a One-Shot time-out (rather than a single cycle pulse), first set the TPOL bit in the Timer Control Register to the start value before enabling ONE-SHOT Mode. After starting the timer, set `TPOL` to the opposite bit value.

Observe the following steps for configuring a timer for ONE-SHOT Mode and initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for ONE-SHOT Mode.
   - Set the prescale value.
   - Set the initial output level (High or Low) if using the Timer Output alternate function.
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 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 and initiate counting.

In ONE-SHOT Mode, the system clock always provides the timer input. The timer period is computed via the following equation:

$$\text{ONE-SHOT Mode Time-Out Period (s)} = \frac{\text{Reload Value} - \text{Start Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

### CONTINUOUS Mode

In CONTINUOUS 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, 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.

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

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CONTINUOUS Mode

  - Configure the timer for GATED Mode
  - Set the prescale value

2. Write to the Timer High and Low Byte registers to set the starting count value. Writing these registers only affects the first pass in GATED Mode. After the first timer reset in GATED Mode, counting always begins at the reset value of `0001H`.
3. Write to the Timer Reload High and Low Byte registers to set the reload 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 is generated for both input deassertion and reload events. If appropriate, configure the timer interrupt to be generated only at the input deassertion 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. Assert the Timer Input signal to initiate the counting.

## CAPTURE/COMPARE Mode

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

Every subsequent acceptable transition (after the first) of the Timer Input signal captures the current count value. The Capture value is written to the Timer PWM High and Low Byte registers. When the Capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The `INPCAP` bit in TxCTL0 Register is set to indicate the timer interrupt is caused by an input capture event.

If no Capture event occurs, the timer counts up to the 16-bit Compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the Compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. The `INPCAP` bit in TxCTL0 Register is cleared to indicate the timer interrupt is not because of an input capture event.

Observe the following steps for configuring a timer for CAPTURE/COMPARE Mode and initiating the count:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE/COMPARE Mode
   - Set the prescale value

  – Set the Capture edge (rising or falling) for the Timer Input

2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
3. Write to the Timer Reload High and Low Byte registers to set the Compare value.
4. 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

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

| 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. |

# Infrared Encoder/Decoder

Z8 Encore! XP F082A Series products contain a fully-functional, high-performance UART to Infrared Encoder/Decoder (endec). The infrared endec is integrated with an on-chip UART to allow easy communication between the Z8 Encore! XP MCU and IrDA Physical Layer Specification, Version 1.3-compliant infrared transceivers. Infrared communication provides secure, reliable, low-cost, point-to-point communication between PCs, PDAs, cell phones, printers and other infrared enabled devices.

## Architecture

Figure 16 displays the architecture of the infrared endec.

