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, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	32KB (32K 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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f3222vs020eg

Revision History

Each instance in the Revision History table reflects a change to this document from its previous revision. For more details, refer to the corresponding pages or appropriate links listed in the table below.

Date	Revision Level	Description	Page
Jan 2013	24	Restored 40-pin PDIP package to Signal and Pin Descriptions and Packaging chapters.	7, 286
Feb 2012	23	Corrected formatting of I_{DDS} section, Table 107; corrected language in the General Purpose RAM section of Appendix A;	202, 248
Sep 2011	22	Revised Flash Sector Protect Register description; revised Packaging chapter.	178, 286
Mar 2008	21	Changed title to Z8 Encore! XP F64xx Series.	All
Feb 2008	20	Changed Z8 Encore! XP 64K Series Flash Microcontrollers to Z8 Encore! XP F64xx Series Flash Microcontrollers. Deleted three sentences that mentioned Z8R642. Removed the 40 PDIP package. Added ZENETSC0100ZACG to the end of the Ordering Information table. Changed the flag status to unaffected for BIT, BSET, and BCLR in the eZ8 CPU Instruction Summary table.	287, 234
Dec 2007	19	Updated Zilog logo, Disclaimer section, and implemented style guide. Updated Table 113. Changed Z8 Encore! 64K Series to Z8 Encore! XP 64K Series Flash Microcontrollers throughout the document.	All
Dec 2006	18	Updated Flash Memory Electrical Characteristics and Timing table and Ordering Information chapter.	213, 287
Nov 2006	17	Updated Part Number Suffix Designations section.	292

Table of Contents

Block Diagram

Figure 1 displays the architecture of the Z8 Encore! XP F64xx Series.

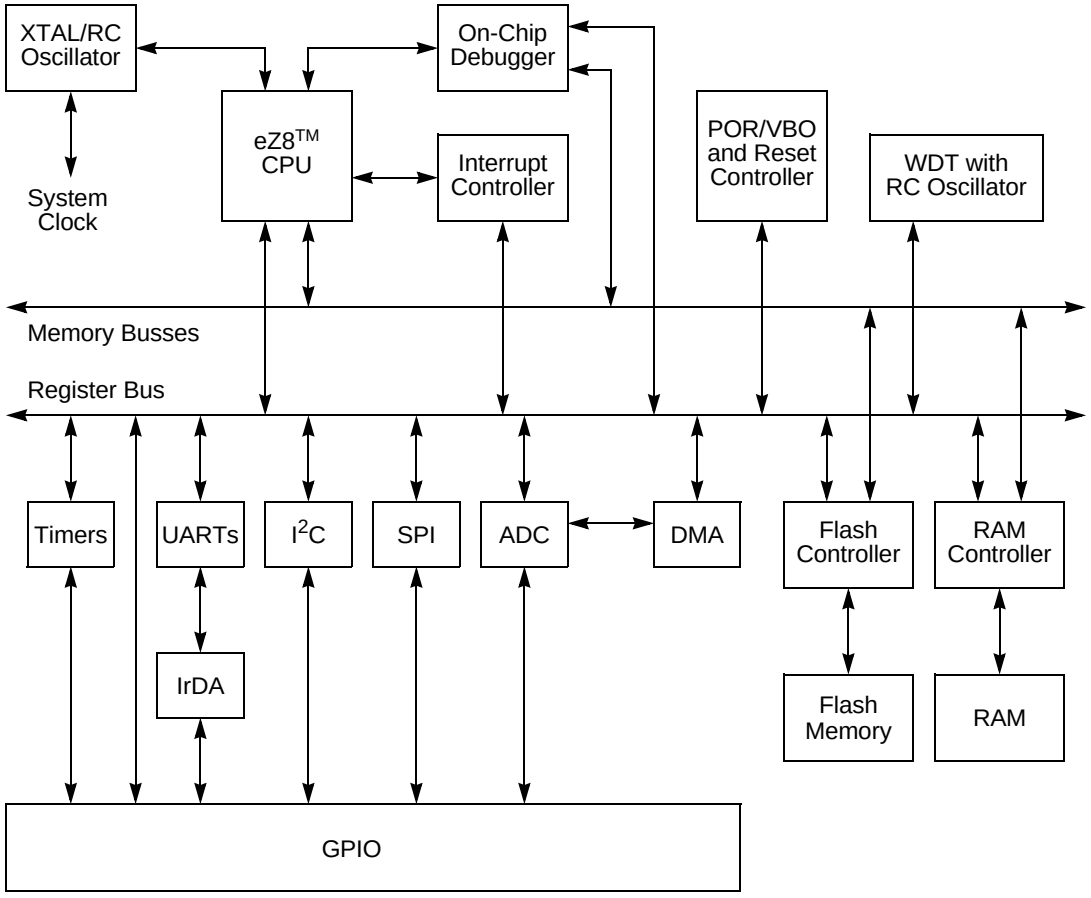

Figure 1. Z8 Encore! XP F64xx Series Block Diagram

CPU and Peripheral Overview

The latest 8-bit eZ8 CPU meets the continuing demand for faster and more code-efficient microcontrollers. The eZ8 CPU executes a superset of the original Z8 instruction set.

