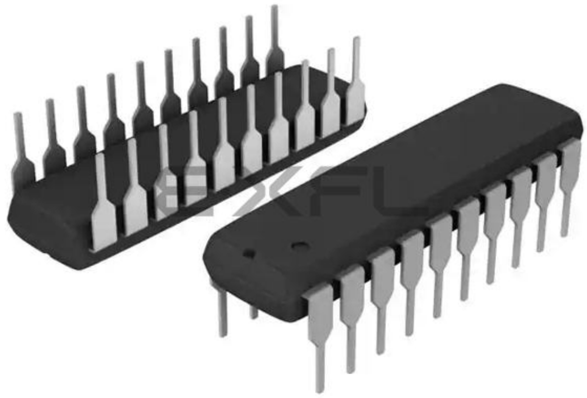
Zilog - Z8F0421PH020EC Datasheet



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	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0421ph020ec

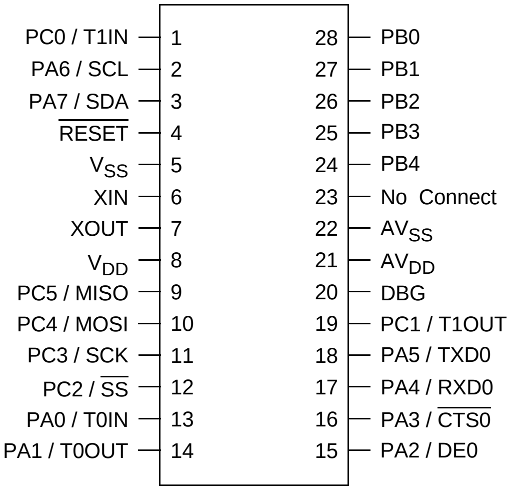


Figure 5. Z8F0812 and Z8F0412 in 28-Pin SOIC and PDIP Packages

Signal Descriptions

Table 3 describes Z8 Encore! XP® F0822 Series signals. See Pin Configurations on page 7 to determine the signals available for the specific package styles

.

Table 3. Signal Descriptions

Signal Mnemonic	I/O	Description
General-Purpose I/O Ports A-H		
PA[7:0]	I/O	**Port C**—These pins are used for general-purpose I/O and supports 5 V-tolerant inputs.
PB[4:0]	I/O	**Port B**—These pins are used for general-purpose I/O.
PC[5:0]	I/O	**Port C**—These pins are used for general-purpose I/O and support 5 V-tolerant inputs.
I^{2}C Controller		
SCL	I/O	**Serial Clock**—This open-drain pin clocks data transfers in accordance with the I^{2}C standard protocol. This pin is multiplexed with a GPIO pin. When the GPIO pin is configured for alternate function to enable the SCL function, this pin is open-drain.
SDA	I/O	**Serial Data**—This open-drain pin transfers data between the I^{2}C and a slave. This pin is multiplexed with a GPIO pin. When the GPIO pin is configured for alternate function to enable the SDA function, this pin is open-drain.

Address Space

The eZ8 CPU accesses three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, Peripheral, and GPIO Port Control Registers.
- The Program Memory contains addresses for all memory locations having executable code and/or data.
- The Data Memory contains addresses for all memory locations that hold data only.

These three address spaces are covered briefly in the following sections. For more information on the eZ8 CPU and its address space, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com.

Register File

The Register File address space in the Z8 Encore! XP® is 4 KB (4096 bytes). It is composed of two sections—Control Registers and General-Purpose Registers. When instructions are executed, registers are read from when defined as sources and written to when defined as destinations. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 1 KB Register File address space is reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256-byte Control Register section is reserved (unavailable). Reading from the reserved Register File addresses returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. Z8 Encore! XP F0822 Series contains 1 KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

Program Memory

The eZ8 CPU supports 64 KB of Program Memory address space. Z8 Encore! XP® F0822 Series contain 4 KB to 8 KB on-chip Flash in the Program Memory address space, depending on the device. Reading from Program Memory addresses outside the available Flash addresses returns FFH. Writing to unimplemented Program Memory addresses produces no effect Table 5 describes the Program Memory Maps for Z8 Encore! XP F0822 Series devices.

