



Welcome to [E-XFL.COM](https://www.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 <sup>2</sup> C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/zilog/z8f1622vs020sg">https://www.e-xfl.com/product-detail/zilog/z8f1622vs020sg</a>

# **Table of Contents**

Revision History .....	iii
List of Figures .....	xi
List of Tables .....	xiii
Introduction .....	1
Features .....	1
Part Selection Guide .....	2
Block Diagram .....	3
CPU and Peripheral Overview .....	3
General-Purpose Input/Output .....	4
Flash Controller .....	4
10-Bit Analog-to-Digital Converter .....	4
UARTs .....	4
I <sup>2</sup> C .....	5
Serial Peripheral Interface .....	5
Timers .....	5
Interrupt Controller .....	5
Reset Controller .....	5
On-Chip Debugger .....	5
DMA Controller .....	6
Signal and Pin Descriptions .....	7
Available Packages .....	7
Pin Configurations .....	8
Signal Descriptions .....	14
Pin Characteristics .....	17
Address Space .....	18
Register File .....	18
Program Memory .....	19
Data Memory .....	20
Information Area .....	20
Register File Address Map .....	22
Reset and Stop Mode Recovery .....	28
Reset Types .....	28
Reset Sources .....	29
Power-On Reset .....	30
Voltage Brown-Out Reset .....	31

eZ8 CPU Instruction Set .....	225
Assembly Language Programming Introduction .....	225
Assembly Language Syntax .....	226
eZ8 CPU Instruction Notation .....	227
Condition Codes .....	229
eZ8 CPU Instruction Classes .....	230
eZ8 CPU Instruction Summary .....	234
Flags Register .....	243
Op Code Maps .....	244
Appendix A. Register Tables .....	248
General Purpose RAM .....	248
Timer 0 .....	248
Universal Asynchronous Receiver/Transmitter (UART) .....	256
Inter-Integrated Circuit (I <sup>2</sup> C) .....	261
Serial Peripheral Interface .....	263
Analog-to-Digital Converter (ADC) .....	266
Direct Memory Access (DMA) .....	266
Interrupt Request (IRQ) .....	270
General-Purpose Input/Output (GPIO) .....	274
Watchdog Timer .....	282
Flash .....	284
Packaging .....	286
Ordering Information .....	287
Part Number Suffix Designations .....	292
Index .....	293
Customer Support .....	303

Figure 31.	10-Bit Addressed Slave Data Transfer Format . . . . .	136
Figure 32.	Receive Data Transfer Format for a 7-Bit Addressed Slave . . . . .	138
Figure 33.	Receive Data Format for a 10-Bit Addressed Slave . . . . .	139
Figure 34.	Analog-to-Digital Converter Block Diagram . . . . .	162
Figure 35.	Flash Memory Arrangement . . . . .	170
Figure 36.	On-Chip Debugger Block Diagram . . . . .	183
Figure 37.	Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #1 of 2 . . . . .	184
Figure 38.	Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #2 of 2 . . . . .	185
Figure 39.	OCD Data Format . . . . .	186
Figure 40.	Recommended 20MHz Crystal Oscillator Configuration . . . . .	197
Figure 41.	Connecting the On-Chip Oscillator to an External RC Network . . . . .	198
Figure 42.	Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45kΩ Resistor . . . . .	199
Figure 43.	Typical Active Mode Idd vs. System Clock Frequency . . . . .	205
Figure 44.	Maximum Active Mode Idd vs. System Clock Frequency . . . . .	206
Figure 45.	Typical HALT Mode Idd vs. System Clock Frequency . . . . .	207
Figure 46.	Maximum HALT Mode Icc vs. System Clock Frequency . . . . .	208
Figure 47.	Maximum STOP Mode IDD with VBO Enabled vs. Power Supply Voltage . . . . .	209
Figure 48.	Maximum STOP Mode IDD with VBO Disabled vs. Power Supply Voltage . . . . .	210
Figure 49.	Analog-to-Digital Converter Frequency Response . . . . .	215
Figure 50.	Port Input Sample Timing . . . . .	217
Figure 51.	GPIO Port Output Timing . . . . .	218
Figure 52.	On-Chip Debugger Timing . . . . .	219
Figure 53.	SPI Master Mode Timing . . . . .	220
Figure 54.	SPI Slave Mode Timing . . . . .	221
Figure 55.	I <sup>2</sup> C Timing . . . . .	222
Figure 56.	UART Timing with CTS . . . . .	223
Figure 57.	UART Timing without CTS . . . . .	224
Figure 58.	Flags Register . . . . .	243
Figure 59.	Op Code Map Cell Description . . . . .	244
Figure 60.	First Op Code Map . . . . .	246
Figure 61.	Second Op Code Map after 1FH . . . . .	247

