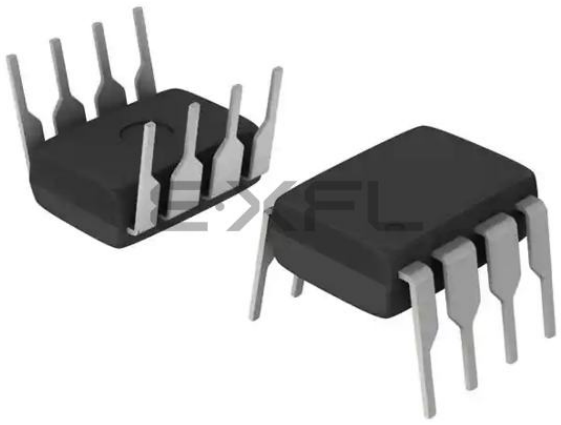

Welcome to E-XFL.COM

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	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K 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	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f082apb020eg

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
F85	Reserved	—	XX	
Oscillator Control				
F86	Oscillator Control	OSCCTL	A0	196
F87–F8F	Reserved	—	XX	
Comparator 0				
F90	Comparator 0 Control	CMP0	14	141
F91–FBF	Reserved	—	XX	
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	60
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	63
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	63
FC3	Interrupt Request 1	IRQ1	00	61
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	65
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	65
FC6	Interrupt Request 2	IRQ2	00	62
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	66
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	67
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	68
FCE	Shared Interrupt Select	IRQSS	00	68
FCF	Interrupt Control	IRQCTL	00	69
GPIO Port A				
FD0	Port A Address	PAADDR	00	44
FD1	Port A Control	PACTL	00	46
FD2	Port A Input Data	PAIN	XX	46
FD3	Port A Output Data	PAOUT	00	46
GPIO Port B				
FD4	Port B Address	PBADDR	00	44
FD5	Port B Control	PBCTL	00	46
FD6	Port B Input Data	PBIN	XX	46
FD7	Port B Output Data	PBOUT	00	46
GPIO Port C				
FD8	Port C Address	PCADDR	00	44

Notes:
1. XX = Undefined.
2. Refer to the eZ8 CPU Core User Manual (UM0128).

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 $\overline{\text{RESET}}$ alternate function during system resets, driving it Low holds the chip in a reset state until the pin is released.

Table 23. Port A–D Output Control Subregisters (PxOC)

Bit	7	6	5	4	3	2	1	0
Field	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
RESET	00H (Ports A-C); 01H (Port D)							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 03H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] POCx	**Port Output Control** These bits function independently of the alternate function bit and always disable the drains if set to 1. 0 = The source current is enabled for any output mode unless overridden by the alternate function (push-pull output). 1 = The source current for the associated pin is disabled (open-drain mode).

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

Port A–D High Drive Enable Subregisters

The Port A–D High Drive Enable Subregister, shown in Table 24, is accessed through the port A–D Control Register by writing `04H` to the Port A–D Address Register. Setting the bits in the Port A–D High Drive Enable subregisters to 1 configures the specified port pins for high current output drive operation. The Port A–D High Drive Enable subregister affects the pins directly and, as a result, alternate functions are also affected.

Table 24. Port A–D High Drive Enable Subregisters (PxHDE)

Bit	7	6	5	4	3	2	1	0
Field	PHDE7	PHDE6	PHDE5	PHDE4	PHDE3	PHDE2	PHDE1	PHDE0
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 04H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] PHDEx	**Port High Drive Enabled** 0 = The port pin is configured for standard output current drive. 1 = The port pin is configured for high output current drive.

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

Table 39. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	Reserved	Reserved	ADCENH
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FC1H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1ENH	**Timer 1 Interrupt Request Enable High Bit**
[5] T0ENH	**Timer 0 Interrupt Request Enable High Bit**
[4] U0RENH	**UART 0 Receive Interrupt Request Enable High Bit**
[3] U0TENH	**UART 0 Transmit Interrupt Request Enable High Bit**
[2:1]	**Reserved** These bits are reserved and must be programmed to 00.
[0] ADCENH	**ADC Interrupt Request Enable High Bit**

Table 40. IRQ0 Enable Low Bit Register (IRQ0ENL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENL	T0ENL	U0RENL	U0TENL	Reserved	Reserved	ADCENL
RESET	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R	R	R/W
Address	FC2H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1ENL	**Timer 1 Interrupt Request Enable Low Bit**
[5] T0ENL	**Timer 0 Interrupt Request Enable Low Bit**

