



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	I ² C, 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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0811hh020ec

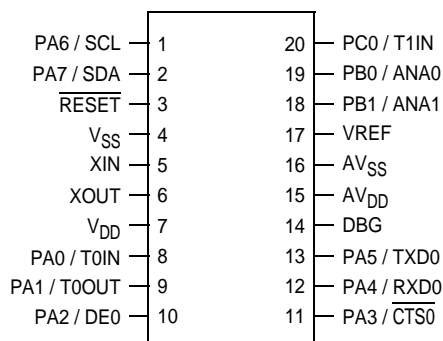


Figure 2. Z8F0821 and Z8F0421 in 20-Pin SSOP and PDIP Packages

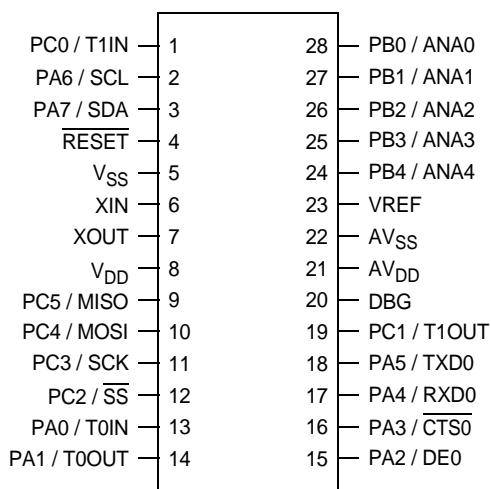


Figure 3. Z8F0822 and Z8F0422 in 28-Pin SOIC and PDIP Packages

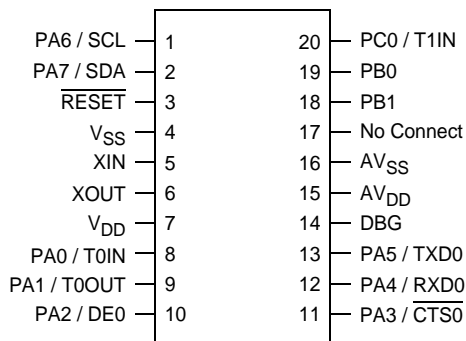
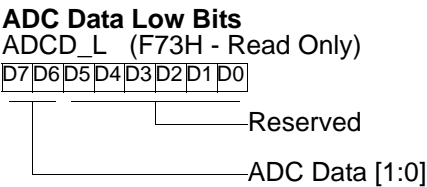
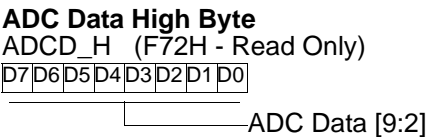
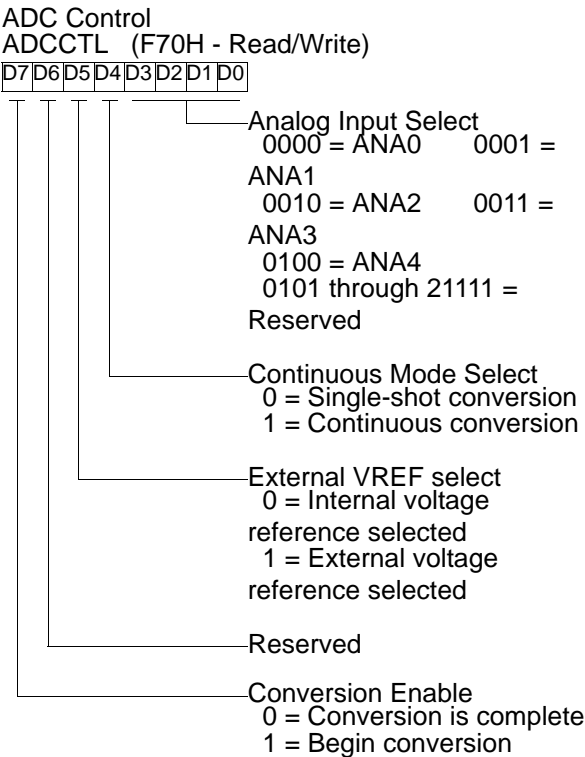
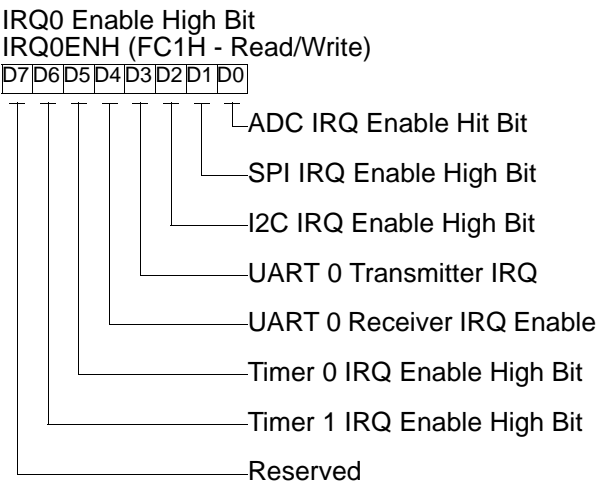
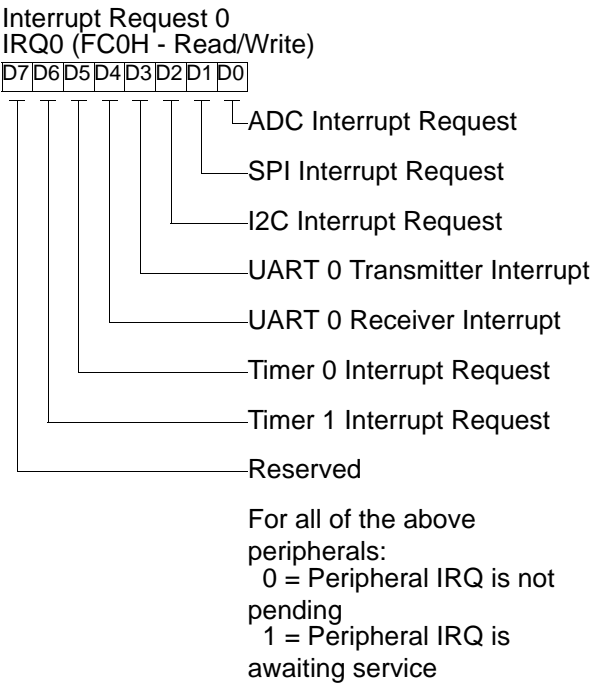