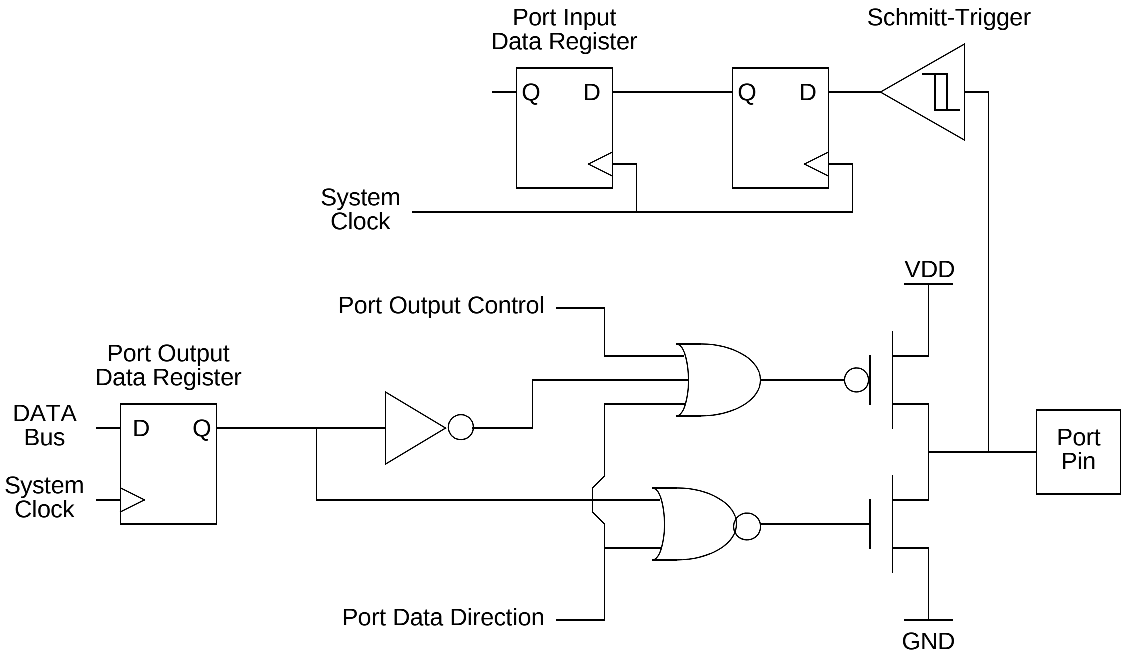


Figure 8. GPIO Port Pin Block Diagram

Table 12. Port Alternate Function Mapping

Port	Pin	Mnemonic	Alternate Function Description
Port A	PA0	T0IN	Timer 0 Input
	PA1	T0OUT	Timer 0 Output
	PA2	DE	UART 0 Driver Enable
	PA3	$\overline{\text{CTS0}}$	UART 0 Clear to Send
	PA4	RXD0 / IRRX0	UART 0 / IrDA 0 Receive Data
	PA5	TXD0 / IRTX0	UART 0 / IrDA 0 Transmit Data
	PA6	SCL	I^{2}C Clock (automatically open-drain)
	PA7	SDA	I^{2}C Data (automatically open-drain)
Port B	PB0	ANA0	ADC Analog Input 0
	PB1	ANA1	ADC Analog Input 1
	PB2	ANA2	ADC Analog Input 2
	PB3	ANA3	ADC Analog Input 3
	PB4	ANA4	ADC Analog Input 4

Port A–C Output Data Register

The Port A–C Output Data Register (Table 23) controls the output data to the pins.

Table 23. Port A–C Output Data Register (P*x*OUT)

BITS	7	6	5	4	3	2	1	0
FIELD	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0							
R/W	R/W							
ADDR	FD3H, FD7H, FDBH							

POUT[7:0]—Port Output Data

These bits contain the data to be driven to the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation.
0 = Drive a logical 0 (Low).
1 = Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding Port Output Control Register bit to 1.

TH and TL—Timer High and Low Bytes
These 2 bytes, {TMRH[7:0], TMRL[7:0]}, contain the current 16-bit timer count value.

Timer Reload High and Low Byte Registers

The Timer 0–1 Reload High and Low Byte (TxRH and TxRL) Registers (Table 41) store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the Timer Reload High Byte register are stored in a temporary holding register. When a write to the Timer Reload Low Byte Register occurs, the temporary holding register value is written to the Timer High Byte Register. This operation allows simultaneous updates of the 16-bit Timer Reload value.

In COMPARE mode, the Timer Reload High and Low Byte Registers store the 16-bit Compare value.

Table 41. Timer 0–1 Reload High Byte Register (TxRH)

BITS	7	6	5	4	3	2	1	0
FIELD	TRH							
RESET	1							
R/W	R/W							
ADDR	F02H, F0AH							

Table 42. Timer 0–1 Reload Low Byte Register (TxRL)

BITS	7	6	5	4	3	2	1	0
FIELD	TRL							
RESET	1							
R/W	R/W							
ADDR	F03H, F0BH							

TRH and TRL—Timer Reload Register High and Low
These two bytes form the 16-bit Reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value which initiates a timer reload to `0001H`. In COMPARE mode, these two bytes form the 16-bit Compare value.

Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers (Table 43 and Table 44) are used for Pulse-Width Modulator (PWM) operations. These registers also store the Capture values for the CAPTURE and CAPTURE/COMPARE modes.

Receiving IrDA Data

Data received from the infrared transceiver through the IR_RXD signal through the RXD pin is decoded by the Infrared Endec and passed to the UART. The UART's baud rate clock is used by the Infrared Endec to generate the demodulated signal (RXD) that drives the UART. Each UART/Infrared data bit is 16-clocks wide. Figure 19 displays data reception. When the Infrared Endec is enabled, the UART's RXD signal is internal to the Z8 Encore! XP® F0822 Series products while the IR_RXD signal is received through the RXD pin.

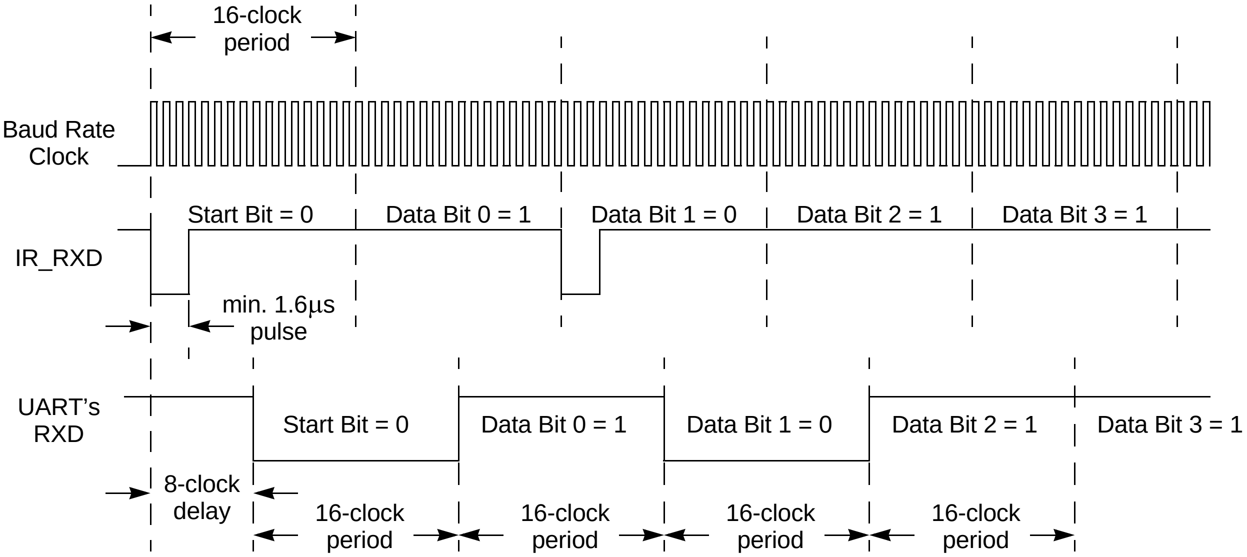


Figure 19. Infrared Data Reception

! **Caution:** *The system clock frequency must be at least 1.0 MHz to ensure proper reception of the 1.6 µs minimum width pulses allowed by the IrDA standard.*

Endec Receiver Synchronization

The IrDA receiver uses a local baud rate clock counter (0 to 15 clock periods) to generate an input stream for the UART and to create a sampling window for detection of incoming pulses. The generated UART input (UART RXD) is delayed by 8 baud rate clock periods with respect to the incoming IrDA data stream. When a falling edge in the input data stream is detected, the Endec counter is reset. When the count reaches a value of 8, the UART RXD value is updated to reflect the value of the decoded data. When the count reaches 12 baud clock periods, the sampling window for the next incoming pulse opens. The window remains open until the count again reaches 8 (or in other words 24 baud clock periods since the previous pulse was detected). This gives the Endec a sampling window

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

If the slave does not acknowledge, the Not Acknowledge interrupt occurs (NCKI bit is set in the Status register, ACK bit is cleared). Software responds to the Not Acknowledge interrupt by setting the STOP bit and clearing the TXI bit. The I2C Controller sends the STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore the following steps).

7. The I^{2}C Controller shifts in the byte of data from the I^{2}C Slave on the SDA signal. The I^{2}C Controller sends a Not Acknowledge to the I^{2}C Slave if the NAK bit is set (last byte), else it sends an Acknowledge.
8. The I^{2}C Controller asserts the Receive interrupt (RDRF bit set in the Status register).
9. Software responds by reading the I^{2}C Data Register which clears the RDRF bit. If there is only one more byte to receive, set the NAK bit of the I^{2}C Control Register.
10. If there are more bytes to transfer, return to Step 7.
11. After the last byte is shifted in, a Not Acknowledge interrupt is generated by the I^{2}C Controller.
12. Software responds by setting the `STOP` bit of the I^{2}C Control Register.
13. A STOP condition is sent to the I^{2}C Slave, the STOP and NCKI bits are cleared.

