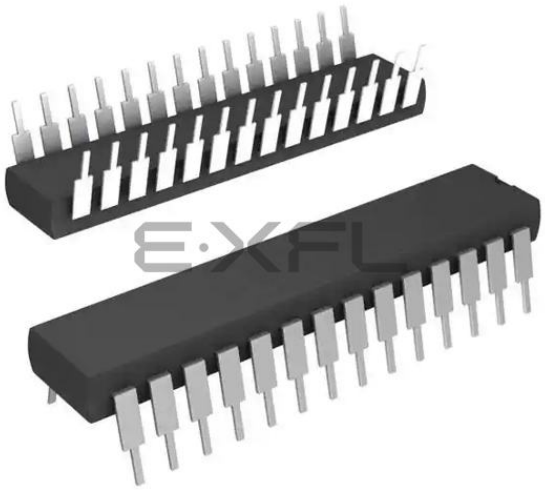


Welcome to E-XFL.COM

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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	19
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 5x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0422pj020sc

Reset and Stop Mode Recovery

The Reset Controller within the Z8 Encore! XP® F0822 Series controls Reset and Stop Mode Recovery operation. In typical operation, the following events cause a Reset to occur:

- Power-On Reset (POR)
- Voltage Brownout
- WDT time-out (when configured through the WDT_RES Option Bit to initiate a Reset)
- External $\overline{\text{RESET}}$ pin assertion
- On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)

When the Z8 Encore! XP F0822 Series device is in STOP mode, a Stop Mode Recovery is initiated by any of the following events:

- WDT time-out
- GPIO Port input pin transition on an enabled Stop Mode Recovery source
- DBG pin driven Low

Reset Types

Z8 Encore! XP F0822 Series provides two types of reset operation (System Reset and Stop Mode Recovery). The type of reset is a function of both the current operating mode of the Z8 Encore! XP F0822 Series device and the source of the Reset. Table 8 lists the types of Resets and their operating characteristics.

Table 8. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	**Control Registers**	**eZ8 CPU**	**Reset Latency (Delay)**
System Reset	Reset (as applicable)	Reset	66 WDT Oscillator cycles + 16 System Clock cycles
Stop Mode Recovery	Unaffected, except WDT_CTL register	Reset	66 WDT Oscillator cycles + 16 System Clock cycles

Power-On Reset

Each device in the Z8 Encore! XP® F0822 Series contains an internal POR circuit. The POR circuit monitors the supply voltage and holds the device in the Reset state until the supply voltage reaches a safe operating level. After the supply voltage exceeds the POR voltage threshold (V_{POR}), the POR Counter is enabled and counts 66 cycles of the WDT oscillator. After the POR counter times out, the XTAL Counter is enabled to count a total of 16 system clock pulses. The device is held in the Reset state until both the POR Counter and XTAL counter have timed out. After the Z8 Encore! XP F0822 Series device exits the POR state, the eZ8 CPU fetches the Reset vector. Following POR, the POR status bit in the Watchdog Timer Control Register (WDTCTL) is set to 1.

Figure 6 displays POR operation. See Electrical Characteristics for POR threshold voltage (V_{POR}).