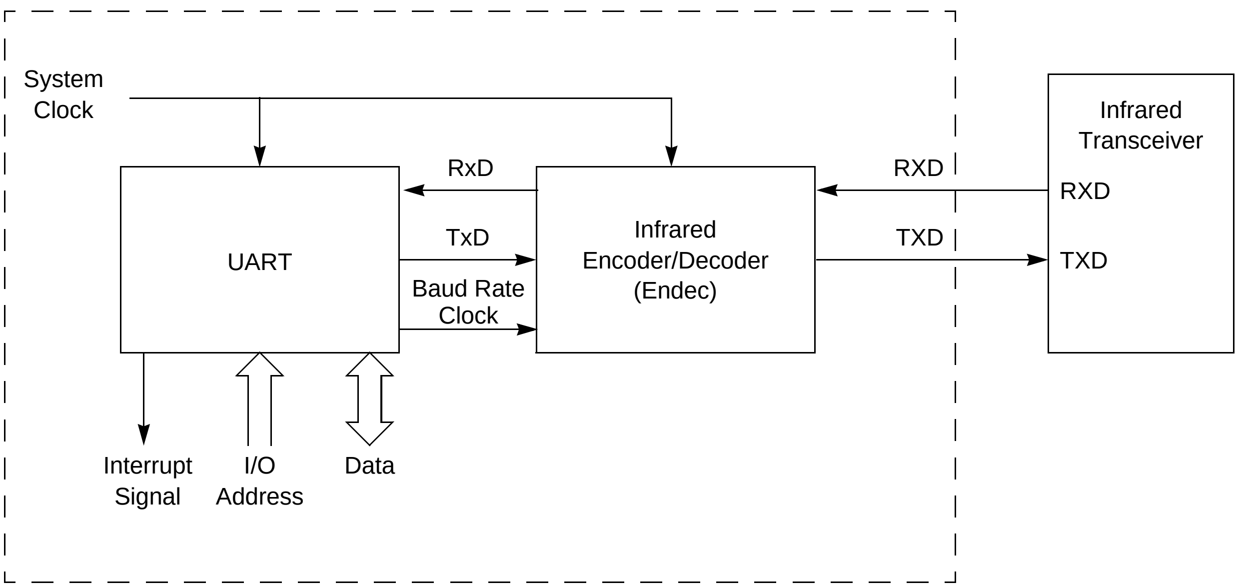


**Figure 16. Infrared Data Communication System Block Diagram**

## Operation

When the infrared endec is enabled, the transmit data from the associated on-chip UART is encoded as digital signals in accordance with the IrDA standard and output to the infrared transceiver through the TXD pin. Likewise, data received from the infrared transceiver is passed to the infrared endec through the RXD pin, decoded by the infrared endec and passed to the UART. Communication is half-duplex, which means simultaneous data transmission and reception is not allowed.

# Analog-to-Digital Converter

The analog-to-digital converter (ADC) converts an analog input signal to its digital representation. The features of this sigma-delta ADC include:

- 11-bit resolution in DIFFERENTIAL Mode
- 10-bit resolution in SINGLE-ENDED Mode
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports
- $9^{th}$ analog input obtained from temperature sensor peripheral
- 11 pairs of differential inputs also multiplexed with general-purpose I/O ports
- Low-power operational amplifier (LPO)
- Interrupt on conversion complete
- Bandgap generated internal voltage reference with two selectable levels
- Manual in-circuit calibration is possible employing user code (offset calibration)
- Factory calibrated for in-circuit error compensation

## Architecture

Figure 19 displays the major functional blocks of the ADC. An analog multiplexer network selects the ADC input from the available analog pins, ANA0 through ANA7.

The input stage of the ADC allows both differential gain and buffering. The following input options are available:

- Unbuffered input (SINGLE-ENDED and DIFFERENTIAL modes)
- Buffered input with unity gain (SINGLE-ENDED and DIFFERENTIAL modes)
- LPO output with full pin access to the feedback path

For the reserved values, all input switches are disabled to avoid leakage or other undesirable operation. ADC samples taken with reserved bit settings are undefined.

SINGLE-ENDED Mode:

0000 = ANA0 (transimpedance amp output when enabled)
0001 = ANA1 (transimpedance amp inverting input)
0010 = ANA2 (transimpedance amp noninverting input)
0011 = ANA3
0100 = ANA4
0101 = ANA5
0110 = ANA6
0111 = ANA7
1000 = Reserved
1001 = Reserved
1010 = Reserved
1011 = Reserved
1100 = Hold transimpedance input nodes (ANA1 and ANA2) to ground.
1101 = Reserved
1110 = Temperature Sensor.
1111 = Reserved.

DIFFERENTIAL Mode (noninverting input and inverting input respectively):

0000 = ANA0 and ANA1
0001 = ANA2 and ANA3
0010 = ANA4 and ANA5
0011 = ANA1 and ANA0
0100 = ANA3 and ANA2
0101 = ANA5 and ANA4
0110 = ANA6 and ANA5
0111 = ANA0 and ANA2
1000 = ANA0 and ANA3
1001 = ANA0 and ANA4
1010 = ANA0 and ANA5
1011 = Reserved
1100 = Reserved
1101 = Reserved
1110 = Reserved
1111 = Manual Offset Calibration Mode

## ADC Control/Status Register 1

The ADC Control/Status Register 1 (ADCCTL1) configures the input buffer stage, enables the threshold interrupts and contains the status of both threshold triggers. It is also used to select the voltage reference configuration.