Table 34.	IRQ2 Enable High Bit Register (IRQ2ENH) .....	58
Table 35.	IRQ2 Enable Low Bit Register (IRQ2ENL) .....	59
Table 36.	Interrupt Edge Select Register (IRQES) .....	60
Table 37.	Interrupt Port Select Register (IRQPS) .....	60
Table 38.	Interrupt Control Register (IRQCTL) .....	61
Table 39.	Timer 0–3 High Byte Register (TxH) .....	73
Table 40.	Timer 0–3 Low Byte Register (TxL) .....	73
Table 41.	Timer 0–3 Reload High Byte Register (TxRH) .....	74
Table 42.	Timer 0–3 Reload Low Byte Register (TxRL) .....	74
Table 43.	Timer 0–3 PWM High Byte Register (TxPWMH) .....	75
Table 44.	Timer 0–3 PWM Low Byte Register (TxPWML) .....	75
Table 45.	Timer 0–3 Control 0 Register (TxCTL0) .....	76
Table 46.	Timer 0–3 Control 1 Register (TxCTL1) .....	77
Table 47.	Watchdog Timer Approximate Time-Out Delays .....	81
Table 48.	Watchdog Timer Control Register (WDTCTL) .....	84
Table 49.	Watchdog Timer Events .....	85
Table 50.	Watchdog Timer Reload Upper Byte Register (WDTU) .....	85
Table 51.	Watchdog Timer Reload High Byte Register (WDTH) .....	86
Table 52.	Watchdog Timer Reload Low Byte Register (WDTL) .....	86
Table 53.	UART Transmit Data Register (UxTXD) .....	99
Table 54.	UART Receive Data Register (UxRXD) .....	99
Table 55.	UART Status 0 Register (UxSTAT0) .....	100
Table 56.	UART Status 1 Register (UxSTAT1) .....	101
Table 57.	UART Control 0 Register (UxCTL0) .....	102
Table 58.	UART Control 1 Register (UxCTL1) .....	103
Table 59.	UART Address Compare Register (UxADDR) .....	105
Table 60.	UART Baud Rate High Byte Register (UxBRH) .....	106
Table 61.	UART Baud Rate Low Byte Register (UxBRL) .....	106
Table 62.	UART Baud Rates .....	107
Table 63.	SPI Clock Phase (PHASE) and Clock Polarity (CLKPOL) Operation ...	117
Table 64.	SPI Data Register (SPIDATA) .....	122
Table 65.	SPI Control Register (SPICTL) .....	122
Table 66.	SPI Status Register (SPISTAT) .....	123
Table 67.	SPI Mode Register (SPIMODE) .....	125
Table 68.	SPI Diagnostic State Register (SPIDST) .....	126
Table 69.	SPI Baud Rate High Byte Register (SPIBRH) .....	127

Table 106.	Absolute Maximum Ratings .....	200
Table 107.	DC Characteristics .....	202
Table 108.	Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing .....	211
Table 109.	Reset and Stop Mode Recovery Pin Timing .....	212
Table 110.	External RC Oscillator Electrical Characteristics and Timing .....	212
Table 111.	Flash Memory Electrical Characteristics and Timing .....	213
Table 112.	Watchdog Timer Electrical Characteristics and Timing .....	213
Table 113.	Analog-to-Digital Converter Electrical Characteristics and Timing .....	214
Table 114.	AC Characteristics .....	216
Table 115.	GPIO Port Input Timing .....	217
Table 116.	GPIO Port Output Timing .....	218
Table 117.	On-Chip Debugger Timing .....	219
Table 118.	SPI Master Mode Timing .....	220
Table 119.	SPI Slave Mode Timing .....	221
Table 120.	I <sup>2</sup> C Timing .....	222
Table 121.	UART Timing with CTS .....	223
Table 122.	UART Timing without CTS .....	224
Table 123.	Assembly Language Syntax Example 1 .....	226
Table 124.	Assembly Language Syntax Example 2 .....	227
Table 125.	Notational Shorthand .....	227
Table 126.	Additional Symbols .....	228
Table 127.	Condition Codes .....	229
Table 128.	Arithmetic Instructions .....	230
Table 129.	Bit Manipulation Instructions .....	231
Table 130.	Block Transfer Instructions .....	231
Table 131.	CPU Control Instructions .....	232
Table 132.	Load Instructions .....	232
Table 133.	Logical Instructions .....	233
Table 134.	Program Control Instructions .....	233
Table 135.	Rotate and Shift Instructions .....	234
Table 136.	eZ8 CPU Instruction Summary .....	234
Table 137.	Op Code Map Abbreviations .....	244
Table 138.	Timer 0–3 High Byte Register (TxH) .....	248
Table 139.	Timer 0–3 Low Byte Register (TxL) .....	249
Table 140.	Timer 0–3 Reload High Byte Register (TxRH) .....	249

## Voltage Brown-Out Reset

The devices in the Z8 Encore! XP F64xx Series provide low Voltage Brown-Out 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 remains below the Power-On Reset voltage threshold ( $V_{POR}$ ), the VBO block holds the device in the Reset state.

