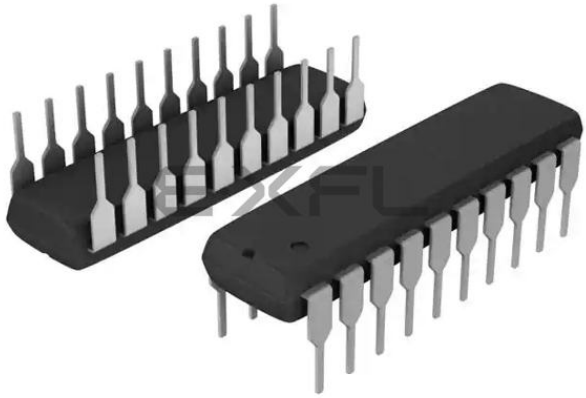


Welcome to E-XFL.COM

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	I²C, IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	11
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	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0811ph020sc

The eZ8 CPU features include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required Program Memory.
- Software stack allows much greater depth in subroutine calls and interrupts than hardware stacks.
- Compatible with existing Z8® code.
- Expanded internal Register File allows access of up to 4 KB.
- New instructions improve execution efficiency for code developed using higher-level programming languages, including C.
- Pipelined instruction fetch and execution.
- New instructions for improved performance including BIT, BSWAP, BTJ, CPC, LDC, LDCI, LEA, MULT, and SRL.
- New instructions support 12-bit linear addressing of the Register File.
- Up to 10 MIPS operation.
- C-Compiler friendly.
- 2 to 9 clock cycles per instruction.

For more information regarding the eZ8 CPU, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com.

General Purpose Input/Output

Z8 Encore! XP® F0822 Series features 11 to 19 port pins (Ports A–C) for General Purpose Input/Output (GPIO). The number of GPIO pins available is a function of package. Each pin is individually programmable. Ports A and C supports 5 V-tolerant inputs.

Flash Controller

The Flash Controller programs and erases the Flash memory.

10-Bit Analog-to-Digital Converter

The optional Analog-to-Digital Converter (ADC) converts an analog input signal to a 10-bit binary number. The ADC accepts inputs from 2 to 5 different analog input sources.

UART

The Universal Asynchronous Receiver/Transmitter (UART) is full-duplex and capable of handling asynchronous data transfers. The UART supports 8-bit and 9-bit data modes and selectable parity.

Interrupt Vectors and Priority

The interrupt controller supports three levels of interrupt priority. Level 3 is the highest priority, Level 2 is the second highest priority, and Level 1 is the lowest priority. If all the interrupts were enabled with identical interrupt priority (all as Level 2 interrupts), then interrupt priority would be assigned from highest to lowest as specified in Table 24. Level 3 interrupts always have higher priority than Level 2 interrupts which in turn always have higher priority than Level 1 interrupts. Within each interrupt priority level (Level 1, Level 2, or Level 3), priority is assigned as specified in Table 24. Reset, WDT interrupt (if enabled), and Illegal Instruction Trap always have highest priority.

Interrupt Assertion

Interrupt sources assert their interrupt requests for only a single system clock period (single pulse). When the interrupt request is acknowledged by the eZ8 CPU, the corresponding bit in the Interrupt Request Register is cleared until the next interrupt occurs. Writing a 0 to the corresponding bit in the Interrupt Request Register likewise clears the interrupt request.

Caution: *The following style of coding to clear bits in the Interrupt Request Registers is not recommended. All incoming interrupts received between execution of the first LDX command and the last LDX command is lost.*

Poor coding style resulting in lost interrupt requests:

```
LDX r0, IRQ0
AND r0, MASK
LDX IRQ0, r0
```

Note: *To avoid missing interrupts, the following style of coding to clear bits in the Interrupt Request 0 register is recommended:*

Good coding style that avoids lost interrupt requests:

```
ANDX IRQ0, MASK
```

Software Interrupt Assertion

Program code generates interrupts directly. Writing 1 to the desired bit in the Interrupt Request Register triggers an interrupt (assuming that interrupt is enabled). When the interrupt request is acknowledged by the eZ8 CPU, the bit in the Interrupt Request Register is automatically cleared to 0.

Caution: *The following style of coding to generate software interrupts by setting bits in the Interrupt Request Registers is not recommended. All incoming interrupts received between execution of the first LDX command and the last LDX command is lost.*

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register (Table 37) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port input pin. The minimum pulse width must be greater than 1 system clock to guarantee capture of the edge triggered interrupt. Edge detection for pulses less than 1 system clock are not guaranteed.