**Table 74. ADC Control/Status Register 1 (ADCCTL1)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Field | REFSELH | Reserved | | | | BUFMODE[2:0] | | |
| RESET | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F71H | | | | | | | |

| Bit | Description |
|---|---|
| [7]<br>REFSELH | **Voltage Reference Level Select High Bit**<br>In conjunction with the Low bit (REFSELL) in ADC Control Register 0, this determines the level of the internal voltage reference; the following details the effects of {REFSELH, REFSELL}; this reference is independent of the Comparator reference.<br>00= Internal Reference Disabled, reference comes from external pin.<br>01= Internal Reference set to 1.0V.<br>10= Internal Reference set to 2.0V (default).<br>11= Reserved. |
| [6:3] | **Reserved**<br>These bits are reserved and must be programmed to 0000. |
| [2:0]<br>BUFMODE[2:0] | **Input Buffer Mode Select**<br>000 = Single-ended, unbuffered input.<br>001 = Single-ended, buffered input with unity gain.<br>010 = Reserved.<br>011 = Reserved.<br>100 = Differential, unbuffered input.<br>101 = Differential, buffered input with unity gain.<br>110 = Reserved.<br>111 = Reserved. |

## ADC Data High Byte Register

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

# *Temperature Sensor*

The on-chip Temperature Sensor allows you to measure temperature on the die with either the on-board ADC or on-board comparator. This block is factory calibrated for in-circuit software correction. Uncalibrated accuracy is significantly worse, therefore the temperature sensor is not recommended for uncalibrated use.

## Temperature Sensor Operation

The on-chip temperature sensor is a Proportional to Absolute Temperature (PTAT) topology. A pair of Flash option bytes contain the calibration data. The temperature sensor can be disabled by a bit in the Power Control Register 0 section on page 33 to reduce power consumption.

The temperature sensor can be directly read by the ADC to determine the absolute value of its output. The temperature sensor output is also available as an input to the comparator for threshold type measurement determination. The accuracy of the sensor when used with the comparator is substantially less than when measured by the ADC.

If the temperature sensor is routed to the ADC, the ADC must be configured in unity-gain buffered mode (for details, see the Input Buffer Stage section on page 133). The value read back from the ADC is a signed number, although it is always positive.

The sensor is factory-trimmed through the ADC using the external 2.0 V reference. Unless the sensor is retrimmed for use with a different reference, it is most accurate when used with the external 2.0 V reference.

Because this sensor is an on-chip sensor, Zilog recommends that the user account for the difference between ambient and die temperature when inferring ambient temperature conditions.

During normal operation, the die undergoes heating that causes a mismatch between the ambient temperature and that measured by the sensor. For best results, the Z8 Encore! XP device must be placed into STOP Mode for sufficient time such that the die and ambient temperatures converge (this time is dependent on the thermal design of the system). The temperature sensor measurement must then be made immediately after recovery from STOP Mode.

The following equation defines the transfer function between the temperature sensor output voltage and the die temperature. This is needed for comparator threshold measurements.

$$V = 0.01 \times T + 0.65$$

## Flash Operation Timing Using the Flash Frequency Registers

Before performing either a program or erase operation on Flash memory, you must first configure the Flash Frequency High and Low Byte registers. The Flash Frequency registers allow programming and erasing of the Flash with system clock frequencies ranging from 32 kHz (32768 Hz) through 20 MHz.

The Flash Frequency High and Low Byte registers combine to form a 16-bit value, `FFREQ`, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz). This value is calculated using the following equation:

$$\text{FFREQ[15:0]} = \frac{\text{System Clock Frequency (Hz)}}{1000}$$

---

**! Caution:** Flash programming and erasure are not supported for system clock frequencies below 32 kHz (32768 Hz) or above 20 MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure operation of the Z8 Encore! XP F082A Series devices.

---

## Flash Code Protection Against External Access

The user code contained within the Flash memory can be protected against external access by the on-chip debugger. Programming the FRP Flash option bit prevents reading of the user code with the On-Chip Debugger. See the Flash Option Bits chapter on page 159 and the On-Chip Debugger chapter on page 180 for more information.

## Flash Code Protection Against Accidental Program and Erasure

The Z8 Encore! XP F082A Series provides several levels of protection against accidental program and erasure of the Flash memory contents. This protection is provided by a combination of the Flash option bits, the register locking mechanism, the page select redundancy and the sector level protection control of the Flash Controller.

### Flash Code Protection Using the Flash Option Bits

The FRP and FWP Flash option bits combine to provide three levels of Flash Program Memory protection, as shown in Table 79. See the Flash Option Bits chapter on page 159 for more information.

**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.

# Operation

This section describes the interface and modes of operation of the On-Chip Debugger.