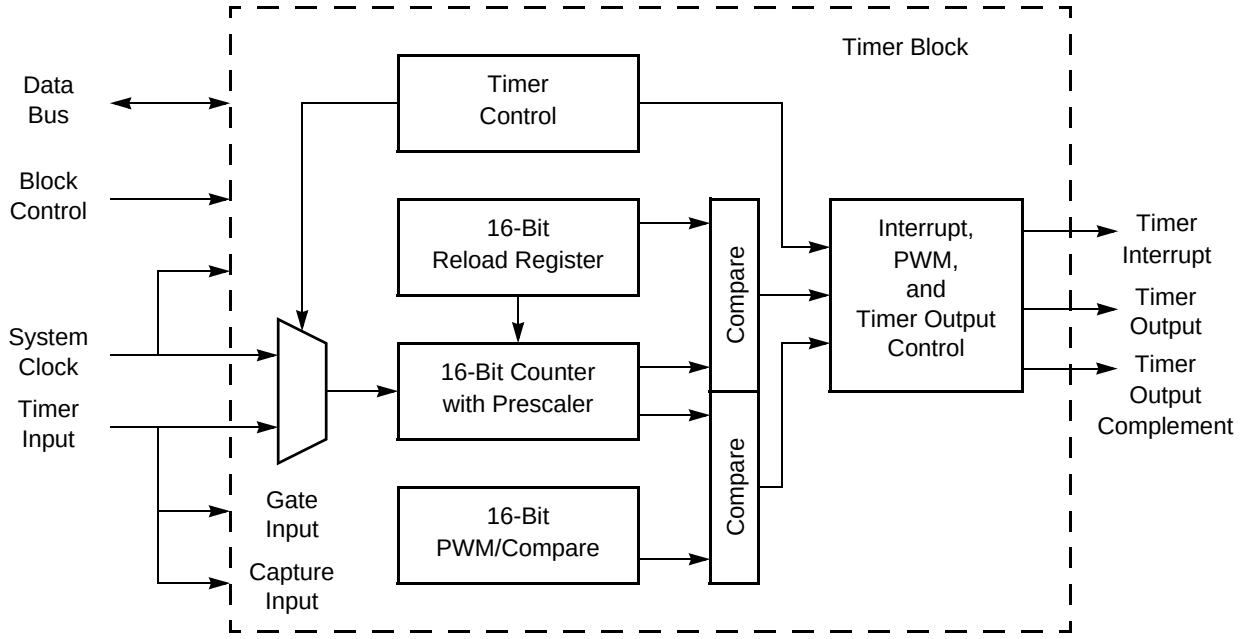

Figure 9. Timer Block Diagram

Operation

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

Timer Operating Modes

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

ONE-SHOT Mode

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

Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state for one system clock cycle (from Low to High or from High to Low) upon timer Reload. If

5. 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 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.
6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

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

CAPTURE RESTART Mode

In CAPTURE RESTART Mode, the current timer count value is recorded when the acceptable external Timer Input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the 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 because of 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 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. The INPCAP bit in TxCTL0 Register is cleared to indicate the timer interrupt is not caused by an input capture event.

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

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for CAPTURE RESTART Mode by writing the TMODE bits in the TxCTL1 Register and the TMODEHI bit in TxCTL0 Register
   - 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).

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTH, WDTL) registers, shown in Tables 60 through 62, form the 24-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes. The 24-bit reload value ranges across bits [23:0] to encompass the three bytes {WDTU[7:0], WDTH[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.

Table 60. Watchdog Timer Reload Upper Byte Register (WDTU)

Bit	7	6	5	4	3	2	1	0
Field	WDTU							
RESET	00H							
R/W	R/W*							
Address	FF1H							

Note: A read returns the current WDT count value; a write sets the appropriate reload value.

Bit	Description
[7:0] WDTU	**WDT Reload Upper Byte** Most-significant byte (MSB); bits[23:16] of the 24-bit WDT reload value.

Table 61. Watchdog Timer Reload High Byte Register (WDTH)

Bit	7	6	5	4	3	2	1	0
Field	WDTH							
RESET	04H							
R/W	R/W*							
Address	FF2H							

Note: A read returns the current WDT count value; a write sets the appropriate reload value.

Bit	Description
[7:0] WDTH	**WDT Reload High Byte** Middle byte; bits[15:8] of the 24-bit WDT reload value.