Read Transaction with a 10-Bit Address

Figure 31 displays the read transaction format for a 10-bit addressed slave. The shaded regions indicate data transferred from the I^{2}C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I^{2}C Controller.

S	Slave Address 1st 7 bits	W=0	A	Slave Address 2nd Byte	A	S	Slave Address 1st 7 bits	R=1	A	Data	A	Data	$\overline{A}$	P

Figure 31. Receive Data Format for a 10-Bit Addressed Slave

The first seven bits transmitted in the first byte are `11110XX`. The two bits `XX` are the two most-significant bits of the 10-bit address. The lowest bit of the first byte transferred is the write control bit.

Follow the steps below for the data transfer procedure for a read operation to a 10-bit addressed slave:

1. Software writes `11110B` followed by the two address bits and a 0 (write) to the I^{2}C Data Register.
2. Software asserts the `START` and TXI bits of the I^{2}C Control Register.
3. The I^{2}C Controller sends the Start condition.
4. The I^{2}C Controller loads the I^{2}C Shift register with the contents of the I^{2}C Data Register.

STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore the following steps).

19. The I^{2}C Controller shifts in a byte of data from the I^{2}C Slave on the SDA signal. The I^{2}C Controller sends a Not Acknowledge to the I^{2}C Slave if the NAK bit is set (last byte), else it sends an Acknowledge.
20. The I^{2}C Controller asserts the Receive interrupt (RDRF bit set in the Status register).
21. Software responds by reading the I^{2}C Data Register which clears the RDRF bit. If there is only one more byte to receive, set the NAK bit of the I^{2}C Control Register.
22. If there are one or more bytes to transfer, return to step 19.
23. After the last byte is shifted in, a Not Acknowledge interrupt is generated by the I^{2}C Controller.
24. Software responds by setting the `STOP` bit of the I^{2}C Control Register.
25. A STOP condition is sent to the I^{2}C Slave and the STOP and NCKI bits are cleared.

I^{2}C Control Register Definitions

I^{2}C Data Register

The I^{2}C Data Register (Table 70) holds the data that is to be loaded into the I^{2}C Shift register during a write to a slave. This register also holds data that is loaded from the I^{2}C Shift register during a read from a slave. The I^{2}C Shift Register is not accessible in the Register File address space, but is used only to buffer incoming and outgoing data.

Table 70. I^{2}C Data Register (I2CDATA)

BITS	7	6	5	4	3	2	1	0
FIELD	DATA							
RESET	0							
R/W	R/W							
ADDR	F50H							

Flash Memory

The products in Z8 Encore! XP® F0822 Series feature either 8 KB (8192) or 4 KB (4096) bytes of Flash memory with Read/Write/Erase capability. The Flash memory is programmed and erased in-circuit by either user code or through the OCD.

The Flash memory array is arranged in 512-byte per page. The 512-byte page is the minimum Flash block size that can be erased. The Flash memory is divided into eight sectors which is protected from programming and erase operations on a per sector basis.

Table 80 describes the Flash memory configuration for each device in the Z8F082x family. Table 81 lists the sector address ranges. Figure 33 on page 154 displays the Flash memory arrangement.

Table 80. Flash Memory Configurations

Part Number	Flash Size	Number of Pages	Flash Memory Addresses	Sector Size	Number of Sectors	Pages per Sector
Z8F08xx	8 KB (8192)	16	0000H - 1FFFH	1 KB (1024)	8	2
Z8F04xx	4 KB (4096)	8	0000H - 0FFFH	0.5 KB (512)	8	1

Table 81. Flash Memory Sector Addresses

	Flash Sector Address Ranges	
Sector Number	**Z8F04xx**	**Z8F08xx**
0	0000H-01FFH	0000H-03FFH
1	0200H-03FFH	0400H-07FFH
2	0400H-05FFH	0800H-0BFFH
3	0600H-07FFH	0C00H-0FFFH
4	0800H-09FFH	1000H-13FFH
5	0A00H-0BFFH	1400H-17FFH
6	0C00H-0DFFH	1800H-1BFFH
7	0E00H-0FFFH	1C00H-1FFFH

Table 82. Z8 Encore! XP® F0822 Series Information Area Map

Flash Memory Address (Hex)	Function
FE00H-FE3FH	Reserved
FE40H-FE53H	**Part Number** 20-character ASCII alphanumeric code Left justified and filled with zeros
FE54H-FFFFH	Reserved

Operation

The Flash Controller provides the proper signals and timing for Byte Programming, Page Erase, and Mass Erase of the Flash memory. The Flash Controller contains a protection mechanism, using the Flash Control Register (FCTL), to prevent accidental programming or erasure. The following subsections provide details on the various operations (Lock, Unlock, Sector Protect, Byte Programming, Page Erase, and Mass Erase).