Figure 4. Z8F0811 and Z8F0411 in 20-Pin SSOP and PDIP Packages





Watchdog Timer

Watchdog Timer (WDT) protects against corrupt or unreliable software, power faults, and other system-level problems which can place the Z8 Encore! XP® F0822 Series device into unsuitable operating states. It includes the following features:

- On-chip RC oscillator.
- A selectable time-out response—Reset or Interrupt.
- 24-bit programmable time-out value.

Operation

WDT is a retriggerable one-shot timer that resets or interrupts the Z8 Encore! XP F0822 Series device when the WDT reaches its terminal count. It uses its own dedicated on-chip RC oscillator as its clock source. The WDT has only two modes of operation—ON and OFF. When enabled, it always counts and must be refreshed to prevent a time-out. An enable is performed by executing the WDT instruction or by setting the WDT_AO Option Bit. The WDT_AO bit enables the WDT to operate all the time, even if a WDT instruction has not been executed.

The WDT is a 24-bit reloadable downcounter that uses three 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is given by the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{10}$$

where the WDT reload value is the decimal value of the 24-bit value given by {WDTU[7:0], WDTH[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10 kHz. WDT cannot be refreshed once it reaches 000002H. The WDT Reload Value must not be set to values below 000004H. Table 47 provides information on approximate time-out delays for minimum and maximum WDT reload values.

Table 47. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value	WDT Reload Value	Approximate Time-Out Delay (with 10 kHz typical WDT Oscillator Frequency)	
(Hex)	(Decimal)	Typical	Description
000004	4	400 μs	Minimum time-out delay
FFFFFF	16,777,215	1677.5 s	Maximum time-out delay