After the supply voltage again exceeds the Power-On Reset voltage threshold, the devices progress through a full system reset sequence, as described in the Power-On Reset section. Following Power-On Reset, the POR status bit in the Watchdog Timer Control (WDTCTL) Register is set to 1. Figure 9 displays Voltage Brown-Out operation. For the VBO and POR threshold voltages ( $V_{VBO}$  and  $V_{POR}$ ), see the [Electrical Characteristics](#) chapter on page 200.

The Voltage Brown-Out circuit can be either enabled or disabled during STOP Mode. Operation during STOP Mode is set by the VBO\_AO option bit. For information about configuring VBO\_AO, see the [Option Bits](#) chapter on page 180.

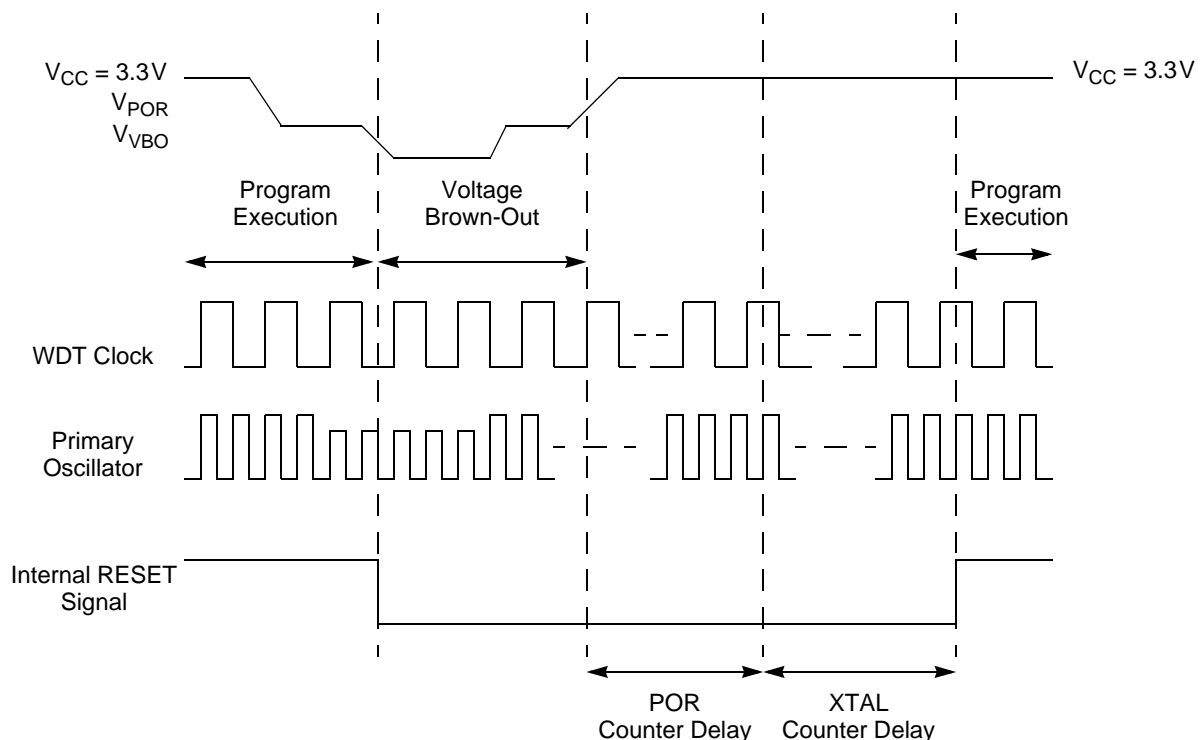


Figure 9. Voltage Brown-Out Reset Operation

## Timer 0–3 PWM High and Low Byte Registers

The Timer 0–3 PWM High and Low Byte (TxPWMH and TxPWML) registers, shown in Tables 43 and 44, are used for Pulse-Width Modulator (PWM) operations. These registers also store the capture values for the CAPTURE and CAPTURE/COMPARE modes.

**Table 43. Timer 0–3 PWM High Byte Register (TxPWMH)**

Bit	7	6	5	4	3	2	1	0
Field	PWMH							
RESET	0							
R/W	R/W							
Address	F04H, F0CH, F14H, F1CH							

**Table 44. Timer 0–3 PWM Low Byte Register (TxPWML)**

Bit	7	6	5	4	3	2	1	0
Field	PWML							
RESET	0							
R/W	R/W							
Address	F05H, F0DH, F15H, F1DH							

Bit	Description
[7:0]	<b>Pulse-Width Modulator High and Low Bytes</b>
PWMH, PWML	These two bytes, {PWMH[7:0], PWML[7:0]}, form a 16-bit value that is compared to the current 16-bit timer count. When a match occurs, the PWM output changes state. The PWM output value is set by the TPOL bit in the Timer Control 1 Register (TxCTL1) Register. The TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in CAPTURE or CAPTURE/COMPARE modes.