Table 37. Interrupt Edge Select Register (IRQES)

BITS	7	6	5	4	3	2	1	0
FIELD	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
RESET	0							
R/W	R/W							
ADDR	FCDH							

IES*x*—Interrupt Edge Select *x*
0 = An interrupt request is generated on the falling edge of the PA*x* input.
1 = An interrupt request is generated on the rising edge of the PA*x* input.
Where *x* indicates the specific GPIO Port pin number (0 through 7).

Interrupt Control Register

The Interrupt Control (IRQCTL) register (Table 38) contains the master enable bit for all interrupts.

Table 38. Interrupt Control Register (IRQCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	IRQE	Reserved						
RESET	0							
R/W	R/W	R						
ADDR	FCFH							

IRQE—Interrupt Request Enable
This bit is set to 1 by execution of an EI (Enable Interrupts) or IRET (Interrupt Return) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, Reset or by a direct register write of a 0 to this bit.
0 = Interrupts are disabled.
1 = Interrupts are enabled.

Reserved—Must be 0

Timer 0–1 Control 1 Registers

The Timer 0–1 Control (TxCTL) registers enable/disable the timers, set the prescaler value, and determine the timer operating mode.

Table 46. Timer 0–1 Control Register (TxCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	TPOL	PRES			TMODE		
RESET	0							
R/W	R/W							
ADDR	F07H, F0FH							

TEN—Timer Enable
0 = Timer is disabled.
1 = Timer enabled to count.

TPOL—Timer Input/Output Polarity
Operation of this bit is a function of the current operating mode of the timer.

ONE-SHOT Mode
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.

CONTINUOUS Mode
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.

COUNTER Mode
If the timer is enabled the Timer Output signal is complemented after timer reload.
0 = Count occurs on the rising edge of the Timer Input signal.
1 = Count occurs on the falling edge of the Timer Input signal.

PWM Mode
0 = Timer Output is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon Reload.
1 = Timer Output is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon Reload.

CAPTURE Mode
0 = Count is captured on the rising edge of the Timer Input signal.
1 = Count is captured on the falling edge of the Timer Input signal.