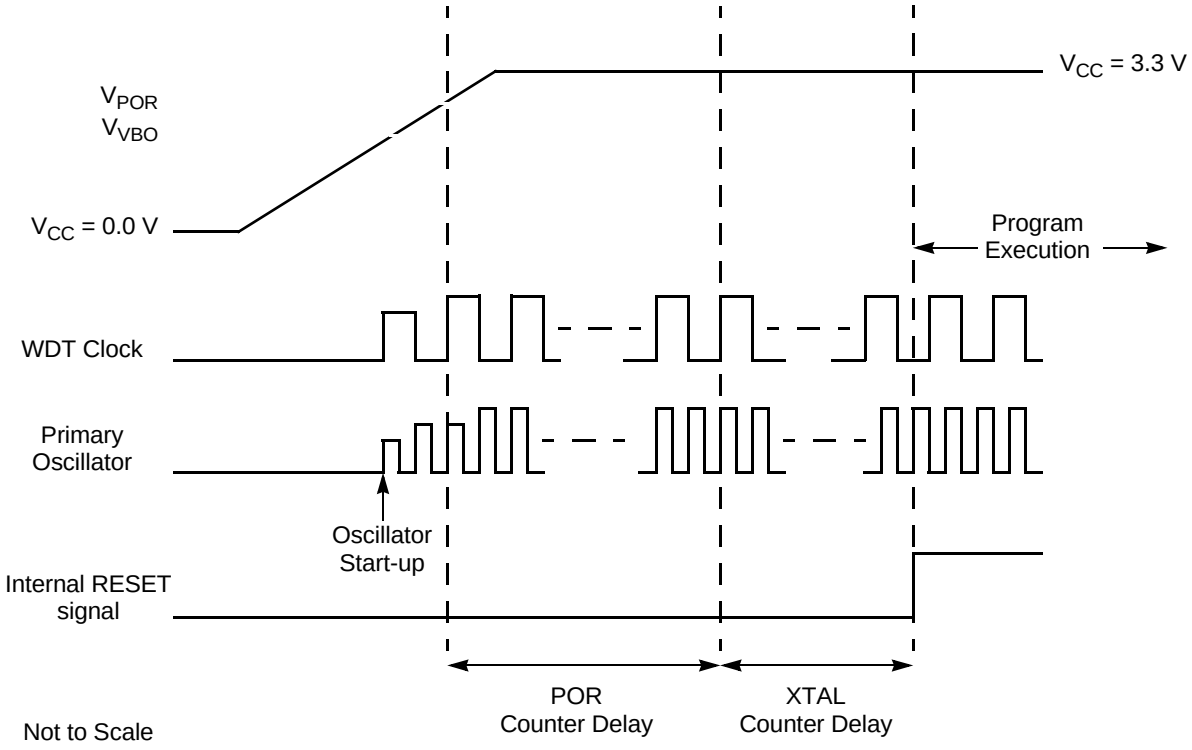


Figure 6. Power-On Reset Operation

Voltage Brownout Reset

The devices in Z8 Encore! XP F0822 Series provide low Voltage Brownout protection. The VBO circuit senses when the supply voltage drops to an unsafe level (below the VBO threshold voltage) and forces the device into the Reset state. While the supply voltage

Table 12. Port Alternate Function Mapping (Continued)

Port	Pin	Mnemonic	Alternate Function Description
Port C	PC0	T1IN	Timer 1 Input
	PC1	T1OUT	Timer 1 Output
	PC2	$\overline{SS}$	SPI Slave Select
	PC3	SCK	SPI Serial Clock
	PC4	MOSI	SPI Master Out Slave In
	PC5	MISO	SPI Master In Slave Out

GPIO Interrupts

Many of GPIO port pins are used as interrupt sources. Some port pins are configured to generate an interrupt request on either the rising edge or falling edge of the pin input signal. Other port pin interrupts generate an interrupt when any edge occurs (both rising and falling). For more details on interrupts using the GPIO pins, see GPIO Port Pin Block Diagram on page 48.

GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data, and output data. Table 13 lists the GPIO Port Registers and Sub-Registers. Use the Port A–C Address and Control Registers together to provide access to sub-registers for Port configuration and control.

Table 13. GPIO Port Registers and Sub-Registers

Port Register Mnemonic	Port Register Name
P*x*ADDR	Port A–C Address Register (selects sub-registers)
P*x*CTL	Port A–C Control Register (provides access to sub-registers)
P*x*IN	Port A–C Input Data Register
P*x*OUT	Port A–C Output Data Register
Port Sub-Register Mnemonic	**Port Register Name**
P*x*DD	Data Direction
P*x*AF	Alternate Function

pins.To determine the alternate function associated with each port pin, see GPIO Port Pin Block Diagram on page 48.