Table 47. Watchdog Timer Approximate Time-Out Delays

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

## Watchdog Timer Refresh

When first enabled, the Watchdog Timer is loaded with the value in the Watchdog Timer Reload registers. The Watchdog Timer then counts down to 000000H unless a WDT instruction is executed by the eZ8 CPU. Execution of the WDT instruction causes the downcounter to be reloaded with the WDT reload value stored in the Watchdog Timer Reload registers. Counting resumes following the reload operation.

When the Z8 Encore! XP F64xx Series devices are operating in DEBUG Mode (through the On-Chip Debugger), the Watchdog Timer is continuously refreshed to prevent spurious Watchdog Timer time-outs.

## Watchdog Timer Time-Out Response

The Watchdog Timer times out when the counter reaches 000000H. A time-out of the Watchdog Timer generates either an interrupt or a Reset. The WDT\_RES option bit determines the time-out response of the Watchdog Timer. For information about programming of the WDT\_RES option bit, see the [Option Bits](#) chapter on page 180.

### WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the Watchdog Timer issues an interrupt request to the interrupt controller and sets the WDT status bit in the Watchdog Timer Control Register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executing code from the vector address. After time-out and interrupt generation, the Watchdog Timer counter rolls over to its maximum value of FFFFFFFH and continues counting. The Watchdog Timer counter is not automatically returned to its reload value.

### WDT Interrupt in STOP Mode

If configured to generate an interrupt when a time-out occurs and the Z8 Encore! XP F64xx Series devices are in STOP Mode, the Watchdog Timer automatically initiates a Stop Mode Recovery and generates an interrupt request. Both the WDT status bit and the stop bit in the Watchdog Timer Control Register are set to 1 following WDT time-out in

Bit	Description (Continued)
[1] STOP	<b>Stop Bit Select</b> 0 = The transmitter sends one stop bit. 1 = The transmitter sends two stop bits.
[0] LBEN	<b>Loop Back Enable</b> 0 = Normal operation. 1 = All transmitted data is looped back to the receiver.

Table 58. UART Control 1 Register (UxCTL1)

Bit	7	6	5	4	3	2	1	0
Field	MPMD[1]	MPEN	MPMD[0]	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
RESET	0							
R/W	R/W							
Address	F43H and F4BH							

Bit	Description
[7,5] MPMD[1,0]	<b>MULTIPROCESSOR Mode</b> If MULTIPROCESSOR (9-Bit) Mode is enabled, 00 = The UART generates an interrupt request on all received bytes (data and address). 01 = The UART generates an interrupt request only on received address bytes. 10 = The UART generates an interrupt request when a received address byte matches the value stored in the Address Compare Register and on all successive data bytes until an address mismatch occurs. 11 = The UART generates an interrupt request on all received data bytes for which the most recent address byte matched the value in the Address Compare Register.
[6] MPEN	<b>MULTIPROCESSOR (9-bit) Enable</b> This bit is used to enable MULTIPROCESSOR (9-Bit) Mode. 0 = Disable MULTIPROCESSOR (9-Bit) Mode. 1 = Enable MULTIPROCESSOR (9-Bit) Mode.
[4] MPBT	<b>MULTIPROCESSOR Bit Transmit</b> This bit is applicable only when MULTIPROCESSOR (9-Bit) Mode is enabled. 0 = Send a 0 in the multiprocessor bit location of the data stream (9th bit). 1 = Send a 1 in the multiprocessor bit location of the data stream (9th bit).
[3] DEPOL	<b>Driver Enable Polarity</b> 0 = DE signal is Active High. 1 = DE signal is Active Low.

The next time  $\overline{SS}$  asserts, the MISO pin outputs SPIDAT[7], regardless of where the previous transaction left off. Writing a 1 to ABT clears this error flag.

## **SPI Interrupts**

When SPI interrupts are enabled, the SPI generates an interrupt after character transmission/reception completes in both MASTER and SLAVE modes. A character can be defined to be 1 through 8 bits by the NUMBITS field in the SPI Mode Register. In slave mode it is not necessary for  $\overline{SS}$  to deassert between characters to generate the interrupt. The SPI in Slave mode can also generate an interrupt if the  $\overline{SS}$  signal deasserts prior to transfer of all the bits in a character (see description of slave abort error above). 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 Interrupt Service Routine to generate future interrupts. To start the transfer process, an SPI interrupt may 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 Baud Rate Generator time-out. This timer function must be enabled by setting the BIRQ bit in the SPICTL Register. This Baud Rate Generator 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 Baud Rate Generator creates a lower frequency serial clock (SCK) for data transmission synchronization between the Master and the external Slave. The input to the Baud Rate Generator 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:

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

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

When the SPI is disabled, the Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. Observe the following procedure to configure the Baud Rate Generator 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 appropriate 16-bit count value into the SPI Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BIRQ bit in the SPI Control Register to 1.