## OCD Interface

The on-chip debugger uses the DBG pin for communication with an external host. This one-pin interface is a bidirectional, open-drain interface that transmits and receives data. Data transmission is half-duplex, in that transmit and receive cannot occur simultaneously. The serial data on the DBG pin is sent using the standard asynchronous data format defined in RS-232. This pin creates an interface from the Z8 Encore! XP F082A Series products to the serial port of a host PC using minimal external hardware.Two different methods for connecting the DBG pin to an RS-232 interface are displayed in Figure 24 and Figure 25. The recommended method is the buffered implementation displayed in Figure 25. The DBG pin has a internal pull-up resistor which is sufficient for some applications (for more details about the pull-up current, see the Electrical Characteristics chapter on page 226). For OCD operation at higher data rates or in noisy systems, an external pull-up resistor is recommended.

**!** **Caution:** For operation of the on-chip debugger, all power pins ($V_{DD}$ and $AV_{DD}$) must be supplied with power and all ground pins ($V_{SS}$ and $AV_{SS}$) must be properly grounded. The DBG pin is open-drain and may require an external pull-up resistor to ensure proper operation.

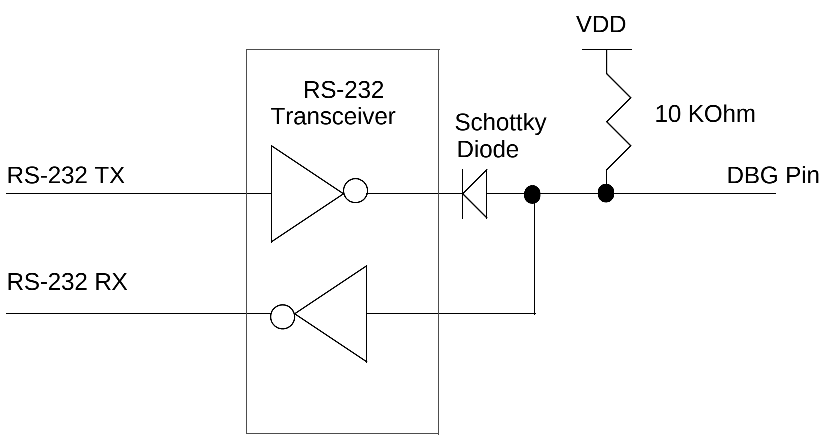


**Figure 24. Interfacing the On-Chip Debugger’s DBG Pin with an RS-232 Interface; #1 of 2**

# Oscillator Control

The Z8 Encore! XP F082A Series devices uses five possible clocking schemes, each user-selectable:

- Internal precision trimmed RC oscillator (IPO)
- On-chip oscillator using off-chip crystal or resonator
- On-chip oscillator using external RC network
- External clock drive
- On-chip low power Watchdog Timer oscillator
- Clock failure detection circuitry

In addition, Z8 Encore! XP F082A Series devices contain clock failure detection and recovery circuitry, allowing continued operation despite a failure of the system clock oscillator.

## Operation

This chapter discusses the logic used to select the system clock and handle primary oscillator failures.

### System Clock Selection

The oscillator control block selects from the available clocks. Table 112 details each clock source and its usage.