Caution: *Do not enable alternate function for GPIO port pins which do not have an associated alternate function. Failure to follow this guideline can result in unpredictable operation.*

Table 17. Port A–CA–C Alternate Function Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
RESET	0							
R/W	R/W							
ADDR	If 02H in Port A–C Address Register, accessible through the Port A–C Control Register							

AF[7:0]—Port Alternate Function enabled

0 = The port pin is in NORMAL mode and the DDx bit in the Port A–C Data Direction sub-register determines the direction of the pin.

1 = The alternate function is selected. Port pin operation is controlled by the alternate function.

Port A–C Output Control Sub-Registers

The Port A–C Output Control sub-register (Table 18) is accessed through the Port A–C Control Register by writing 03H to the Port A–C Address Register. Setting the bits in the Port A–C Output Control sub-registers to 1 configures the specified port pins for open-drain operation. These sub-registers affect the pins directly and, as a result, alternate functions are also affected.

Table 18. Port A–C Output Control Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
RESET	0							
R/W	R/W							
ADDR	If 03H in Port A–C Address Register, accessible through the Port A–C Control Register							

POC[7:0]—Port Output Control

These bits function independently of the alternate function bit and always disable the drains if set to 1.

0 = The drains are enabled for any output mode (unless overridden by the

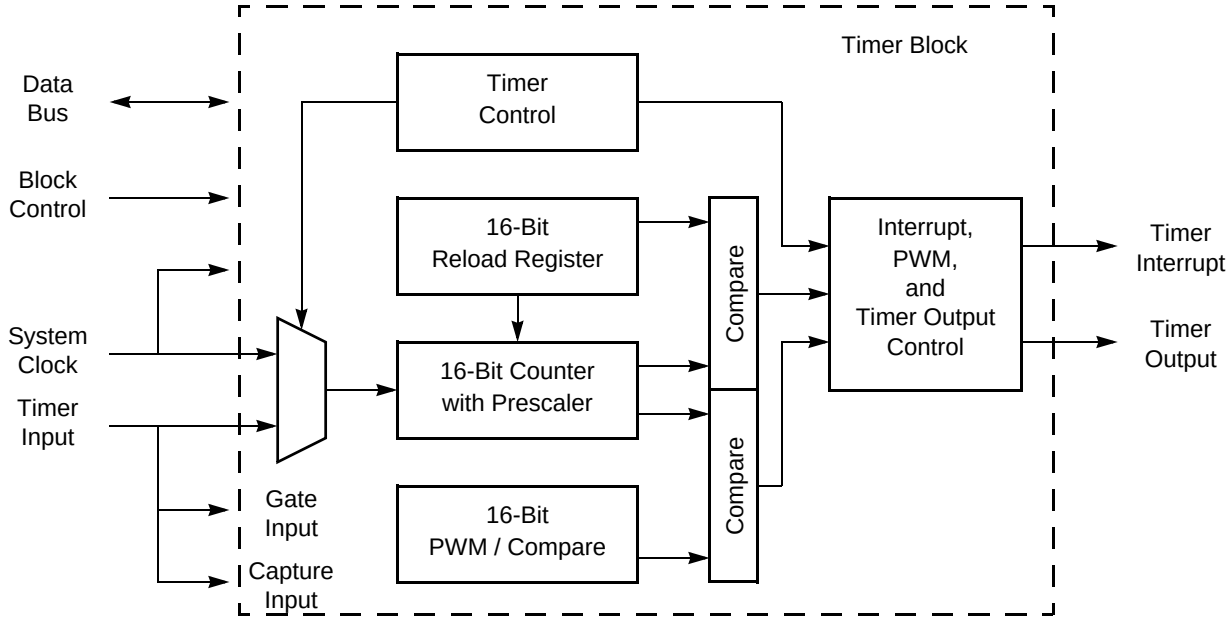


Figure 10. Timer Block Diagram

Timer Operating Modes

The timers are 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. On 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. Then, 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 vice-versa) on timer Reload. If it is required for the Timer Output to make a permanent state change on One-Shot time-out, first set the TPOL bit in the Timer Control Register to the start value before beginning ONE-SHOT mode. Then, after starting the timer, set TPOL to the opposite bit value.