Timing Using the Flash Frequency Registers

Before performing a program or erase operation on the Flash memory, you must first configure the Flash Frequency High and Low Byte registers. The Flash Frequency registers allow programming and erasure of the Flash with system clock frequencies ranging from 20 kHz through 20 MHz (the valid range is limited to the device operating frequencies).

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 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 20 kHz, above 20 MHz, or outside of the device operating frequency range. The Flash Frequency High and Low Byte registers must be loaded with the correct value to insure proper Flash programming and erase operations.*

Option Bits

Option Bits allow user configuration of certain aspects of Z8 Encore! XP® F0822 Series operation. The feature configuration data is stored in Flash Memory and read during Reset. Features available for control through the Option Bits are:

- Watchdog Timer time-out response selection–interrupt or Reset.
- Watchdog Timer enabled at Reset.
- The ability to prevent unwanted read access to user code in Flash Memory.
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Flash Memory.
- Voltage Brownout configuration-always enabled or disabled during STOP mode to reduce STOP mode power consumption.
- Oscillator mode selection-for high, medium, and low power crystal oscillators, or external RC oscillator.

Operation

Option Bit Configuration By Reset

During any reset operation (System Reset, Reset, or Stop Mode Recovery), the Option Bits are automatically read from the Flash Memory and written to Option Configuration registers. The Option Configuration registers control operation of the devices within the Z8 Encore! XP F0822 Series. Option Bit control is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access. Each time the Option Bits are programmed or erased, the device must be Reset for the change to take place (Flash version only)

Option Bit Address Space

The first two bytes of Flash Memory at addresses `0000H` (Table 89 on page 164) and `0001H` (Table 90 on page 165) are reserved for the user programmable Option Bits. The byte at Program Memory address `0000H` configures user options. The byte at Flash Memory address `0001H` is reserved for future use and must be left in its unprogrammed state.

The host transmits a Serial Break on the DBG pin when first connecting to the Z8 Encore! XP® F0822 Series device or when recovering from an error. A Serial Break from the host resets the Auto-Baud Generator/Detector but does not reset the OCD Control Register. A Serial Break leaves the device in DEBUG mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns High. Because of the open-drain nature of the DBG pin, the host can send a Serial Break to the OCD even if the OCD is transmitting a character.

Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the OCD. If Breakpoints are enabled, the OCD idles the eZ8 CPU and enters DEBUG mode. If Breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

If breakpoints are enabled, the OCD can be configured to automatically enter DEBUG mode, or to loop on the break instruction. If the OCD is configured to loop on the BRK instruction, then the CPU is still enabled to service DMA and interrupt requests.

The loop on BRK instruction can be used to service interrupts in the background. For interrupts to be serviced in the background, there cannot be any breakpoints in the ISR. Otherwise, the CPU stops on the breakpoint in the interrupt routine. For interrupts to be serviced in the background, interrupts must also be enabled. Debugging software should not automatically enable interrupts when using this feature, since interrupts are typically disabled during critical sections of code where interrupts should not occur (such as adjusting the stack pointer or modifying shared data).

Software can poll the IDLE bit of the OCDSTAT register to determine if the OCD is looping on a BRK instruction. When software wants to stop the CPU on the BRK instruction it is looping on, software should not set the DBGMODE bit of the OCDCTL register. The CPU can have vectored to and be in the middle of an ISR when this bit gets set. Instead, software must clear the BRKLP bit. This allows the CPU to finish the ISR it is in and return the BRK instruction. When the CPU returns to the BRK instruction it was previously looping on, it automatically sets the DBGMODE bit and enter DEBUG mode.

Software should also note that the majority of the OCD commands are still disabled when the eZ8 CPU is looping on a BRK instruction. The eZ8 CPU must be stopped and the part must be in DEBUG mode before these commands can be issued.

Breakpoints in Flash Memory

The BRK instruction is opcode 00H, which corresponds to the fully programmed state of a byte in Flash memory. To implement a Breakpoint, write 00H to the desired address, overwriting the current instruction. To remove a Breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

A “reset and stop” function can be achieved by writing `81H` to this register. A “reset and go” function can be achieved by writing `41H` to this register. If the device is in DEBUG mode, a “run” function can be implemented by writing `40H` to this register.