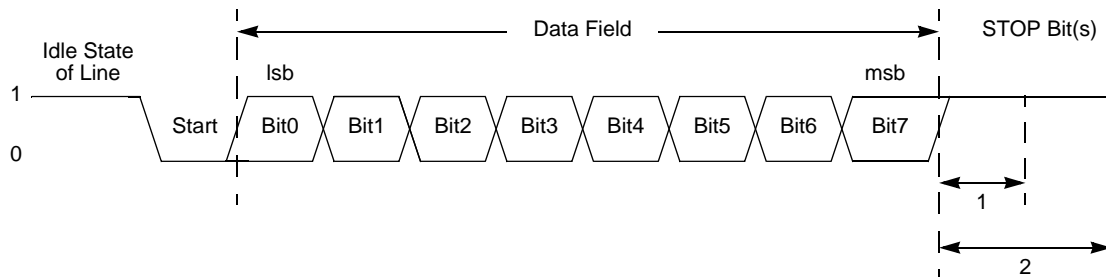


Figure 12. UART Asynchronous Data Format without Parity

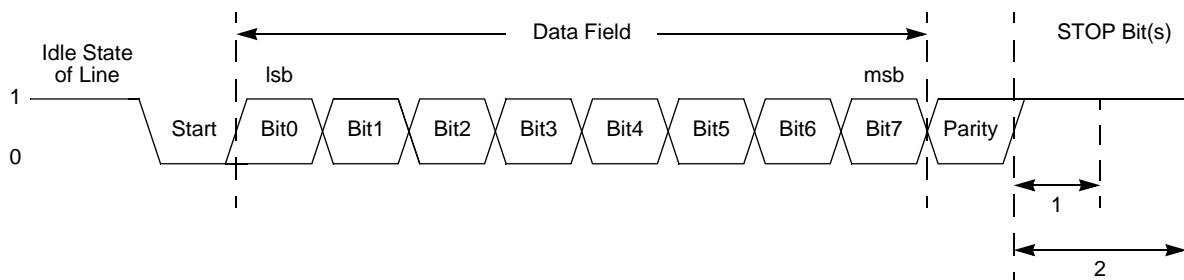


Figure 13. UART Asynchronous Data Format with Parity

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.

Clear To Send Operation

The CTS pin, if enabled by the CTSE bit of the UART Control 0 register, performs flow control on the outgoing transmit datastream. The Clear To Send ($\overline{\text{CTS}}$) input pin is sampled one system clock before beginning any new character transmission. To delay transmission of the next data character, an external receiver must deassert $\overline{\text{CTS}}$ at least one system clock cycle before a new data transmission begins. For multiple character transmissions, this would be done during STOP bit transmission. If $\overline{\text{CTS}}$ deasserts in the middle of a character transmission, the current character is sent completely.

Multiprocessor (9-bit) Mode

The UART has a MULTIPROCESSOR (9-bit) mode that uses an extra (9th) bit for selective communication when a number of processors share a common UART bus. In MULTIPROCESSOR mode (also referred to as 9-bit mode), the multiprocessor bit is transmitted following the 8-bits of data and immediately preceding the STOP bit(s) as displayed in Figure 14. The character format is as displayed in Figure 14.

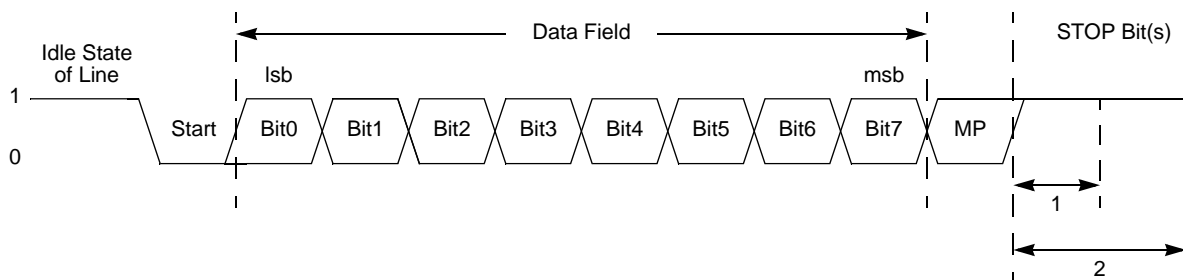


Figure 14. UART Asynchronous MULTIPROCESSOR Mode Data Format

In MULTIPROCESSOR (9-bit) mode, the Parity bit location (9th bit) 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 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 need to access the UART when it receives data directed to other devices on the

0, then reading the UART Receive Data Register clears this bit.

0 = No overrun error occurred.

1 = An overrun error occurred.

FE—Framing Error

This bit indicates that a framing error (no STOP bit following data reception) was detected. Reading the UART Receive Data Register clears this bit.

0 = No framing error occurred.

1 = A framing error occurred.

BRKD—Break Detect

This bit indicates that a break occurred. If the data bits, parity/multiprocessor bit, and STOP bit(s) are all zeros then this bit is set to 1. Reading the UART Receive Data Register clears this bit.

0 = No break occurred.

1 = A break occurred.

TDRE—Transmitter Data Register Empty

This bit indicates that the UART Transmit Data Register is empty and ready for additional data. Writing to the UART Transmit Data Register resets this bit.

0 = Do not write to the UART Transmit Data Register.

1 = The UART Transmit Data Register is ready to receive an additional byte to be transmitted.

TXE—Transmitter Empty

This bit indicates that the transmit shift register is empty and character transmission is finished.

0 = Data is currently transmitting.

1 = Transmission is complete.

CTS— $\overline{\text{CTS}}$ Signal

When this bit is read it returns the level of the $\overline{\text{CTS}}$ signal.