Follow the steps below for configuring a timer for ONE-SHOT mode and initiating the count:

1. Write to the Timer Control Register to:
   – Disable the timer
   – Configure the timer for ONE-SHOT mode

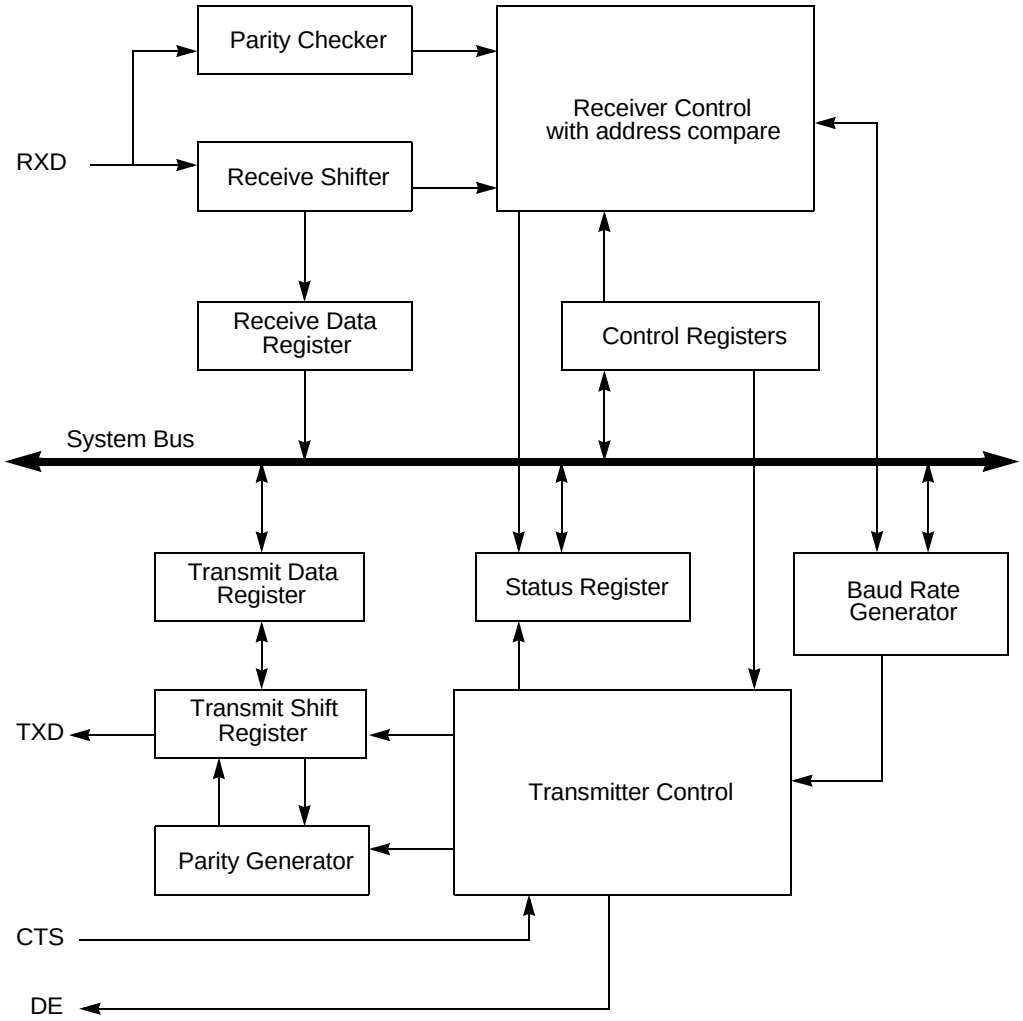


Figure 11. UART Block Diagram

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.