Table 94. OCD Control Register (OCDCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	DBGMODE	BRKEN	DBGACK	BRKLOOP	BRKPC	BRKZRO	Reserved	RST
RESET	0							
R/W	R/W			R				R/W

DBGMODE—Debug Mode
Setting this bit to 1 causes the device to enter DEBUG mode. When in DEBUG mode, the eZ8 CPU stops fetching new instructions. Clearing this bit causes the eZ8 CPU to start running again. This bit is automatically set when a BRK instruction is decoded and Breakpoints are enabled. If the Read Protect Option Bit is enabled, this bit can only be cleared by resetting the device, it cannot be written to 0.
0 = The Z8 Encore! XP F0822 Series device is operating in NORMAL mode.
1 = The Z8 Encore! XP F0822 Series device is in DEBUG mode.

BRKEN—Breakpoint Enable
This bit controls the behavior of the BRK instruction (opcode `00H`). By default, Breakpoints are disabled and the BRK instruction behaves like an NOP instruction. If this bit is set to 1 and a BRK instruction is decoded, the OCD takes action dependent upon the BRKLOOP bit.
0 = BRK instruction is disabled.
1 = BRK instruction is enabled.

DBGACK—Debug Acknowledge
This bit enables the debug acknowledge feature. If this bit is set to 1, then the OCD sends an Debug Acknowledge character (`FFH`) to the host when a Breakpoint occurs.
0 = Debug Acknowledge is disabled.
1 = Debug Acknowledge is enabled.

BRKLOOP—Breakpoint Loop
This bit determines what action the OCD takes when a BRK instruction is decoded if breakpoints are enabled (BRKEN is 1). If this bit is 0, then the DBGMODE bit is automatically set to 1 and the OCD enter DEBUG mode. If BRKLOOP is set to 1, then the eZ8 CPU loops on the BRK instruction.
0 = BRK instruction sets DBGMODE to 1.
1 = eZ8 CPU loops on BRK instruction.

BRKPC—Break when PC == OCDCNTR
If this bit is set to 1, then the OCDCNTR register is used as a hardware breakpoint. When the program counter matches the value in the OCDCNTR register, DBGMODE is

On-Chip Peripheral AC and DC Electrical Characteristics

Table 99 provides information on the Power-On Reset and Voltage Brownout electrical characteristics.

Table 99. Power-On Reset and Voltage Brownout Electrical Characteristics and Timing

		T_A = -40 °C to 105 °C				
Symbol	**Parameter**	**Minimum**	**Typical[1]**	**Maximum**	**Units**	**Conditions**
V_{POR}	Power-On Reset Voltage Threshold	2.15	2.40	2.60	V	$V_{DD} = V_{POR}$
V_{VBO}	Voltage Brownout Reset Voltage Threshold	2.05	2.30	2.55	V	$V_{DD} = V_{VBO}$
	V_{POR} to V_{VBO} hysteresis	50	100	–	mV	
	Starting V_{DD} voltage to ensure valid POR	–	V_{SS}	–	V	
T_{ANA}	POR Analog Delay	–	50	–	µs	$V_{DD} > V_{POR}$; T_{POR} Digital Reset delay follows T_{ANA}
T_{POR}	POR Digital Delay	–	5.0	–	ms	50 WDT Oscillator cycles (10 kHz) + 16 System Clock cycles (20 MHz)
T_{VBO}	Voltage Brownout Pulse Rejection Period	–	10	–	µs	$V_{DD} < V_{VBO}$ to generate a Reset.
T_{RAMP}	Time for VDD to transition from V_{SS} to V_{POR} to ensure valid Reset	0.10	–	100	ms	

[1] Data in the typical column is from characterization at 3.3 V and 25 °C. These values are provided for design guidance only and are not tested in production.

I^2C Timing

Figure 53 and Table 110 provide timing information for I^2C pins.