**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.

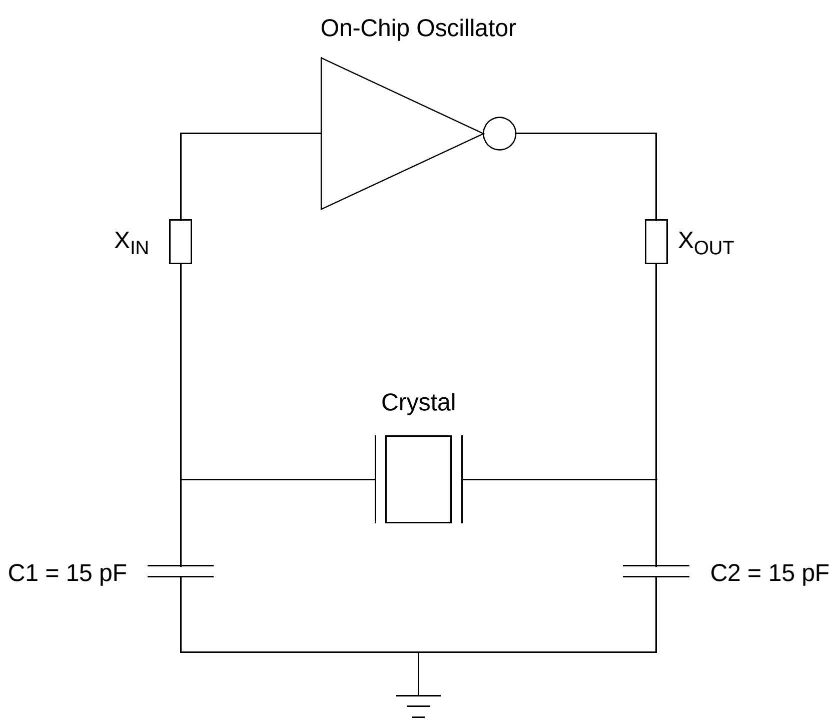


**Figure 27. Recommended 20 MHz Crystal Oscillator Configuration**

**Table 132. Power Consumption (Continued)**

| Symbol | Parameter | $V_{DD}$ = 2.7 V to 3.6 V | | | Units | Conditions |
|---|---|---|---|---|---|---|
| | | Typical[1] | Maximum Std Temp[2] | Maximum Ext Temp[3] | | |
| $I_{DD}$ ADCRef | ADC Internal Reference Supply Current | 0 | | | µA | See Note 4. |
| $I_{DD}$ CMP | Comparator supply Current | 150 | 180 | 190 | µA | See Note 4. |
| $I_{DD}$ LPO | Low-Power Operational Amplifier Supply Current | 3 | 5 | 5 | µA | Driving a high-impedance load. |
| $I_{DD}$ TS | Temperature Sensor Supply Current | 60 | | | µA | See Note 4. |
| $I_{DD}$ BG | Band Gap Supply Current | 320 | 480 | 500 | µA | For 20-/28-pin devices. |
| | | | | | | For 8-pin devices. |

Notes:
1. Typical conditions are defined as $V_{DD}$ = 3.3V and +30°C.
2. Standard temperature is defined as $T_A$ = 0°C to +70°C; these values not tested in production for worst case behavior, but are derived from product characterization and provided for design guidance only.
3. Extended temperature is defined as $T_A$ = –40°C to +105°C; these values not tested in production for worst case behavior, but are derived from product characterization and provided for design guidance only.
4. For this block to operate, the bandgap circuit is automatically turned on and must be added to the total supply current. This bandgap current is only added once, regardless of how many peripherals are using it.

## General Purpose I/O Port Output Timing

Figure 35 and Table 144 provide timing information for GPIO port pins.

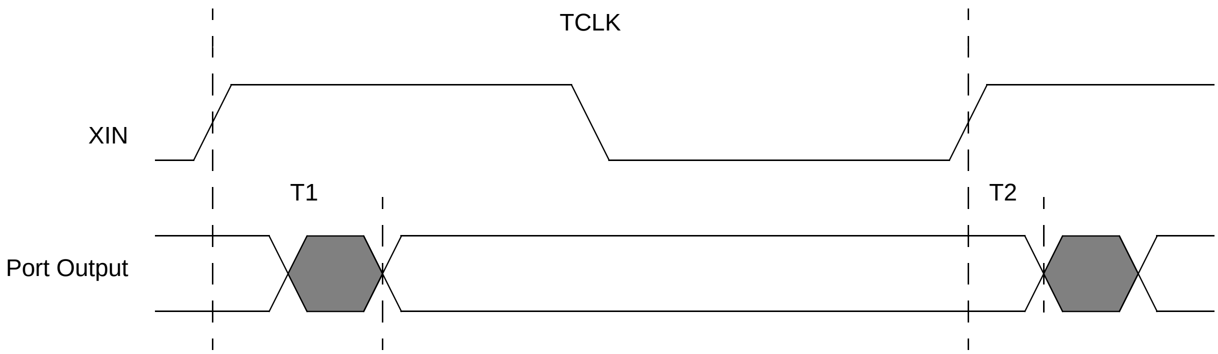


**Figure 35. GPIO Port Output Timing**

**Table 144. GPIO Port Output Timing**

| | | Delay (ns) | |
|---|---|---|---|
| **Parameter** | **Abbreviation** | **Minimum** | **Maximum** |
| **GPIO port pins** | | | |
| $T_1$ | $X_{IN}$ Rise to Port Output Valid Delay | – | 15 |
| $T_2$ | $X_{IN}$ Rise to Port Output Hold Time | 2 | – |