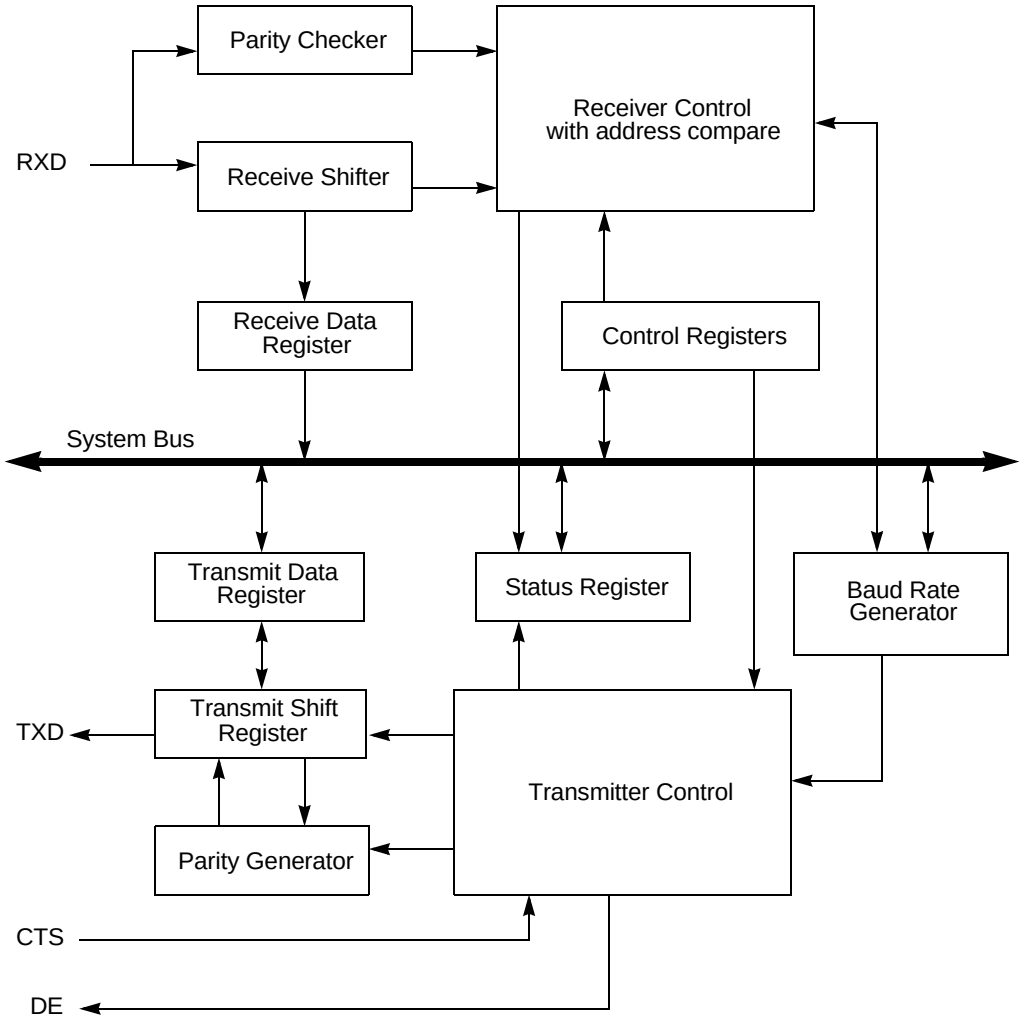


Figure 11. UART Block Diagram

Operation

Data Format

The UART always transmits and receives data in an 8-bit data format, least-significant bit first. An even or odd parity bit is optionally added to the data stream. Each character begins with an active Low START bit and ends with either 1 or 2 active High STOP bits. Figure12 on page 91 and Figure13 on page 91 display the asynchronous data format used by the UART without parity and with parity, respectively.

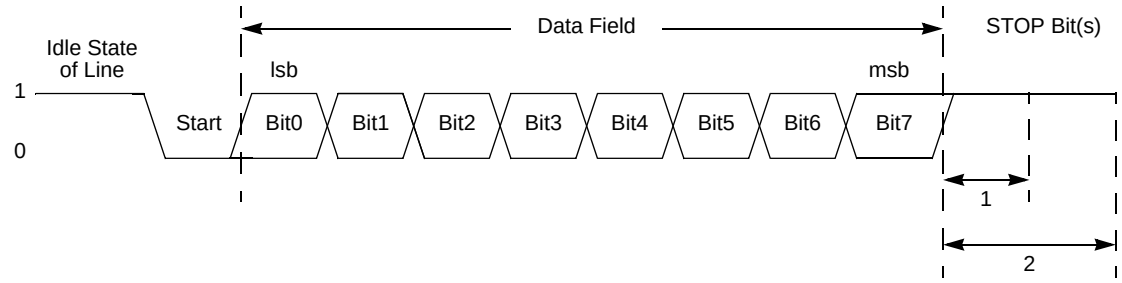


Figure 12. UART Asynchronous Data Format without Parity

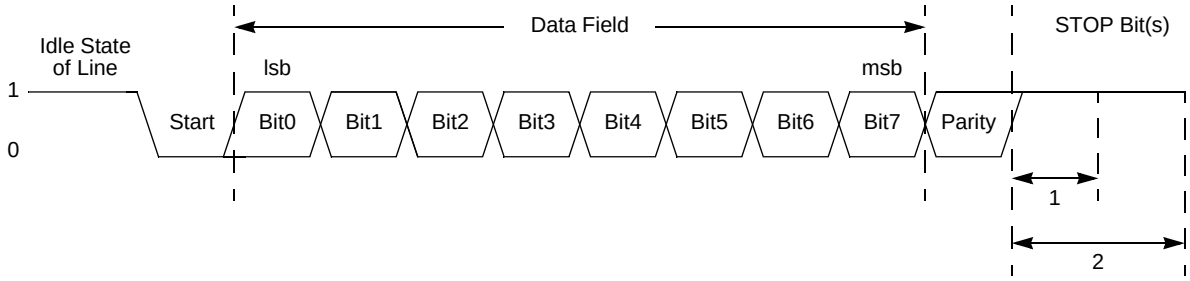


Figure 13. UART Asynchronous Data Format with Parity

Transmitting Data using Polled Method

Follow the steps below to transmit data using polled method of operation:

1. Write to the UART Baud Rate High Byte and Low Byte registers to set the required baud rate.
2. Enable the UART pin functions by configuring the associated GPIO Port pins for alternate function operation.
3. If MULTIPROCESSOR mode is required, write to the UART Control 1 Register to enable multiprocessor (9-bit) mode functions.
   – Set the Multiprocessor Mode Select (MPEN) to enable MULTIPROCESSOR mode.
4. Write to the UART Control 0 Register to:
   – Set the transmit enable bit (TEN) to enable the UART for data transmission
   – If parity is required, and MULTIPROCESSOR mode is not enabled, set the parity enable bit (PEN) and select either even or odd parity (PSEL).
   – Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.

Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte is received and is available in the UART Receive Data Register. This interrupt can be disabled independent of the other receiver interrupt sources. The received data interrupt occurs once the receive character is received and placed in the Receive Data Register. Software must respond to this received data available condition before the next character is completely received to avoid an overrun error. In MULTIPROCESSOR mode (MPEN = 1), the receive data interrupts are dependent on the multiprocessor configuration and the most recent address byte
- A break is received
- An overrun is detected
- A data framing error is detected

UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data Register. The break detect and overrun status bits are not displayed until the valid data is read.