6. Read data from the UART Receive Data Register. If operating in MULTIPROCESSOR (9-bit) Mode, further actions may be required depending on the MULTIPROCESSOR Mode bits MPMD[1:0].
7. Return to Step 4 to receive additional data.

Receiving Data using the Interrupt-Driven Method

The UART Receiver interrupt indicates the availability of new data (and error conditions). Observe the following steps to configure the UART receiver for interrupt-driven operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the acceptable baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Receiver interrupt and set the acceptable priority.
5. Clear the UART Receiver interrupt in the applicable Interrupt Request Register.
6. Write to the UART Control 1 Register to enable Multiprocessor (9-bit) mode functions, if appropriate.
   - Set the Multiprocessor Mode Select (MPEN) to Enable MULTIPROCESSOR Mode.
   - Set the Multiprocessor Mode Bits, MPMD[1:0], to select the acceptable address matching scheme.
   - Configure the UART to interrupt on received data and errors or errors only (interrupt on errors only is unlikely to be useful for Z8 Encore! devices without a DMA block)
7. Write the device address to the Address Compare Register (automatic MULTIPROCESSOR Modes only).
8. Write to the UART Control 0 Register to:
   - Set the receive enable bit (REN) to enable the UART for data reception
   - Enable parity, if appropriate and if multiprocessor mode is not enabled and select either even or odd parity
9. Execute an EI instruction to enable interrupts.

In MULTIPROCESSOR (9-bit) Mode, the Parity (9th) bit location becomes the multiprocessor control bit. The UART Control 1 and Status 1 registers provide MULTIPROCESSOR (9-bit) Mode control and status information. If an automatic address matching scheme is enabled, the UART Address Compare Register holds the network address of the device.

MULTIPROCESSOR (9-bit) Mode Receive Interrupts

When MULTIPROCESSOR Mode is enabled, the UART only processes frames addressed to it. The determination of whether a frame of data is addressed to the UART can be made in hardware, software or some combination of the two, depending on the multiprocessor configuration bits. In general, the address compare feature reduces the load on the CPU, because it does not require access to the UART when it receives data directed to other devices on the multi-node network. The following three MULTIPROCESSOR Modes are available in hardware:

- Interrupt on all address bytes
- Interrupt on matched address bytes and correctly framed data bytes
- Interrupt only on correctly framed data bytes

These modes are selected with MPMD[1:0] in the UART Control 1 Register. For all multiprocessor modes, bit MPEN of the UART Control 1 Register must be set to 1.

The first scheme is enabled by writing 01b to MPMD[1:0]. In this mode, all incoming address bytes cause an interrupt, while data bytes never cause an interrupt. The interrupt service routine must manually check the address byte that caused triggered the interrupt. If it matches the UART address, the software clears MPMD[0]. Each new incoming byte interrupts the CPU. The software is responsible for determining the end of the frame. It checks for the end-of-frame by reading the MPRX bit of the UART Status 1 Register for each incoming byte. If MPRX=1, a new frame has begun. If the address of this new frame is different from the UART's address, MPMD[0] must be set to 1 causing the UART interrupts to go inactive until the next address byte. If the new frame's address matches the UART's, the data in the new frame is processed as well.

The second scheme requires the following: set MPMD[1:0] to 10B and write the UART's address into the UART Address Compare Register. This mode introduces additional hardware control, interrupting only on frames that match the UART's address. When an incoming address byte does not match the UART's address, it is ignored. All successive data bytes in this frame are also ignored. When a matching address byte occurs, an interrupt is issued and further interrupts now occur on each successive data byte. When the first data byte in the frame is read, the NEWFRM bit of the UART Status 1 Register is asserted. All successive data bytes have NEWFRM=0. When the next address byte occurs, the hardware compares it to the UART's address. If there is a match, the interrupts continues and the NEWFRM bit is set for the first byte of the new frame. If there is no match, the UART ignores all incoming bytes until the next address match.

The window remains open until the count again reaches 8 (that is, 24 baud clock periods since the previous pulse was detected), giving the Endec a sampling window of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the Endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the Endec clock counter is reset, resynchronizing the Endec to the incoming signal, allowing the Endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the Endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All infrared endec configuration and status information is set by the UART Control registers as defined in the Universal Asynchronous Receiver/Transmitter section on page 99.

! Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.

#3	#3	#3	#3

4. Round the result and discard the least significant two bytes (equivalent to dividing by 2^{16}).

#3	#3	#3	#3

–

0x00	0x00	0x80	0x00

=

#4 MSB	#4 LSB

5. Determine the sign of the gain correction factor using the sign bits from Step 2. If the offset-corrected ADC value *and* the gain correction word both have the same sign, then the factor is positive and remains unchanged. If they have differing signs, then the factor is negative and must be multiplied by –1.