eZ8 CPU features include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required program memory

If TPOL is set to 0, the ratio of the PWM output High time to the total period is calculated using the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is calculated using the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

CAPTURE Mode

In CAPTURE Mode, the current timer count value is recorded when the appropriate external timer input transition occurs. The Capture count value is written to the Timer PWM High and Low Byte Registers. The timer input is the system clock. The TPOL bit in the Timer Control 1 Register determines if the Capture occurs on a rising edge or a falling edge of the timer input signal. When the capture event occurs, an interrupt is generated and the timer continues counting.

The timer continues counting up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt and continues counting.

Observe the following procedure for configuring a timer for CAPTURE Mode and initiating the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for CAPTURE Mode
   - Set the prescale value
   - Set the Capture edge (rising or falling) for the timer input
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Clear the Timer PWM High and Low Byte registers to `0000H`. This allows the software to determine if interrupts were generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain `0000H` after the interrupt, then the interrupt was generated by a reload.
5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.

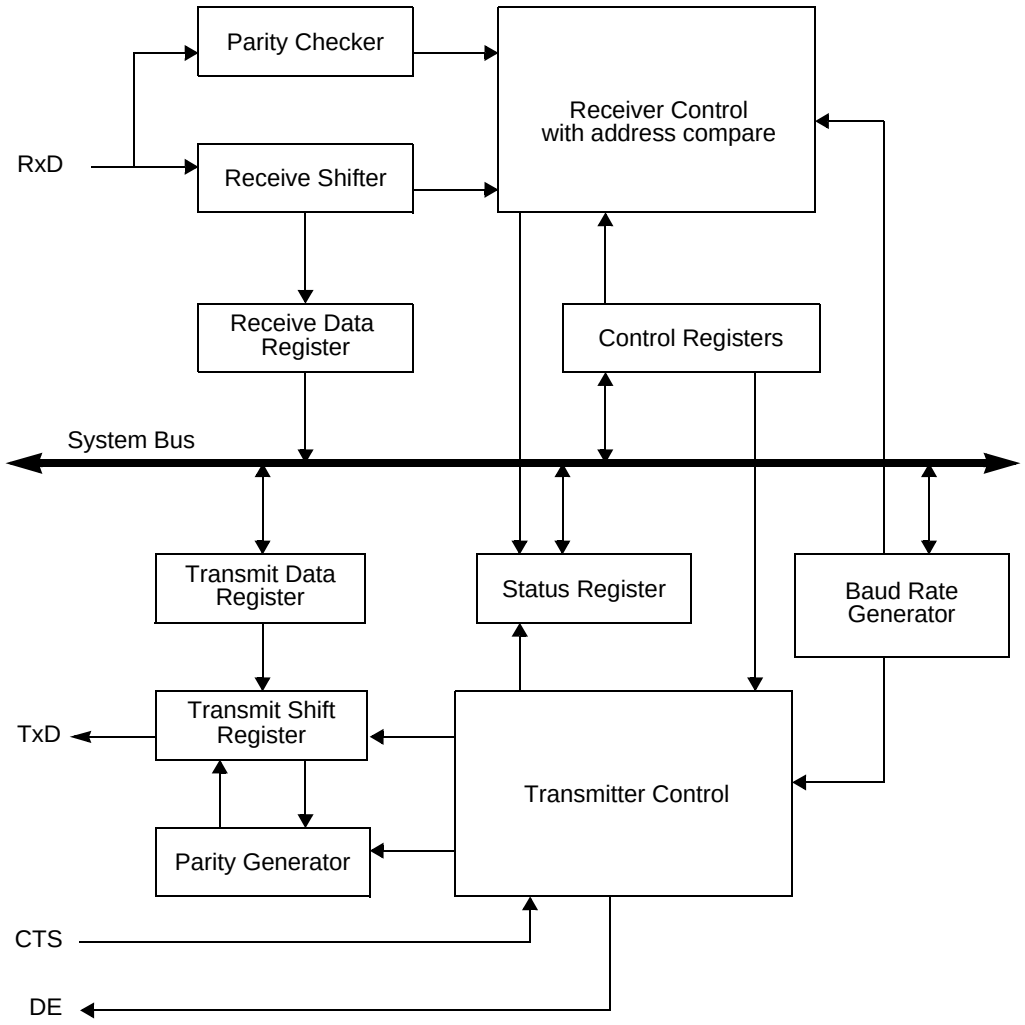

Figure 13. UART Block Diagram

Operation

The UART always transmits and receives data in an 8-bit data format, least significant bit first. An even or odd parity bit can be 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. Figures 14 and 15 display the asynchronous data format employed by the UART without parity and with parity, respectively.