After the valid data has been read, the UART Status 0 Register is updated to indicate the overrun condition (and Break Detect, if applicable). The RDA bit is set to 1 to indicate that the Receive Data Register contains a data byte. However, because the overrun error occurred, this byte cannot contain valid data and should be ignored. The BRKD bit indicates if the overrun was caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data Register must be read again to clear the error bits is the UART Status 0 Register. Updates to the Receive Data Register occur only when the next data word is received.

UART Data and Error Handling Procedure

Figure16 on page 99 displays the recommended procedure for UART receiver ISRs.

Baud Rate Generator Interrupts

If the BRG interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the BRG to function as an additional counter if the UART functionality is not employed.

SPI Status Register

The SPI Status Register indicates the current state of the SPI. All bits revert to their reset state if the SPIEN bit in the SPICTL Register equals 0.

Table 65. SPI Status Register (SPISTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	IRQ	OVR	COL	ABT	Reserved		TXST	SLAS
RESET	0							1
R/W	R/W*				R			
ADDR	F62H							
R/W* = Read access. Write a 1 to clear the bit to 0.								

IRQ—Interrupt Request
If SPIEN = 1, this bit is set if the STR bit in the SPICTL Register is set, or upon completion of an SPI Master or Slave transaction. This bit does not set if SPIEN = 0 and the SPI Baud Rate Generator is used as a timer to generate the SPI interrupt.
0 = No SPI interrupt request pending.
1 = SPI interrupt request is pending.

OVR—Overrun
0 = An overrun error has not occurred.
1 = An overrun error has been detected.

COL—Collision
0 = A multi-master collision (mode fault) has not occurred.
1 = A multi-master collision (mode fault) has been detected.

ABT—SLAVE mode transaction abort
This bit is set if the SPI is configured in SLAVE mode, a transaction is occurring and $\overline{SS}$ deasserts before all bits of a character have been transferred as defined by the NUMBITS field of the SPIMODE Register. The IRQ bit also sets, indicating the transaction has completed.
0 = A SLAVE mode transaction abort has not occurred.
1 = A SLAVE mode transaction abort has been detected.

Reserved—Must be 0

TXST—Transmit Status
0 = No data transmission currently in progress.
1 = Data transmission currently in progress.

SLAS—Slave Select
If SPI enabled as a Slave
0 = $\overline{SS}$ input pin is asserted (Low)
1 = $\overline{SS}$ input is not asserted (High).
If SPI enabled as a Master, this bit is not applicable.

SPI Mode Register

The SPI Mode Register configures the character bit width and the direction and value of the $\overline{SS}$ pin.

Table 66. SPI Mode Register (SPIMODE)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved		DIAG	NUMBITS[2:0]			SSIO	SSV
RESET	0							
R/W	R		R/W					
ADDR	F63H							

Reserved—Must be 0

DIAG–Diagnostic Mode Control bit
This bit is for SPI diagnostics. Setting this bit allows the BRG value to be read using the SPIBRH and SPIBRL Register locations.
0 = Reading SPIBRH, SPIBRL returns the value in the SPIBRH and SPIBRL Registers
1 = Reading SPIBRH returns bits [15:8] of the SPI Baud Rate Generator; and reading SPIBRL returns bits [7:0] of the SPI Baud Rate Counter. The Baud Rate Counter High and Low byte values are not buffered.

Caution: *Take precautions if you are reading the values while BRG is counting.*

NUMBITS[2:0]—Number of Data Bits Per Character to Transfer
This field contains the number of bits to shift for each character transfer. See the SPI Data Register description for information on valid bit positions when the character length is less than 8-bits.
000 = 8 bits
001 = 1 bit
010 = 2 bits
011 = 3 bits
100 = 4 bits
101 = 5 bits
110 = 6 bits
111 = 7 bits

SSIO—Slave Select I/O
0 = $\overline{SS}$ pin configured as an input.
1 = $\overline{SS}$ pin configured as an output (MASTER mode only).

SSV—Slave Select Value
If SSIO = 1 and SPI configured as a Master:
0 = $\overline{SS}$ pin driven Low (0).

Operation

The I²C Controller operates in MASTER mode to transmit and receive data. Only a single master is supported. Arbitration between two masters must be accomplished in software. I²C supports the following operations:

- Master transmits to a 7-bit Slave
- Master transmits to a 10-bit Slave
- Master receives from a 7-bit Slave
- Master receives from a 10-bit Slave

SDA and SCL Signals

I²C sends all addresses, data and acknowledge signals over the SDA line, most-significant bit first. SCL is the common clock for the I²C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I²C) is responsible for driving the SCL clock signal, although the clock signal becomes skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a high level. When the slave releases the clock, the I²C Controller continues the transaction. All data is transferred in bytes and there is no limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the high period of SCL.