#5 MSB	#5 LSB

6. Add the gain correction factor to the original offset corrected value.

#5 MSB	#5 LSB

+

#1 MSB	#1 LSB

=

#6 MSB	#6 LSB

7. Shift the result to the right, using the sign bit determined in Step 1, to allow for the detection of computational overflow.

S →	#6 MSB	#6 LSB

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.

Flash Memory

The products in the Z8 Encore! XP F082A Series feature a nonvolatile Flash memory of 8KB (8192), 4 KB (4096), 2 KB (2048 bytes), or 1KB (1024) with read/write/erase capability. The Flash Memory can be programmed and erased in-circuit by user code or through the On-Chip Debugger. The features include:

- User controlled read and write protect capability
- Sector-based write protection scheme
- Additional protection schemes against accidental program and erasure

Architecture

The Flash memory array is arranged in pages with 512 bytes per page. The 512-byte page is the minimum Flash block size that can be erased. Each page is divided into 8 rows of 64 bytes.

For program or data protection, the Flash memory is also divided into sectors. In the Z8 Encore! XP F082A Series, these sectors are either 1024 bytes (in the 8KB devices) or 512 bytes (all other memory sizes) in size. Page and sector sizes are not generally equal.

The first 2 bytes of Flash Program memory are used as Flash option bits. For more information about their operation, see the Flash Option Bits chapter on page 159.

Table 78 describes the Flash memory configuration for each device in the Z8 Encore! XP F082A Series. Figure 21 displays the Flash memory arrangement.

Table 78. Z8 Encore! XP F082A Series Flash Memory Configurations

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash Sector Size (Bytes)
Z8F08xA	8 (8192)	16	0000H–1FFFH	1024
Z8F04xA	4 (4096)	8	0000H–0FFFH	512
Z8F02xA	2 (2048)	4	0000H–07FFH	512
Z8F01xA	1 (1024)	2	0000H–03FFH	512

Trim Bit Data Register

The Trim Bid Data (TRMDR) Register contains the read or write data for access to the trim option bits (Table 87).

Table 87. Trim Bit Data Register (TRMDR)

Bit	7	6	5	4	3	2	1	0
Field	TRMDR: Trim Bit Data							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	FF7H							

Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses 0000H and 0001H are reserved for the user-programmable Flash option bits.

Flash Program Memory Address 0000H

Table 88. Flash Option Bits at Program Memory Address 0000H

Bit	7	6	5	4	3	2	1	0
Field	WDT_RES	WDT_AO	OSC_SEL[1:0]		VBO_AO	FRP	Reserved	FWP
RESET	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Program Memory 0000H							

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

Bit	Description
[7] WDT_RES	**Watchdog Timer Reset** 0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request. 1 = Watchdog Timer time-out causes a system reset. This setting is the default for unprogrammed (erased) Flash.
[6] WDT_AO	**Watchdog Timer Always On** 0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled. 1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

Table 105. Randomized Lot ID Locations (Continued)

Info Page Address	Memory Address	Usage
6A	FE6A	Randomized Lot ID Byte 13.
6B	FE6B	Randomized Lot ID Byte 12.
6D	FE6D	Randomized Lot ID Byte 11.
6E	FE6E	Randomized Lot ID Byte 10.
70	FE70	Randomized Lot ID Byte 9.
71	FE71	Randomized Lot ID Byte 8.
73	FE73	Randomized Lot ID Byte 7.
74	FE74	Randomized Lot ID Byte 6.
76	FE76	Randomized Lot ID Byte 5.
77	FE77	Randomized Lot ID Byte 4.
79	FE79	Randomized Lot ID Byte 3.
7A	FE7A	Randomized Lot ID Byte 2.
7C	FE7C	Randomized Lot ID Byte 1.
7D	FE7D	Randomized Lot ID Byte 0 (least significant).

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.