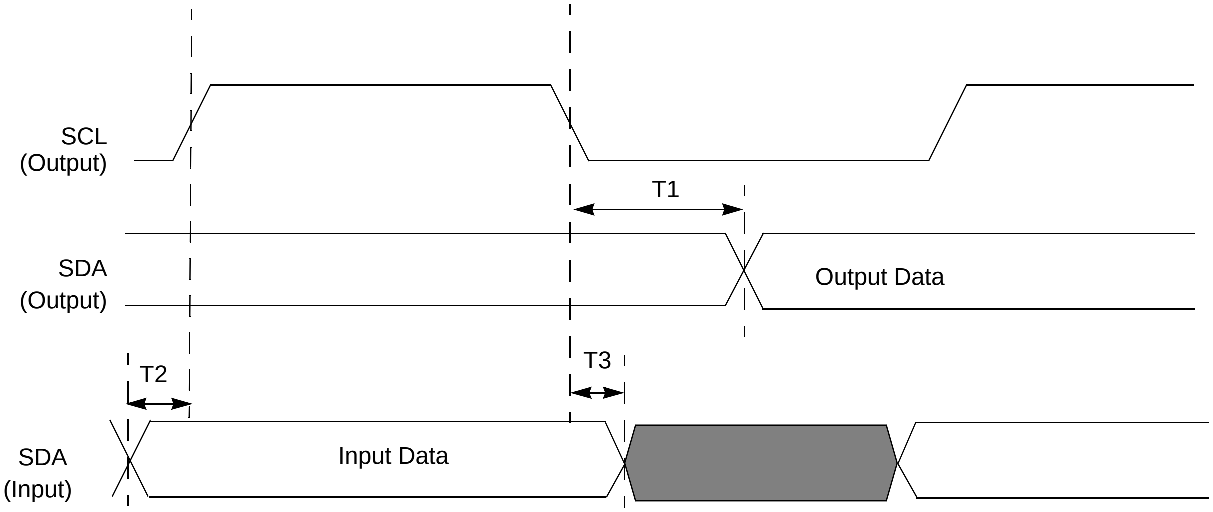


Figure 53. I^2C Timing

Table 110. I^2C Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
I^2C			
T_1	SCL Fall to SDA output delay	SCL period/4	
T_2	SDA Input to SCL rising edge Setup Time	0	
T_3	SDA Input to SCL falling edge Hold Time	0	

UART Timing

Figure 54 and Table 111 provide timing information for UART pins for the case where the Clear To Send input pin ($\overline{\text{CTS}}$) is used for flow control. In this example, it is assumed that the Driver Enable polarity has been configured to be Active Low and is represented here by $\overline{\text{DE}}$. The $\overline{\text{CTS}}$ to $\overline{\text{DE}}$ assertion delay (T1) assumes the UART Transmit Data Register has been loaded with data prior to $\overline{\text{CTS}}$ assertion.