I²C Interrupts

The I²C Controller contains four sources of interrupts—Transmit, Receive, Not Acknowledge, and Baud Rate Generator. These four interrupt sources are combined into a single interrupt request signal to the interrupt controller. The Transmit Interrupt is enabled by the IEN and TXI bits of the control register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the control register. BRG interrupt is enabled by the BIRQ and IEN bits of the control register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I²C Controller and neither the START or STOP bit is set. The Not Acknowledge event sets the NCKI bit of the I²C Status Register and can only be cleared by setting the START or STOP bit in the I²C Control Register. When this interrupt occurs, the I²C Controller waits until either the STOP or START bit is set before performing any action. In an ISR, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

frequency range for the device. The Flash Frequency High and Low Byte Registers must be loaded with the correct value to insure proper program and erase times.

Table 87. Flash Frequency High Byte Register (FFREQH)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQH							
RESET	0							
R/W	R/W							
ADDR	FFAH							

Table 88. Flash Frequency Low Byte Register (FFREQL)

BITS	7	6	5	4	3	2	1	0
FIELD	FFREQL							
RESET	0							
R/W	R/W							
ADDR	FFBH							

FFREQH and FFREQL—Flash Frequency High and Low Bytes

These 2 bytes, {FFREQH[7:0], FFREQL[7:0]}, contain the 16-bit Flash Frequency value.

Flash Memory Address 0000H

Table 89. Option Bits at Flash Memory Address 0000H for 8K Series Flash Devices

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RES	WDT_AO	OSC_SEL[1:0]		VBO_AO	RP	Reserved	FWP
RESET	U							
R/W	R/W							
ADDR	Program Memory 0000H							

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

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 Reset. This setting is the default for unprogrammed (erased) Flash.

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.

OSC_SEL[1:0]—OSCILLATOR Mode Selection

00 = On-chip oscillator configured for use with external RC networks (<4 MHz).

01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).

10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 10.0 MHz).

11 = Maximum power for use with high frequency crystals (8.0 MHz to 20.0 MHz). This setting is the default for unprogrammed (erased) Flash.

VBO_AO—Voltage Brownout Protection Always On

0 = Voltage Brownout Protection is disabled in STOP mode to reduce total power consumption.

1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

RP—Read Protect

0 = User program code is inaccessible. Limited control features are available through the OCD.

1 = User program code is accessible. All OCD commands are enabled. This setting is the default for unprogrammed (erased) Flash.

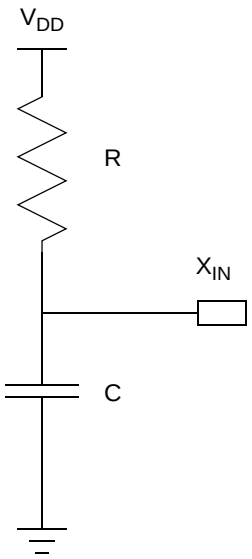


Figure 35. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45 kΩ 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 below equation:

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

Figure 36 on page 170 displays the typical (3.3 V and 25 ^{0}C) oscillator frequency as a function of the capacitor (*C* in pF) employed in the RC network assuming a 45 kΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator XIN pin and the printed circuit board should 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 parasites, external capacitance values in excess of 20 pF are recommended.

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 can interface the Z8 Encore! XP F0822 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 38 and Figure 39.

Caution: *For operation of the OCD, 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 must always be connected to V_{DD} through an external pull-up resistor to insure proper operation.*

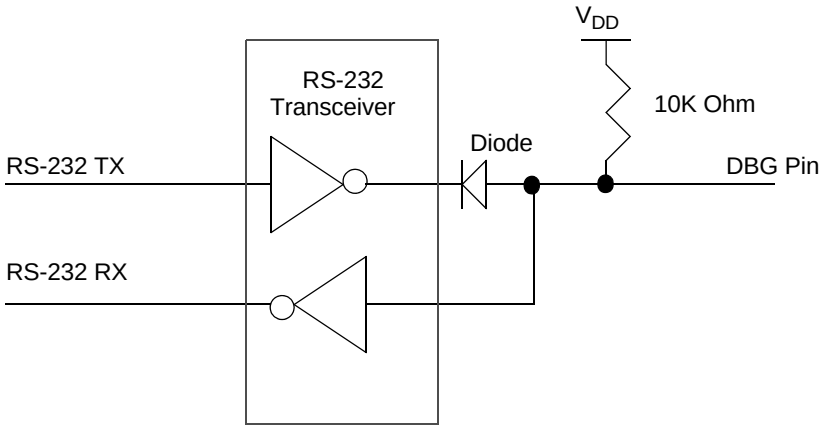


Figure 38. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (1)