## Start and Stop Conditions

The master (I<sup>2</sup>C) drives all Start and Stop signals and initiates all transactions. To start a transaction, the I<sup>2</sup>C Controller generates a start condition by pulling the SDA signal Low while SCL is High. To complete a transaction, the I<sup>2</sup>C Controller generates a Stop condition by creating a low-to-High transition of the SDA signal while the SCL signal is High. The start and stop bits in the I<sup>2</sup>C Control Register control the sending of the Start and Stop conditions. A master is also allowed to end one transaction and begin a new one by issuing a Restart. This is accomplished by setting the start bit at the end of a transaction, rather than the stop bit. Note that the Start condition not sent until the start bit is set and data has been written to the I<sup>2</sup>C Data Register.

## Master Write and Read Transactions

The following sections provide a recommended procedure for performing I<sup>2</sup>C write and read transactions from the I<sup>2</sup>C Controller (master) to slave I<sup>2</sup>C devices. In general software should rely on the TDRE, RDRF and NCKI bits of the status register (these bits generate interrupts) to initiate software actions. When using interrupts or DMA, the TXI bit is set to start each transaction and cleared at the end of each transaction to eliminate a *trailing* transmit interrupt.

Caution should be used in using the ACK status bit within a transaction because it is difficult for software to tell when it is updated by hardware.

When writing data to a slave, the I<sup>2</sup>C pauses at the beginning of the Acknowledge cycle if the data register has not been written with the next value to be sent (TDRE bit in the I<sup>2</sup>C Status Register = 1). In this scenario where software is not keeping up with the I<sup>2</sup>C bus (TDRE asserted longer than one byte time), the Acknowledge clock cycle for byte *n* is delayed until the Data Register is written with byte *n* + 1, and appears to be grouped with the data clock cycles for byte *n*+1. If either the start or stop bit is set, the I<sup>2</sup>C does not pause prior to the Acknowledge cycle because no additional data is sent.

When a Not Acknowledge condition is received during a write (either during the address or data phases), the I<sup>2</sup>C Controller generates the Not Acknowledge interrupt (NCKI = 1) and pause until either the stop or start bit is set. Unless the Not Acknowledge was received on the last byte, the Data Register will already have been written with the next address or data byte to send. In this case the flush bit of the Control Register should be set at the same time the stop or start bit is set to remove the stale transmit data and enable subsequent transmit interrupts.

When reading data from the slave, the I<sup>2</sup>C pauses after the data Acknowledge cycle until the receive interrupt is serviced and the RDRF bit of the status register is cleared by reading the I<sup>2</sup>C Data Register. Once the I<sup>2</sup>C data register has been read, the I<sup>2</sup>C reads the next data byte.

The write-only Flash Control Register shares its Register File address with the read-only Flash Status Register.

**Table 93. Flash Control Register (FCTL)**

Bit	7	6	5	4	3	2	1	0
Field	FCMD							
RESET	0							
R/W	W							
Address	FF8H							

Bit	Description
[7:0]	<b>Flash Command*</b>
FCMD	73H = First unlock command. 8CH = Second unlock command. 95H = Page erase command. 63H = Mass erase command 5EH = Flash Sector Protect Register select.

Note: \*All other commands, or any command out of sequence, lock the Flash Controller.

Bit	Description (Continued)
[4] BRKLOOP	<b>Breakpoint Loop</b> 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 entered 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.
[3:1]	<b>Reserved</b> These bits are reserved and must be programmed to 000.
[0] RST	<b>Reset</b> Setting this bit to 1 resets the Z8 Encore! XP F64xx Series devices. The devices go through a normal Power-On Reset sequence with the exception that the On-Chip Debugger is not reset. This bit is automatically cleared to 0 when the reset finishes. 0 = No effect. 1 = Reset the Z8 Encore! XP F64xx Series device.