UART Status 1 Register

This register contains multiprocessor control and status bits.

Table 55. UART Status 1 Register (U0STAT1)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved						NEWFRM	MPRX
RESET	0							
R/W	R				R/W		R	
ADDR	F44H							

The baud rate is set by the UART's Baud Rate Generator and supports IrDA standard baud rates from 9600 baud to 115.2 Kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the Infrared Endec. The Infrared Endec data rate is calculated using the following equation.

$$\text{Infrared Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR_TXD) that drives the infrared transceiver. Each UART/Infrared data bit is 16-clocks wide. If the data to be transmitted is 1, the IR_TXD signal remains low for the full 16-clock period. If the data to be transmitted is 0, a 3-clock high pulse is output following a 7-clock low period. After the 3-clock high pulse, a 6-clock low pulse is output to complete the full 16-clock data period. Figure 18 displays IrDA data transmission. When the Infrared Endec is enabled, the UART's TXD signal is internal to the Z8 Encore! XP® F0822 Series products while the IR_TXD signal is output through the TXD pin.

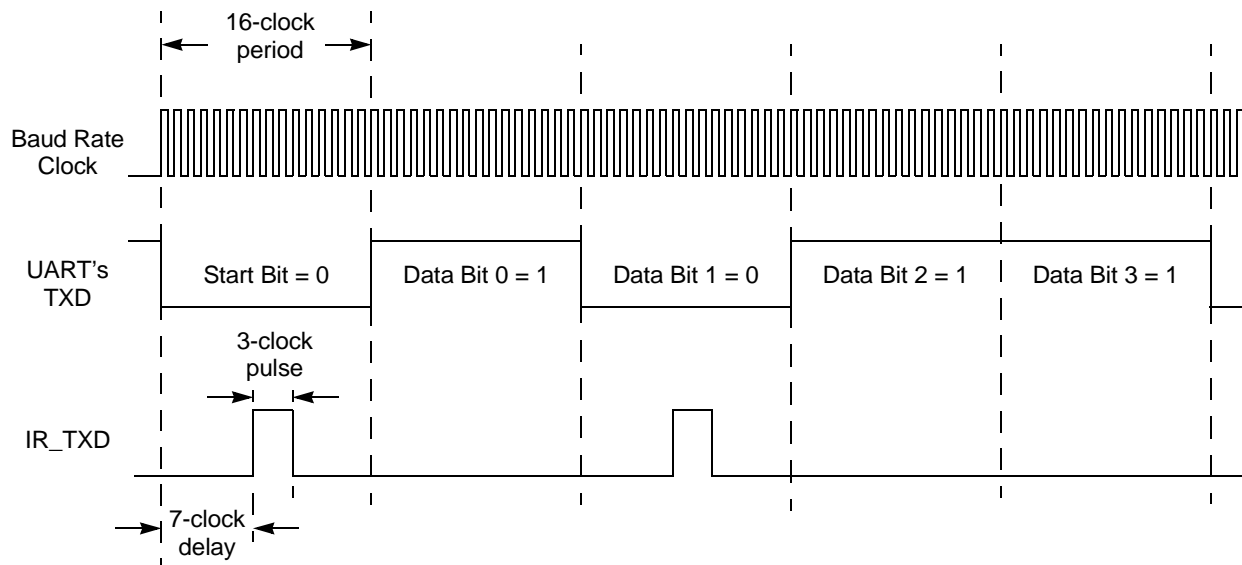


Figure 18. Infrared Data Transmission

necessary for \overline{SS} to deassert between characters to generate the interrupt. The SPI in SLAVE mode also generates an interrupt if the \overline{SS} signal deasserts prior to transfer of all the bits in a character (see description of Slave Abort Error). Writing a 1 to the IRQ bit in the SPI Status Register clears the pending SPI interrupt request. The IRQ bit must be cleared to 0 by the ISR to generate future interrupts. To start the transfer process, an SPI interrupt can be forced by software writing a 1 to the STR bit in the $SPICTL$ Register.

If the SPI is disabled, an SPI interrupt can be generated by a BRG time-out. This timer function must be enabled by setting the $BIRQ$ bit in the $SPICTL$ Register. This BRG time-out does not set the IRQ bit in the $SPISTAT$ Register, just the SPI interrupt bit in the interrupt controller.

SPI Baud Rate Generator

In SPI MASTER mode, the BRG creates a lower frequency serial clock (SCK) for data transmission synchronization between the Master and the external Slave. The input to the BRG is the system clock. The SPI Baud Rate High and Low Byte Registers combine to form a 16-bit reload value, $BRG[15:0]$, for the SPI Baud Rate Generator. The SPI baud rate is calculated using the following equation:

$$\text{SPI Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{2 \times BRG[15:0]}$$

Minimum baud rate is obtained by setting $BRG[15:0]$ to 0000H for a clock divisor value of $(2 \times 65536 = 131072)$.