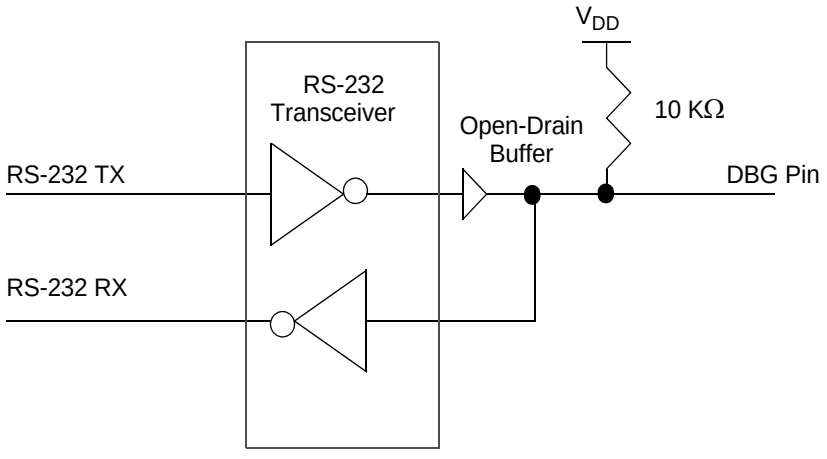


Figure 39. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (2)

automatically set to 1. If this bit is set, the OCDCNTR register does not count when the CPU is running.
0 = OCDCNTR is setup as counter
1 = OCDCNTR generates hardware break when PC == OCDCNTR

BRKZRO—Break when OCDCNTR == 0000H
If this bit is set, then the OCD automatically sets the DBGMODE bit when the OCDCNTR register counts down to 0000H. If this bit is set, the OCDCNTR register is not reset when the part leaves DEBUG Mode.
0 = OCD does not generate BRK when OCDCNTR decrements to 0000H
1 = OCD sets DBGMODE to 1 when OCDCNTR decrements to 0000H

Reserved
These bits are reserved and must be 0.

RST—Reset
Setting this bit to 1 resets the Z8 Encore! XP® F0822 Series device. The device goes through a normal POR sequence with the exception that the OCD is not reset. This bit is automatically cleared to 0 when the reset finishes.
0 = No effect.
1 = Reset the Z8 Encore! XP F0822 Series device.

OCD Status Register

The OCD Status register reports status information about the current state of the debugger and the system.

Table 95. OCD Status Register (OCDSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	IDLE	HALT	RPEN	Reserved				
RESET	0							
R/W	R							

IDLE—CPU Idling
This bit is set if the part is in DEBUG mode (DBGMODE is 1), or if a BRK instruction occurred since the last time OCDCTL was written. This can be used to determine if the CPU is running or if it is idling.
0 = The eZ8 CPU is running.
1 = The eZ8 CPU is either stopped or looping on a BRK instruction.

HALT—HALT Mode
0 = The device is not in HALT mode.
1 = The device is in HALT mode.

Table 97. DC Characteristics (Continued)

Symbol	Parameter	T_A = -40 °C to 105 °C			Units	Conditions
		Minimum	Typical	Maximum		
V_{RAM}	RAM Data Retention	0.7	–	–	V	
I_{IL}	Input Leakage Current	-5	–	+5	µA	V_{DD} = 3.6 V; V_{IN} = VDD or VSS[1]
I_{TL}	Tri-State Leakage Current	-5	–	+5	µA	V_{DD} = 3.6 V
C_{PAD}	GPIO Port Pad Capacitance	–	8.0[2]	–	pF	
C_{XIN}	XIN Pad Capacitance	–	8.0[2]	–	pF	
C_{XOUT}	XOUT Pad Capacitance	–	9.5[2]	–	pF	
I_{PU1}	Weak Pull-up Current	9	20	50	µA	VDD = 2.7–3.6 V. T_A = 0 °C to +70 °C
I_{PU2}	Weak Pull-up Current	7	20	75	µA	VDD = 2.7–3.6 V. T_A = -40 °C to +105 °C

[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 41 on page 189 displays the typical active mode current consumption while operating at 25 ºC, 3.3 V, versus the system clock frequency. All GPIO pins are configured as outputs and driven High.

Figure 47 displays the input frequency response of the ADC.