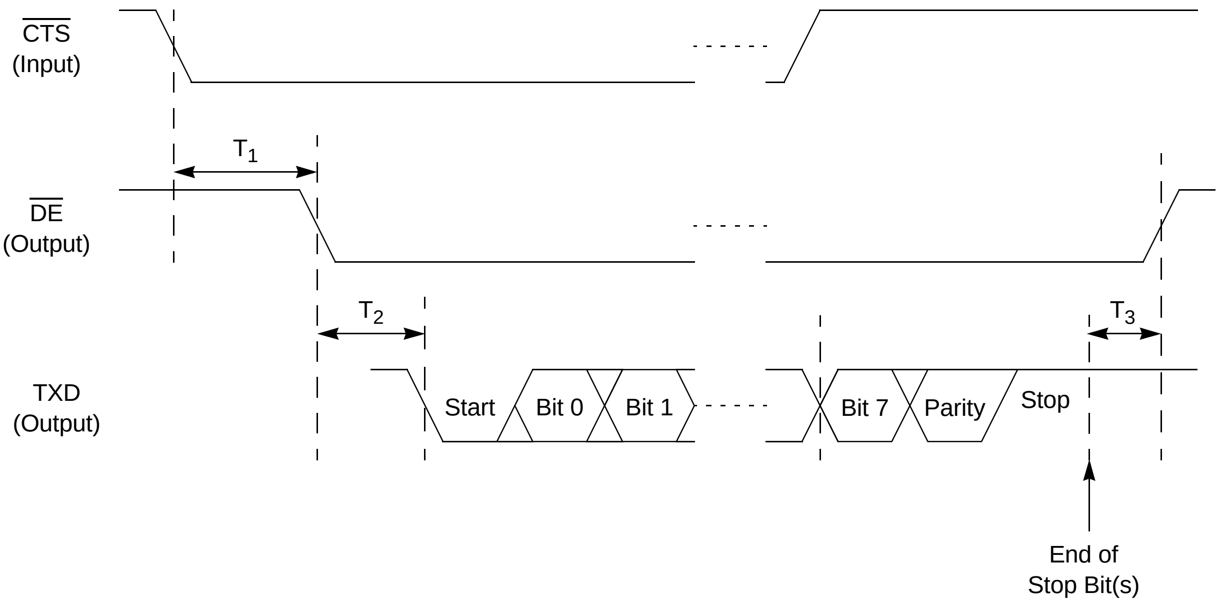


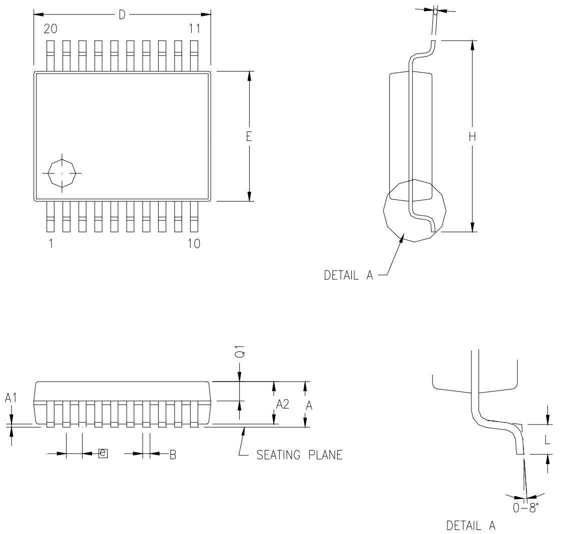
Figure 54. UART Timing with $\overline{\text{CTS}}$

Table 111. UART Timing with $\overline{\text{CTS}}$

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
T_1	$\overline{\text{CTS}}$ Fall to $\overline{\text{DE}}$ Assertion Delay	2 * XIN period	2 * XIN period + 1 Bit period
T_2	$\overline{\text{DE}}$ Assertion to TXD Falling Edge (Start) Delay	1 Bit period	1 Bit period + 1 * XIN period
T_3	End of Stop Bit(s) to $\overline{\text{DE}}$ Deassertion Delay	1 * XIN period	2 * XIN period

Packaging

Figure 60 displays the 20-pin SSOP package available for Z8 Encore! XP® F0822 Series devices.

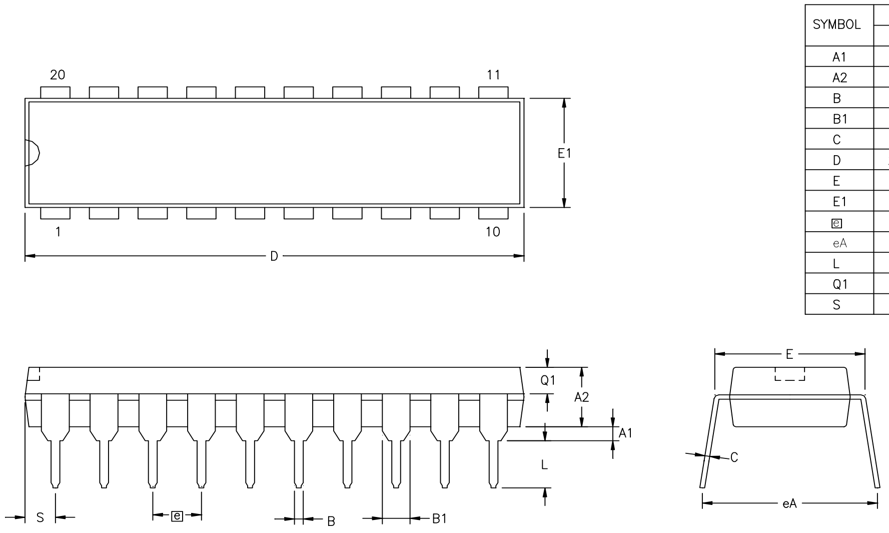


SYMBOL	MILLIMETER			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.73	1.85	1.98	0.068	0.073	0.078
A1	0.05	0.13	0.21	0.002	0.005	0.008
A2	1.68	1.73	1.83	0.066	0.068	0.072
B	0.25	0.30	0.38	0.010	0.012	0.015
C	0.13	0.15	0.22	0.005	0.006	0.009
D	7.07	7.20	7.33	0.278	0.283	0.289
E	5.20	5.30	5.38	0.205	0.209	0.212
e	0.65 BSC			0.0256 BSC		
H	7.65	7.80	7.90	0.301	0.307	0.311
L	0.56	0.75	0.94	0.022	0.030	0.037
Q1	0.74	0.78	0.82	0.029	0.031	0.032

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

Figure 60. 20-Pin Small Shrink Outline Package (SSOP)

Figure 61 displays the 20-pin PDIP package available for Z8 Encore! XP F0822 Series devices.



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.38	0.81	.015	.032
A2	3.25	3.68	.128	.145
B	0.41	0.51	.016	.020
B1	1.47	1.57	.058	.062
C	0.20	0.30	.008	.012
D	25.65	26.16	1.010	1.030
E	7.49	8.26	.295	.325
E1	6.10	6.65	.240	.262
e	2.54 BSC		.100 BSC	
eA	7.87	9.14	.310	.360
L	3.18	3.43	.125	.135
Q1	1.42	1.65	.056	.065
S	1.52	1.65	.060	.065

CONTROLLING DIMENSIONS : INCH

Figure 61. 20-Pin Plastic Dual-Inline Package (PDIP)