When the SPI is disabled, BRG functions as a basic 16-bit timer with interrupt on time-out. Follow the steps below to configure BRG as a timer with interrupt on time-out:

1. Disable the SPI by clearing the $SPIEN$ bit in the SPI Control Register to 0.
2. Load the desired 16-bit count value into the SPI Baud Rate High and Low Byte registers.
3. Enable BRG timer function and associated interrupt by setting the $BIRQ$ bit in the SPI Control Register to 1.

When configured as a general-purpose timer, the interrupt interval is calculated using the following equation:

$$\text{Interrupt Interval (s)} = \text{System Clock Period (s)} \times BRG[15:0]$$

I²C Status Register

The Read-only I²C Status register (Table 71) indicates the status of the I²C Controller.

Table 71. I²C Status Register (I2CSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	TDRE	RDRF	ACK	10B	RD	TAS	DSS	NCKI
RESET	1	0						
R/W	R							
ADDR	F51H							

TDRE—Transmit Data Register Empty

When the I²C Controller is enabled, this bit is 1 when the I²C Data Register is empty. When this bit is set, an interrupt is generated if the TXI bit is set, except when the I²C Controller is shifting in data during the reception of a byte or when shifting an address and the RD bit is set. This bit is cleared by writing to the I2CDATA register.

RDRF—Receive Data Register Full

This bit is set = 1 when the I²C Controller is enabled and the I²C Controller has received a byte of data. When asserted, this bit causes the I²C Controller to generate an interrupt. This bit is cleared by reading the I²C Data Register (unless the read is performed using execution of the OCD's Read Register command).

ACK—Acknowledge

This bit indicates the status of the Acknowledge for the last byte transmitted or received. When set, this bit indicates that an Acknowledge occurred for the last byte transmitted or received. This bit is cleared when IEN = 0 or when a Not Acknowledge occurred for the last byte transmitted or received. It is not reset at the beginning of each transaction and is not reset when this register is read.

! Caution: *Software must be cautious in making decisions based on this bit within a transaction because software cannot tell when the bit is updated by hardware. In the case of write transactions, the I²C pauses at the beginning of the Acknowledge cycle if the next transmit data or address byte has not been written (TDRE = 1) and STOP and START = 0. In this case the ACK bit is not updated until the transmit interrupt is serviced and the Acknowledge cycle for the previous byte completes. For examples on usage of the ACK bit, see Address Only Transaction with a 7-bit Address on page 131 and Address Only Transaction with a 10-bit Address on page 133.*

10B—10-Bit Address

This bit indicates whether a 10-bit or 7-bit address is being transmitted. After the START bit is set, if the five most-significant bits of the address are 11110B, this bit is set. When set, it is reset once the first byte of the address has been sent.

RD—Read

This bit indicates the direction of transfer of the data. It is active High during a read. The status of this bit is determined by the least-significant bit of the I²C Shift register after the START bit is set.

TAS—Transmit Address State

This bit is active High while the address is being shifted out of the I²C Shift Register.

DSS—Data Shift State

This bit is active High while data is being shifted to or from the I²C Shift Register.

NCKI—NACK Interrupt

This bit is set high when a Not Acknowledge condition is received or sent and neither the START nor the STOP bit is active. When set, this bit generates an interrupt that can only be cleared by setting the START or STOP bit, allowing you to specify whether you want to perform a STOP or a repeated START.

I²C Control Register

The I²C Control Register (Table 72) enables the I²C operation.

Table 72. I²C Control Register (I2CCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	IEN	START	STOP	BIRQ	TXI	NAK	FLUSH	FILTEN
RESET	0							
R/W	R/W	R/W1	R/W1	R/W	R/W	R/W1	W1	R/W
ADDR	F52H							

IEN—I²C Enable

1 = The I²C transmitter and receiver are enabled.

0 = The I²C transmitter and receiver are disabled.

START—Send Start Condition

This bit sends the Start condition. Once asserted, it is cleared by the I²C Controller after it sends the START condition or if the IEN bit is deasserted. If this bit is 1, it cannot be cleared to 0 by writing to the register. After this bit is set, the Start condition is sent if there is data in the I²C Data or I²C Shift register. If there is no data in one of these registers, the I²C Controller waits until the data register is written. If this bit is set while the I²C Controller is shifting out data, it generates a START condition after the byte shifts and the acknowledge phase completes. If the STOP bit is also set, it also waits until the STOP condition is sent before sending the START condition.