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.

During an SPI transfer, data is sent and received simultaneously by both the Master and the Slave SPI devices. Separate signals are required for data and the serial clock. When an SPI transfer occurs, a multi-bit (typically 8-bit) character is shifted out one data pin and an multi-bit character is simultaneously shifted in on a second data pin. An 8-bit shift register in the Master and another 8-bit shift register in the Slave are connected as a circular buffer. The SPI shift register is single-buffered in the transmit and receive directions. New data to be transmitted cannot be written into the shift register until the previous transmission is complete and receive data (if valid) has been read.

SPI Signals

The four basic SPI signals are:

- MISO (Master-In, Slave-Out)
- MOSI (Master-Out, Slave-In)
- SCK (Serial Clock)
- $\overline{SS}$ (Slave Select)

The following sections discuss these SPI signals. Each signal is described in both Master and Slave modes.

Master-In/Slave-Out

The Master-In/Slave-Out (MISO) pin is configured as an input in a Master device and as an output in a Slave device. It is one of the two lines that transfer serial data, with the most significant bit sent first. The MISO pin of a Slave device is placed in a high-impedance state if the Slave is not selected. When the SPI is not enabled, this signal is in a high-impedance state.

Master-Out/Slave-In

The Master-Out/Slave-In (MOSI) pin is configured as an output in a Master device and as an input in a Slave device. It is one of the two lines that transfer serial data, with the most significant bit sent first. When the SPI is not enabled, this signal is in a high-impedance state.

Serial Clock

The Serial Clock (SCK) synchronizes data movement both in and out of the device through its MOSI and MISO pins. In MASTER mode, the SPI's Baud Rate Generator creates the serial clock. The Master drives the serial clock out its own SCK pin to the Slave's SCK pin. When the SPI is configured as a Slave, the SCK pin is an input and the clock signal from the Master synchronizes the data transfer between the Master and Slave devices. Slave devices ignore the SCK signal, unless the $\overline{SS}$ pin is asserted. When configured as a slave, the SPI block requires a minimum SCK period of greater than or equal to 8 times the system (XIN) clock period.

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).

BRH = SPI Baud Rate High Byte
Most significant byte, BRG[15:8], of the SPI Baud Rate Generator's reload value.

Table 69. SPI Baud Rate Low Byte Register (SPIBRL)

BITS	7	6	5	4	3	2	1	0
FIELD	BRL							
RESET	1							
R/W	R/W							
ADDR	F67H							

BRL = SPI Baud Rate Low Byte
Least significant byte, BRG[7:0], of the SPI Baud Rate Generator's reload value.

ADCD_L—ADC Data Low Bits
These are the least significant two bits of the 10-bit ADC output. These bits are undefined after a Reset.

Reserved
These bits are reserved and are always undefined.

5. Re-write the page written in step 2 to the Page Select Register.
6. Write Flash Memory using LDC or LDCI instructions to program the Flash.
7. Repeat step 6 to program additional memory locations on the same page.
8. Write 00H to the Flash Control Register to lock the Flash Controller.

Page Erase

Flash memory can be erased one page (512 bytes) at a time. Page Erasing the Flash memory sets all bytes in that page to the value FFH. The Page Select Register identifies the page to be erased. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the Page Erase operation completes. Interrupts that occur when the Page Erase operation is in progress are serviced once the Page Erase operation is complete. When the Page Erase operation is complete, the Flash Controller returns to its locked state. Only pages located in unprotected sectors can be erased.