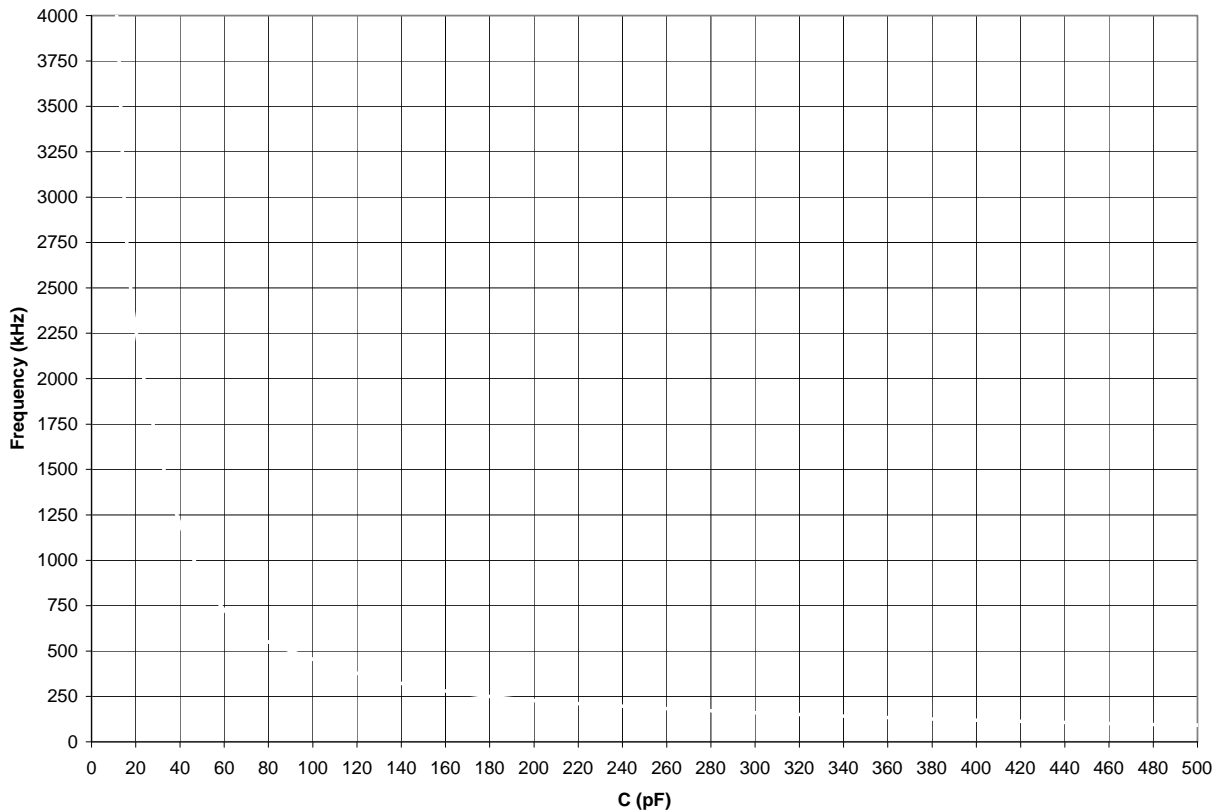
## OCD Status Register

The OCD Status Register, shown in Table 104, reports status information about the current state of the debugger and the system.

**Table 104. OCD Status Register (OCDSTAT)**

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

Bit	Description
[7] IDLE	<b>CPU Idle</b> 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.
[6] HALT	<b>HALT Mode</b> 0 = The device is not in HALT Mode. 1 = The device is in HALT Mode.



**Figure 42. Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45kΩ Resistor**

---

**! Caution:** When using the external RC oscillator mode, the oscillator may stop oscillating if the power supply drops below 2.7 V, but before the power supply drops to the voltage brown-out threshold. The oscillator will resume oscillation as soon as the supply voltage exceeds 2.7 V.

---

Figure 45 displays the typical current consumption in HALT Mode while operating at 25°C plotted opposite the system clock frequency. All GPIO pins are configured as outputs and driven High.