STOP—Send Stop Condition

This bit causes the I²C Controller to issue a STOP condition after the byte in the I²C Shift register has completed transmission or after a byte is received in a receive operation. Once

On-Chip Oscillator

Z8 Encore! XP[®] F0822 Series products feature an on-chip oscillator for use with external crystals with frequencies from 32 kHz to 20 MHz. In addition, the oscillator can support external RC networks with oscillation frequencies up to 4 MHz or ceramic resonators with oscillation frequencies up to 20 MHz. This oscillator generates the primary system clock for the internal eZ8 CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin can also accept a CMOS-level clock input signal (32 kHz–20 MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected.

When configured for use with crystal oscillators or external clock drivers, the frequency of the signal on the X_{IN} input pin determines the frequency of the system clock (that is, no internal clock divider). In RC operation, the system clock is driven by a clock divider (divide by 2) to ensure 50% duty cycle.

Operating Modes

Z8 Encore! XP F0822 Series products support 4 different oscillator modes:

- On-chip oscillator configured for use with external RC networks (<4 MHz).
- Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).
- Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 10.0 MHz).
- Maximum power for use with high frequency crystals or ceramic resonators (8.0 MHz to 20.0 MHz).

The oscillator mode is selected through user-programmable Option Bits. For more information, see Option Bits on page 163.

Crystal Oscillator Operation

Figure 34 on page 168 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 91 on page 168. Resistor R1 is optional and limits total power dissipation by the crystal. The printed circuit board layout 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.

DC Characteristics

Table 97 lists the DC characteristics of the Z8 Encore! XP[®] F0822 Series products. All voltages are referenced to V_{SS} , the primary system ground.

Table 97. DC Characteristics

Symbol	Parameter	$T_A = -40\text{ }^{\circ}\text{C to }105\text{ }^{\circ}\text{C}$			Units	Conditions
		Minimum	Typical	Maximum		
V_{DD}	Supply Voltage	2.7	—	3.6	V	
V_{IL1}	Low Level Input Voltage	-0.3	—	$0.3 \cdot V_{DD}$	V	For all input pins except $\overline{\text{RESET}}$, DBG, and XIN.
V_{IL2}	Low Level Input Voltage	-0.3	—	$0.2 \cdot V_{DD}$	V	For $\overline{\text{RESET}}$, DBG, and XIN.
V_{IH1}	High Level Input Voltage	$0.7 \cdot V_{DD}$	—	5.5	V	Ports A and C pins when their programmable pull-ups are disabled.
V_{IH2}	High Level Input Voltage	$0.7 \cdot V_{DD}$	—	$V_{DD} + 0.3$	V	Port B pins. Ports A and C pins when their programmable pull-ups are enabled.
V_{IH3}	High Level Input Voltage	$0.8 \cdot V_{DD}$	—	$V_{DD} + 0.3$	V	$\overline{\text{RESET}}$, DBG, and XIN pins.
V_{OL1}	Low Level Output Voltage	—	—	0.4	V	$I_{OL} = 2\text{ mA}$; $V_{DD} = 3.0\text{ V}$ High Output Drive disabled.
V_{OH1}	High Level Output Voltage	2.4	—	—	V	$I_{OH} = -2\text{ mA}$; $V_{DD} = 3.0\text{ V}$ High Output Drive disabled.
V_{OL2}	Low Level Output Voltage High Drive	—	—	0.6	V	$I_{OL} = 20\text{ mA}$; $V_{DD} = 3.3\text{ V}$ High Output Drive enabled $T_A = -40\text{ }^{\circ}\text{C to }+70\text{ }^{\circ}\text{C}$
V_{OH2}	High Level Output Voltage High Drive	2.4	—	—	V	$I_{OH} = -20\text{ mA}$; $V_{DD} = 3.3\text{ V}$ High Output Drive enabled; $T_A = -40\text{ }^{\circ}\text{C to }+70\text{ }^{\circ}\text{C}$
V_{OL3}	Low Level Output Voltage High Drive	—	—	0.6	V	$I_{OL} = 15\text{ mA}$; $V_{DD} = 3.3\text{ V}$ High Output Drive enabled; $T_A = +70\text{ }^{\circ}\text{C to }+105\text{ }^{\circ}\text{C}$
V_{OH3}	High Level Output Voltage High Drive	2.4	—	—	V	$I_{OH} = 15\text{ mA}$; $V_{DD} = 3.3\text{ V}$ High Output Drive enabled; $T_A = +70\text{ }^{\circ}\text{C to }+105\text{ }^{\circ}\text{C}$

General Purpose I/O Port Output Timing

Figure 49 and Table 106 provide timing information for GPIO Port pins.