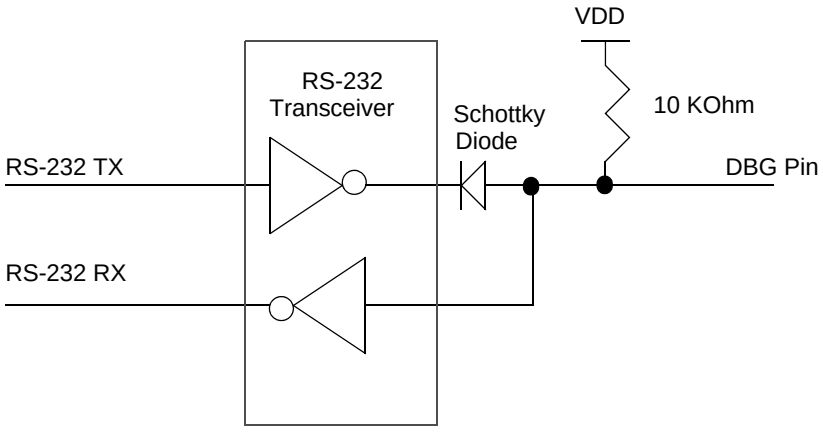

Figure 24. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #1 of 2

Oscillator Operation with an External RC Network

Figure 28 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

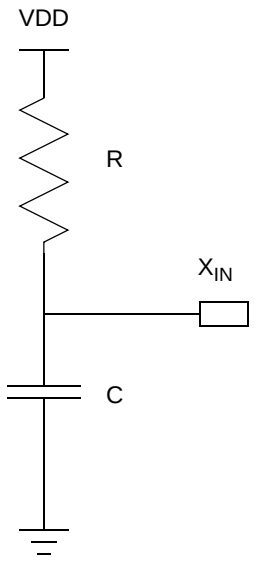

Figure 28. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45kΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40 kΩ. The typical oscillator frequency can be estimated from the values of the resistor (*R* in kΩ) and capacitor (*C* in pF) elements using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1\times10^{6}}{(0.4 \times R \times C) + (4 \times C)}$$

Figure 29 displays the typical (3.3V and 25°C) oscillator frequency as a function of the capacitor (C, in pF) employed in the RC network assuming a 45KΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator X_{IN} pin and the printed circuit board must be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20pF are recommended.

AC Characteristics

The section provides information about the AC characteristics and timing. All AC timing information assumes a standard load of 50pF on all outputs.

Table 133. AC Characteristics

		V_{DD} = 2.7V to 3.6V T_A = –40°C to +105°C (unless otherwise stated)			
Symbol	**Parameter**	**Minimum**	**Maximum**	**Units**	**Conditions**
F_{SYSCLK}	System Clock Frequency	–	20.0	MHz	Read-only from Flash memory
		0.032768	20.0	MHz	Program or erasure of the Flash memory
F_{XTAL}	Crystal Oscillator Frequency	–	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver
T_{XIN}	System Clock Period	50	–	ns	$T_{CLK} = 1/F_{sysclk}$
T_{XINH}	System Clock High Time	20	30	ns	T_{CLK} = 50 ns
T_{XINL}	System Clock Low Time	20	30	ns	T_{CLK} = 50 ns
T_{XINR}	System Clock Rise Time	–	3	ns	T_{CLK} = 50 ns
T_{XINF}	System Clock Fall Time	–	3	ns	T_{CLK} = 50 ns

Table 134. Internal Precision Oscillator Electrical Characteristics

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

Table 139. Analog-to-Digital Converter Electrical Characteristics and Timing (Continued)

		V_{DD} = 3.0 V to 3.6 V T_A = 0°C to +70°C (unless otherwise stated)				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
	Continuous Conversion Time	–	256	–	System clock cycles	All measurements but temperature sensor
			512			Temperature sensor measurement
	Signal Input Bandwidth	–	10		kHz	As defined by -3 dB point
R_S	Analog Source Impedance[4]	–	–	10	kΩ	In unbuffered mode
				500	kΩ	In buffered modes
Zin	Input Impedance	–	150		kΩ	In unbuffered mode at 20MHz[5]
		10	–		MΩ	In buffered modes
Vin	Input Voltage Range	0		V_{DD}	V	Unbuffered Mode
		0.3		V_{DD}–1.1	V	Buffered Modes These values define the range over which the ADC performs within spec; exceeding these values does not cause damage or instability; see DC Characteristics for absolute pin voltage limits.

Notes:
1. Analog source impedance affects the ADC offset voltage (because of pin leakage) and input settling time.
2. Devices are factory calibrated at V_{DD} = 3.3V and T_A = +30°C, so the ADC is maximally accurate under these conditions.
3. LSBs are defined assuming 10-bit resolution.
4. This is the maximum recommended resistance seen by the ADC input pin.
5. The input impedance is inversely proportional to the system clock frequency.