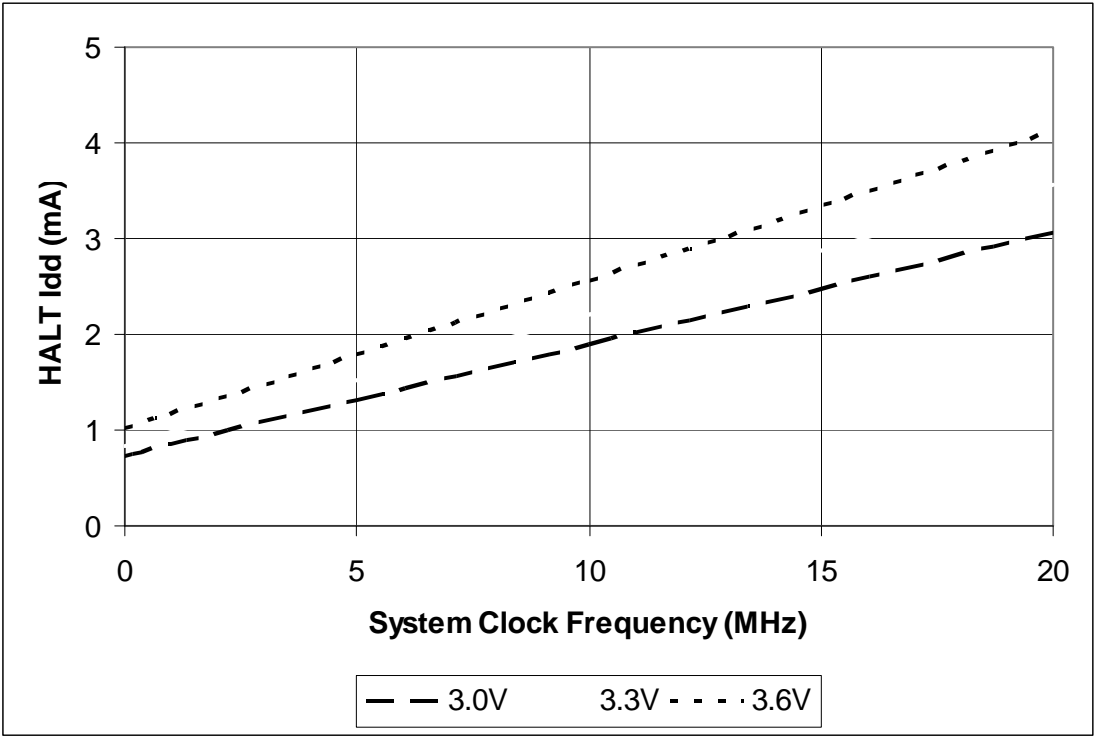


Figure 45. Typical HALT Mode I<sub>DD</sub> vs. System Clock Frequency



## UART Timing

Figure 56 and Table 121 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 ( $T_1$ ) assumes the UART Transmit Data Register has been loaded with data prior to  $\overline{\text{CTS}}$  assertion.

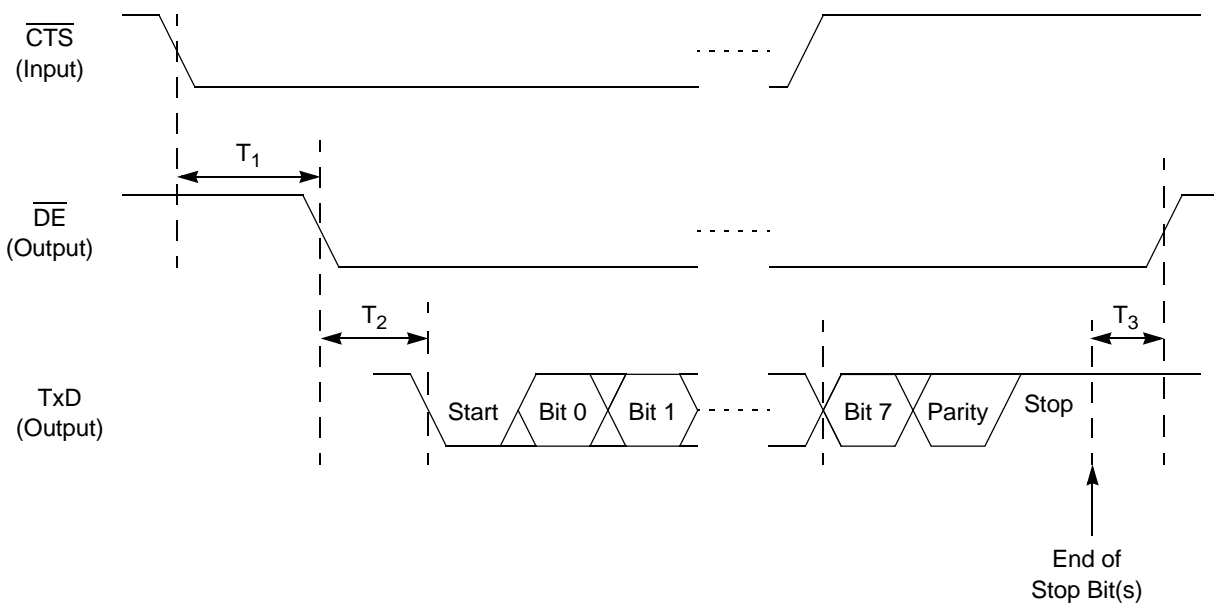


Figure 56. UART Timing with  $\overline{\text{CTS}}$

Table 121. UART Timing with  $\overline{\text{CTS}}$

Parameter	Abbreviation	Delay (ns)	
		Minimum	Maximum
$T_1$	$\overline{\text{CTS}}$ Fall to $\overline{\text{DE}}$ Assertion Delay	$2 * X_{\text{IN}}$ period	$2 * X_{\text{IN}}$ period + 1 bit period
$T_2$	$\overline{\text{DE}}$ Assertion to TxD Falling Edge (Start) Delay	1 bit period	1 bit period + $1 * X_{\text{IN}}$ period
$T_3$	End of stop bit(s) to $\overline{\text{DE}}$ Deassertion Delay	$1 * X_{\text{IN}}$ period	$2 * X_{\text{IN}}$ period

**Hex Address: F55****Table 193. I<sup>2</sup>C Diagnostic State Register (I2CDST)**

Bit	7	6	5	4	3	2	1	0
Field	SCLIN	SDAIN	STPCNT	TXRXSTATE				
RESET	X		0					
R/W	R							
Address	F55H							

**Hex Address: F56****Table 194. I<sup>2</sup>C Diagnostic Control Register (I2CDIAG)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved							DIAG
RESET	0							
R/W	R							R/W
Address	F56H							

**Hex Addresses: F57–F5F**

This address range is reserved.