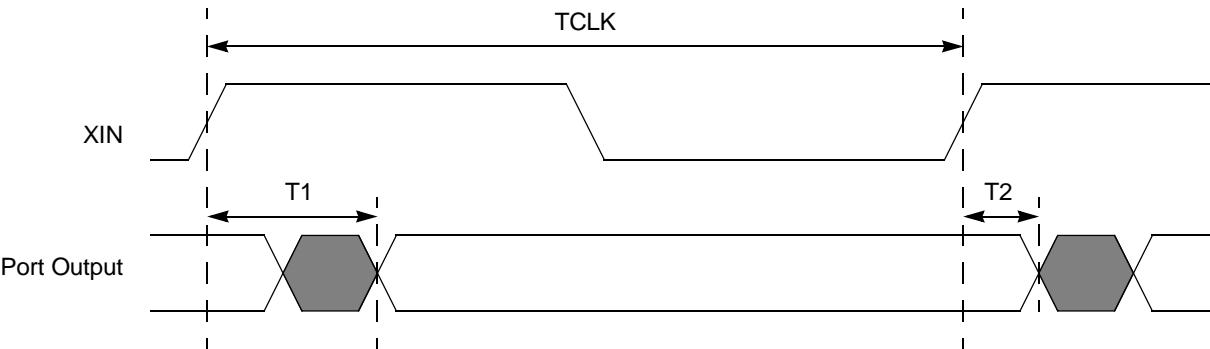


Figure 49. GPIO Port Output Timing

Table 106. GPIO Port Output Timing

Parameter Abbreviation		Delay (ns)	
		Minimum	Maximum
GPIO Port pins			
T ₁	XIN Rise to Port Output Valid Delay	–	15
T ₂	XIN Rise to Port Output Hold Time	2	–

On-Chip Debugger Timing

Figure 50 and Table 107 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4 μ s maximum rise and fall time.

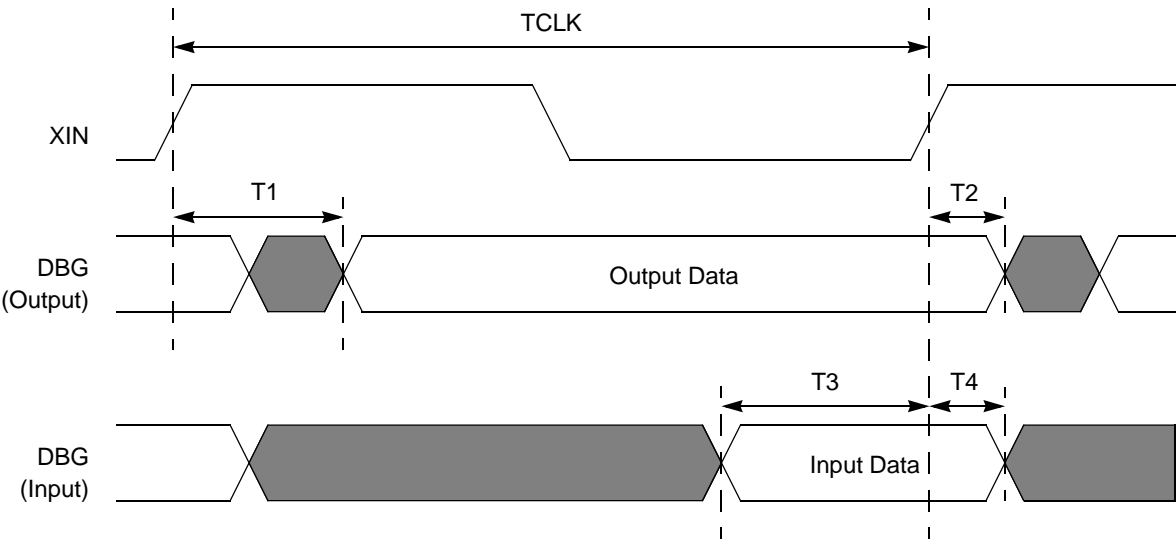


Figure 50. On-Chip Debugger Timing

Table 107. On-Chip Debugger Timing

Parameter Abbreviation		Delay (ns)	
		Minimum	Maximum
DBG			
T ₁	XIN Rise to DBG Valid Delay	–	15
T ₂	XIN Rise to DBG Output Hold Time	2	–
T ₃	DBG to XIN Rise Input Setup Time	10	–
T ₄	DBG to XIN Rise Input Hold Time	5	–
	DBG frequency		System Clock/4

Assembly Language Syntax

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

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

Table 113. Assembly Language Syntax Example 1

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

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

Table 114. Assembly Language Syntax Example 2

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

See the device-specific Product Specification to determine the exact register file range available. The register file size varies, depending on the device type.