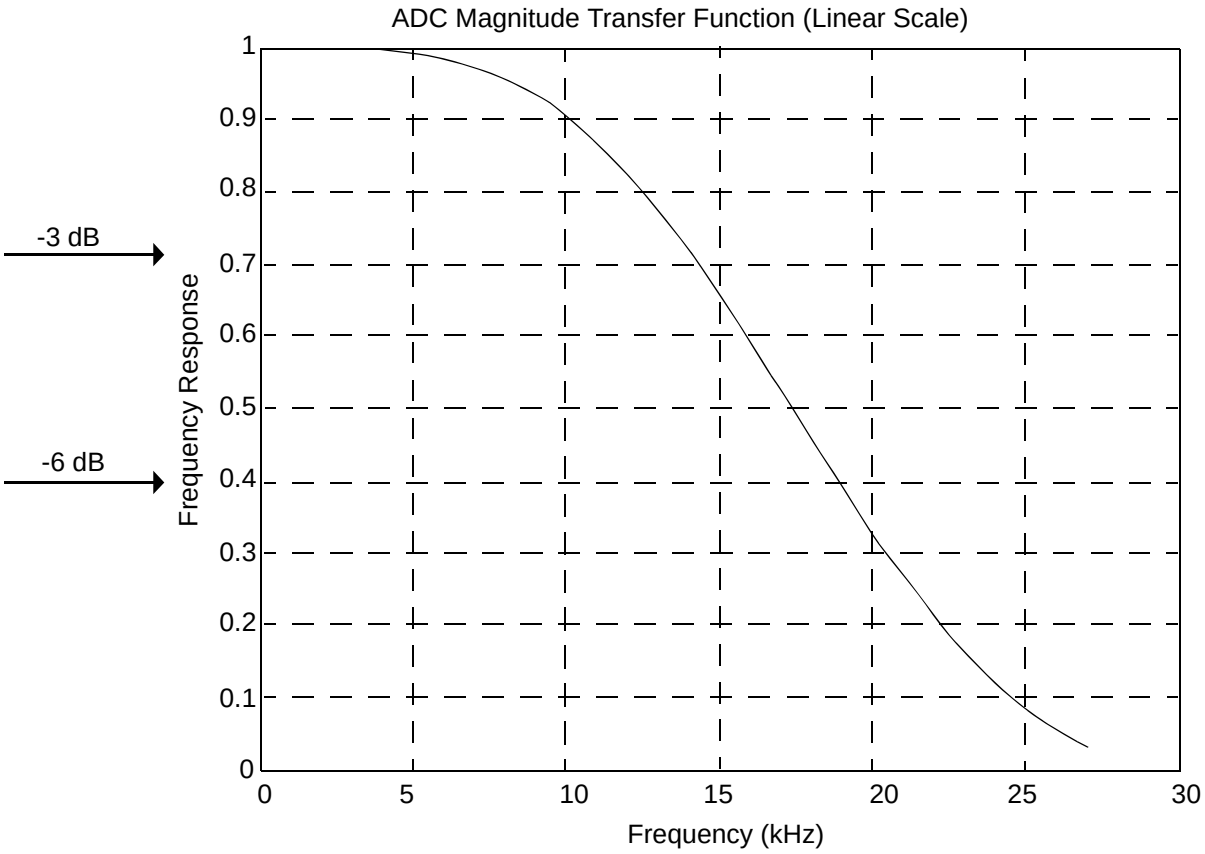


Figure 47. Analog-to-Digital Converter Frequency Response

SPI MASTER Mode Timing

Figure 51 and Table 108 provide timing information for SPI MASTER mode pins. Timing is shown with SCK rising edge used to source MOSI output data, SCK falling edge used to sample MISO input data. Timing on the SS output pin(s) is controlled by software.