Follow the steps below to perform a Page Erase operation:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write the page to be erased to the Page Select Register.
3. Write the first unlock command 73H to the Flash Control Register.
4. Write the second unlock command 8CH to the Flash Control Register.
5. Re-write the page written in step 2 to the Page Select Register.
6. Write the Page Erase command 95H to the Flash Control Register.

Mass Erase

The Flash memory cannot be Mass Erased by user code.

Flash Controller Bypass

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Programming algorithms by controlling the Flash programming signals directly.

Flash Controller Bypass is recommended for gang programming applications and large volume customers who do not require in-circuit programming of the Flash memory.

For more information on bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore! XP*, available for download at www.zilog.com.

- **Write Program Memory (0AH)**—The Write Program Memory command writes data to Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Program Memory (0BH)**—The Read Program Memory command reads data from Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

- **(Flash version only) Write Data Memory (0CH)**—The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to 0). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Data Memory (0DH)**—The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG mode, this command returns FFH for the data.

```
DBG ← 0DH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
```

Figure 55 and Table 112 provide timing information for UART pins for the case where the Clear To Send input signal ($\overline{CTS}$) is not 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 $\overline{DE}$. $\overline{DE}$ asserts after the UART Transmit Data Register has been written. $\overline{DE}$ remains asserted for multiple characters as long as the Transmit Data Register is written with the next character before the current character has completed.

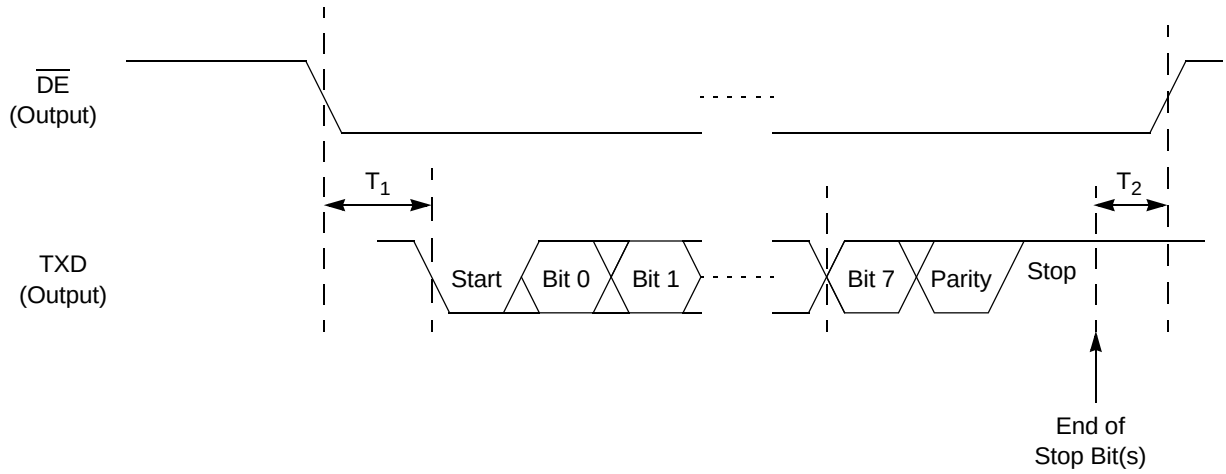


Figure 55. UART Timing without $\overline{CTS}$

Table 112. UART Timing without $\overline{CTS}$

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

Table 116 contains additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 116. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow. For example,