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status flags, and address modes are represented by a notational shorthand that is described in Table 115 on page 211.

Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 57 and Table 127 on page 230. Figure 58 on page 231 and Figure 59 on page 232 provide information on each of the eZ8 CPU instructions.

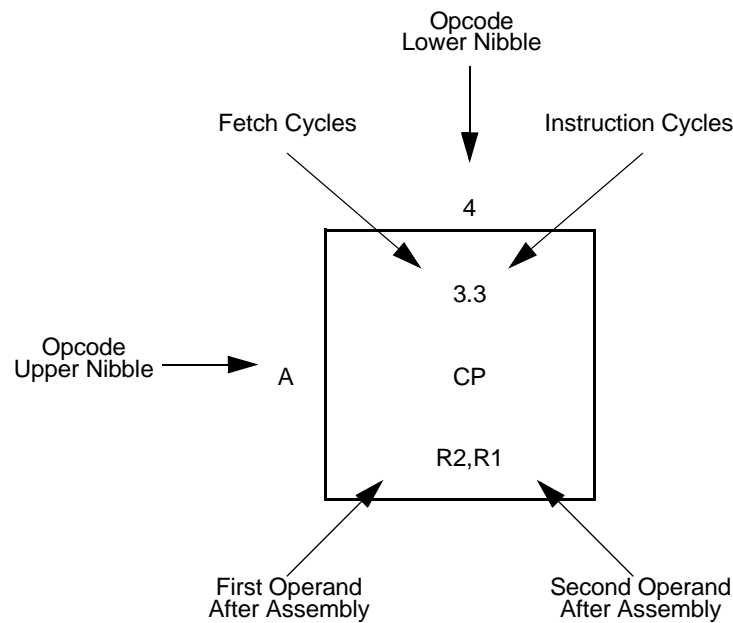


Figure 57. Opcode Map Cell Description

		Lower Nibble (Hex)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Upper Nibble (Hex)	0																
	1																
	2																
	3																
	4																
	5																
	6																
	7																
	8																
	9																
	A			3.3 CPC r1,r2	3.4 CPC r1,lr2	4.3 CPC R2,R1	4.4 CPC IR2,R1	4.3 CPC R1,IM	4.4 CPC IR1,IM	5.3 CPCX ER2,ER1	5.3 CPCX IM,ER1						
	B																
	C	3.2 SRL R1	3.3 SRL IR1														
	D																
	E																
	F																

Figure 59. Second Opcode Map after 1FH

Ir 211
 IRR 211
 Irr 211
 p 211
 R 211
 r 211
 RA 211
 RR 211
 rr 211
 vector 211
 X 211
 notational shorthand 211

O

OCD

architecture 171
 auto-baud detector/generator 174
 baud rate limits 174
 block diagram 171
 breakpoints 175
 commands 176
 control register 181
 data format 173
 DBG pin to RS-232 Interface 172
 debug mode 173
 debugger break 217
 interface 171
 serial errors 174
 status register 183
 timing 202

OCD commands

execute instruction (12H) 181
 read data memory (0DH) 180
 read OCD control register (05H) 179
 read OCD revision (00H) 178
 read OCD status register (02H) 178
 read program counter (07H) 179
 read program memory (0BH) 180
 read program memory CRC (0EH) 181
 read register (09H) 179
 read runtime counter (03H) 178
 step instruction (10H) 181
 stuff instruction (11H) 181

write data memory (0CH) 180
 write OCD control register (04H) 178
 write program counter (06H) 179
 write program memory (0AH) 180
 write register (08H) 179
 on-chip debugger 5
 on-chip debugger (OCD) 171
 on-chip debugger signals 11
 on-chip oscillator 167
 one-shot mode 81
 opcode map
 abbreviations 230
 cell description 229
 first 231
 second after 1FH 232
 Operational Description 89
 OR 217
 ordering information 236
 ORX 217
 oscillator signals 11

P

p 211
 packaging
 PDIP 234
 part selection guide 2
 PC 212
 PDIP 234
 peripheral AC and DC electrical characteristics 195
 PHASE=0 timing (SPI) 117
 PHASE=1 timing (SPI) 118
 pin characteristics 12
 polarity 211
 POP 216
 pop using extended addressing 216
 POPX 216
 port availability, device 47
 port input timing (GPIO) 200
 port output timing, GPIO 201
 power supply signals 11
 power-down, automatic (ADC) 148
 power-on and voltage brown-out 195
 power-on reset (POR) 41