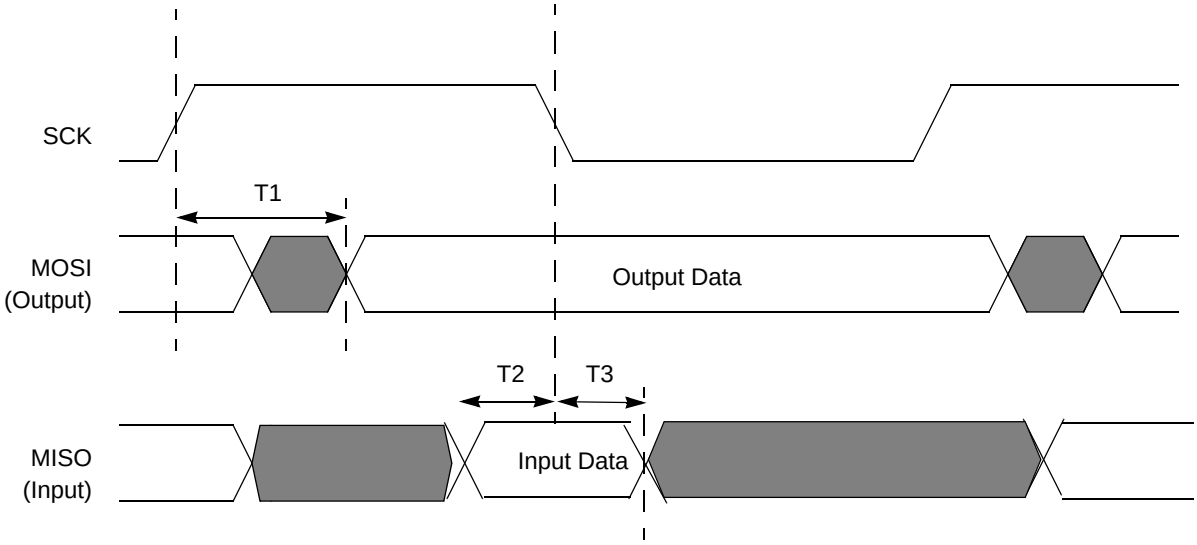


Figure 51. SPI MASTER Mode Timing

Table 108. SPI MASTER Mode Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
SPI MASTER			
T_1	SCK Rise to MOSI output Valid Delay	-5	+5
T_2	MISO input to SCK (receive edge) Setup Time	20	
T_3	MISO input to SCK (receive edge) Hold Time	0	

Table 126. eZ8 CPU Instruction Summary (Continued)

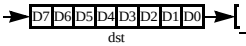
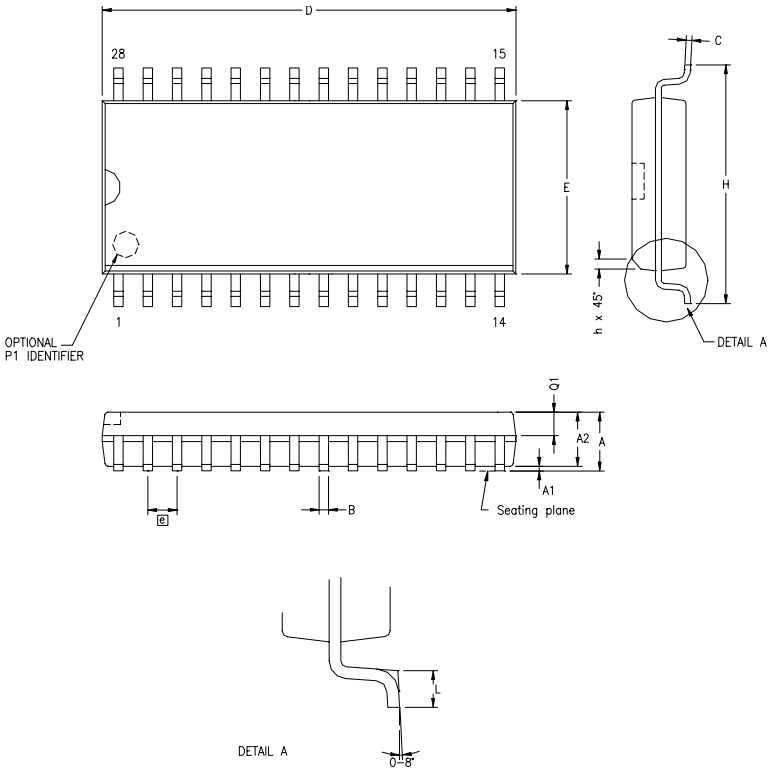
Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
SCF	C ← 1			DF	1	-	-	-	-	-	1	2
SRA dst		R		D0	*	*	*	0	-	-	2	2
		IR		D1							2	3
SRL dst		R		1F C0	*	*	0	*	-	-	3	2
		IR		1F C1							3	3
SRP src	RP ← src		IM	01	-	-	-	-	-	-	2	2
STOP	STOP Mode			6F	-	-	-	-	-	-	1	2
SUB dst, src	dst ← dst – src	r	r	22	*	*	*	*	1	*	2	3
		r	Ir	23							2	4
		R	R	24							3	3
		R	IR	25							3	4
		R	IM	26							3	3
		IR	IM	27							3	4
SUBX dst, src	dst ← dst – src	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29							4	3
SWAP dst	dst[7:4] ↔ dst[3:0]	R		F0	X	*	*	X	-	-	2	2
		IR		F1							2	3
TCM dst, src	(NOT dst) AND src	r	r	62	-	*	*	0	-	-	2	3
		r	Ir	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	-	*	*	0	-	-	4	3
		ER	IM	69							4	3

Figure 62 displays the 28-pin SOIC package available for Z8 Encore! XP F0822 Series devices.



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	2.40	2.64	.094	.104
A1	0.10	0.30	.004	.012
A2	2.24	2.44	.088	.096
B	0.36	0.46	.014	.018
C	0.23	0.30	.009	.012
D	17.78	18.00	.700	.710
E	7.40	7.60	.291	.299
e	1.27 BSC		.050 BSC	
H	10.00	10.65	.394	.419
h	0.30	0.71	.012	.028
L	0.61	1.00	.024	.039
Q1	0.97	1.09	.038	.043

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

Figure 62. 28-Pin Small Outline Integrated Circuit Package (SOIC)