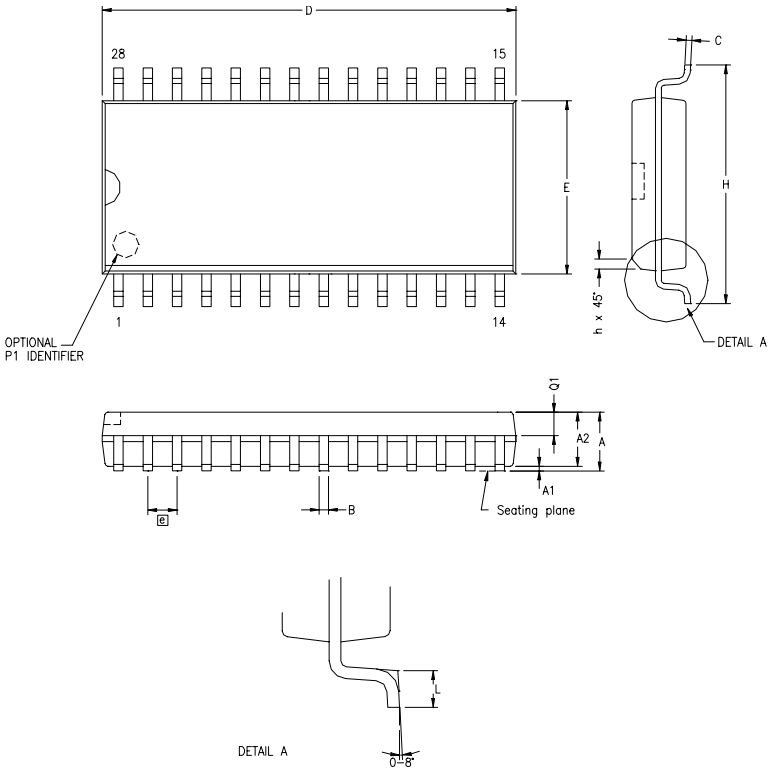
dst ← dst + src

indicates the source data is added to the destination data and the result is stored in the destination location.

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
ADD dst, src	dst ← dst + src	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	03							2	4
		R	R	04							3	3
		R	IR	05							3	4
		R	IM	06							3	3
		IR	IM	07							3	4
ADDX dst, src	dst ← dst + src	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							4	3
AND dst, src	dst ← dst AND src	r	r	52	-	*	*	0	-	-	2	3
		r	Ir	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	dst ← dst AND src	ER	ER	58	-	*	*	0	-	-	4	3
		ER	IM	59							4	3
BCLR bit, dst	dst[bit] ← 0	r		E2	-	-	-	-	-	-	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	-	-	-	-	-	-	2	2
BRK	Debugger Break			00	-	-	-	-	-	-	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	-	-	-	-	-	-	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	X	*	*	0	-	-	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	-	-	-	-	-	-	3	3
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X		r	F6	-	-	-	-	-	-	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	-	-	-	-	-	-	3	3
			Ir	F7							3	4

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)

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	I^2C	SPI	UARTs with IrDA	Description
Z8F08xx with 8 KB Flash										
Standard Temperature: 0 °C to 70 °C										
Z8F0811HH020SC	8 KB	1 KB	11	16	2	0	1	0	1	SSOP 20-pin package
Z8F0811PH020SC	8 KB	1 KB	11	16	2	0	1	0	1	PDIP 20-pin package
Z8F0812SJ020SC	8 KB	1 KB	19	19	2	0	1	1	1	SOIC 28-pin package
Z8F0812PJ020SC	8 KB	1 KB	19	19	2	0	1	1	1	PDIP 28-pin package
Extended Temperature: -40 °C to +105 °C										
Z8F0811HH020EC	8 KB	1 KB	11	16	2	0	1	0	1	SSOP 20-pin package
Z8F0811PH020EC	8 KB	1 KB	11	16	2	0	1	0	1	PDIP 20-pin package
Z8F0812SJ020EC	8 KB	1 KB	19	19	2	0	1	1	1	SOIC 28-pin package
Z8F0812PJ020EC	8 KB	1 KB	19	19	2	0	1	1	1	PDIP 28-pin package

S

T

Customer Support

For answers to technical questions about the product, documentation, or any other issues with Zilog's offerings, please visit Zilog's Knowledge Base at http://www.zilog.com/kb.

For any comments, detail technical questions, or reporting problems, please visit Zilog's Technical Support at http://support.zilog.com.