## Serial Peripheral Interface

For more information about these SPI Control registers, see the [SPI Control Register Definitions](#) section on page 121.

**Hex Address: F60****Table 195. SPI Data Register (SPIDATA)**

Bit	7	6	5	4	3	2	1	0
Field	DATA							
RESET	X							
R/W	R/W							
Address	F60H							

## Analog-to-Digital Converter (ADC)

For more information about these ADC Control registers, see the [ADC Control Register Definitions](#) section on page 165.

### Hex Addresses: F70–F71

This address range is reserved.

### Hex Address: F72

Table 202. ADC Data High Byte Register (ADCD\_H)

Bit	7	6	5	4	3	2	1	0
Field	ADCD_H							
RESET	X							
R/W	R							
Address	F72H							

### Hex Address: F73

Table 203. ADC Data Low Bits Register (ADCD\_L)

Bit	7	6	5	4	3	2	1	0
Field	ADCD_L		Reserved					
RESET	X							
R/W	R							
Address	F73H							

### Hex Addresses: F74–FAF

This address range is reserved.

## Direct Memory Access (DMA)

For more information about these DMA Control registers, see the [DMA Control Register Definitions](#) section on page 152.

## General-Purpose Input/Output (GPIO)

For more information about these GPIO Control registers, see the [GPIO Control Register Definitions](#) section on page 39.

### Hex Address: FD0

**Table 229. Port A–H GPIO Address Registers (PxADDR)**

Bit	7	6	5	4	3	2	1	0
Field	PADDR[7:0]							
RESET	00H							
R/W	R/W							
Address	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

### Hex Address: FD1

**Table 230. Port A–H Control Registers (PxCTL)**

Bit	7	6	5	4	3	2	1	0
Field	PCTL							
RESET	00H							
R/W	R/W							
Address	FD1H, FD5H, FD9H, FDDH, FE1H, FE5H, FE9H, FEDH							

### Hex Address: FD2

**Table 231. Port A–H Input Data Registers (PxIN)**

Bit	7	6	5	4	3	2	1	0
Field	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
RESET	X							
R/W	R							
Address	FD2H, FD6H, FDAH, FDEH, FE2H, FE6H, FEAH, FEEH							

- SPI status (SPISTAT) 124, 265
- status, I2C 143
- status, SPI 124
- UARTx baud rate high byte (UxBRH) 107, 259, 262
- UARTx baud rate low byte (UxBRL) 107, 260, 262
- UARTx Control 0 (UxCTL0) 103, 106, 258, 259, 261
- UARTx control 1 (UxCTL1) 104, 259, 261
- UARTx receive data (UxRXD) 100, 258, 260
- UARTx status 0 (UxSTAT0) 101, 258, 260
- UARTx status 1 (UxSTAT1) 102, 259, 261
- UARTx transmit data (UxTXD) 100, 258, 260
- watchdog timer control (WDTCTL) 85, 283
- watchdog timer reload high byte (WDTH) 87, 284
- watchdog timer reload low byte (WDTL) 87, 284
- watchdog timer reload upper byte (WDTU) 86, 283
- register file 19
- register file address map 23
- register pair 229
- register pointer 229
- reset
  - and STOP mode characteristics 29
  - carry flag 232
  - controller 6
  - sources 30
- RET 234
- return 234
- RL 235
- RLC 235
- rotate and shift instructions 235
- rotate left 235
- rotate left through carry 235
- rotate right 235
- rotate right through carry 235
- RP 229
- RR 229, 235
- rr 229
- RRC 235

## **S**

- SBC 232
- SCF 232, 233
- SDA and SCL (IrDA) signals 131
- second opcode map after 1FH 248
- serial clock 117
- serial peripheral interface (SPI) 114
- set carry flag 232, 233
- set register pointer 233
- shift right arithmetic 235
- shift right logical 235
- signal descriptions 15
- single-shot conversion (ADC) 164
- SIO 6
- slave data transfer formats (I2C) 137
- slave select 117
- software trap 234
- source operand 229
- SP 229
- SPI
  - architecture 114
  - baud rate generator 121
  - baud rate high and low byte register 127
  - clock phase 117
  - configured as slave 115
  - control register 123
  - control register definitions 122
  - data register 122
  - error detection 120
  - interrupts 121
  - mode fault error 120
  - mode register 126
  - multi-master operation 119
  - operation 116
  - overflow error 120
  - signals 116
  - single master, multiple slave system 115
  - single master, single slave system 114
  - status register 124
  - timing, PHASE = 0 118
  - timing, PHASE=1 119
- SPI controller signals 15
- SPI mode (SPIMODE) 126, 265
- SPIBRH register 128, 266