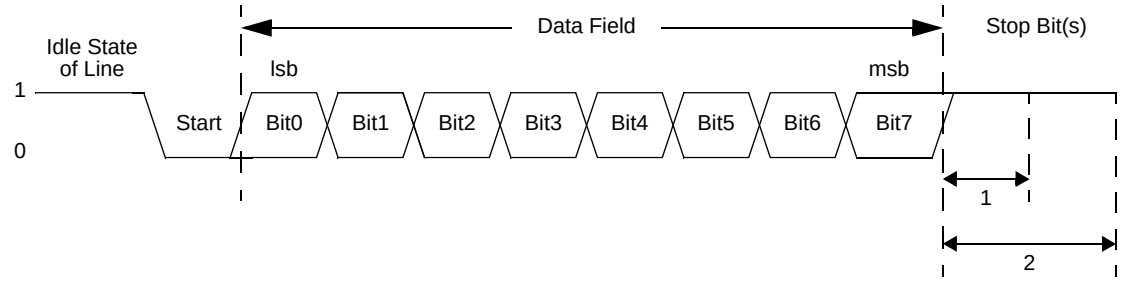

Figure 14. UART Asynchronous Data Format without Parity

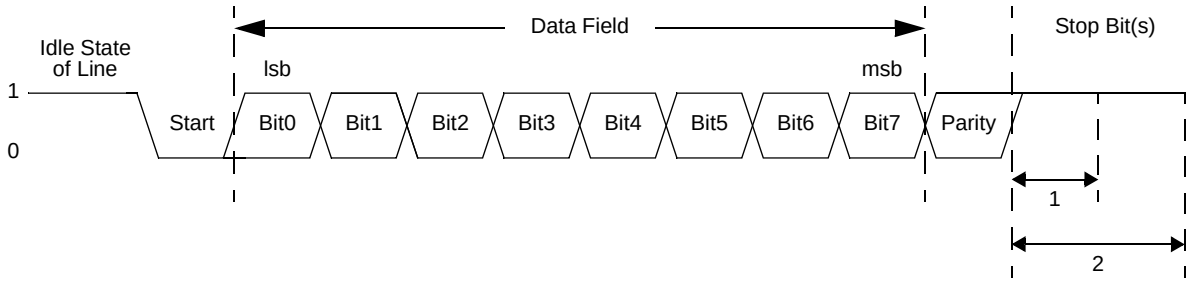

Figure 15. UART Asynchronous Data Format with Parity

Transmitting Data using the Polled Method

Observe the following procedure to transmit data using the polled method of operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. If MULTIPROCESSOR Mode is appropriate, 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 appropriate and MULTIPROCESSOR Mode is not enabled, set the parity enable bit (PEN) and select either Even or Odd parity (PSEL)

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 (X_{IN}) clock period.

The Master and Slave are each capable of exchanging a character of data during a sequence of NUMBITS clock cycles (see the NUMBITS field in the SPI Mode Register section on page 125). In both Master and Slave SPI devices, data is shifted on one edge of the SCK and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI phase and polarity control.

Slave Select

The active Low Slave Select ($\overline{SS}$) input signal selects a Slave SPI device. $\overline{SS}$ must be Low prior to all data communication to and from the Slave device. $\overline{SS}$ must stay Low for the full duration of each character transferred. The $\overline{SS}$ signal may stay Low during the transfer of multiple characters or may deassert between each character.

When the SPI is configured as the only Master in an SPI system, the $\overline{SS}$ pin can be set as either an input or an output. Other GPIO output pins can also be employed to select external SPI Slave devices.

When the SPI is configured as one Master in a multimaster SPI system, the $\overline{SS}$ pin must be set as an input. The $\overline{SS}$ input signal on the Master must be High. If the $\overline{SS}$ signal goes Low (indicating another Master is driving the SPI bus), a collision error flag is set in the SPI Status Register.

SPI Clock Phase and Polarity Control

The SPI supports four combinations of serial clock phase and polarity using two bits in the SPI Control Register. The clock polarity bit, CLKPOL, selects an active high or active Low clock and has no effect on the transfer format. Table 63 lists the SPI Clock Phase and Polarity Operation parameters. The clock phase bit, PHASE, selects one of two fundamentally different transfer formats. For proper data transmission, the clock phase and polarity must be identical for the SPI Master and the SPI Slave. The Master always places data on the MOSI line a half-cycle before the receive clock edge (SCK signal), in order for the Slave to latch the data.

I²C Controller

The I²C Controller makes the Z8 Encore! XP F64xx Series products bus-compatible with the I²C protocol. The I²C Controller consists of two bidirectional bus lines: a serial data signal (SDA) and a serial clock signal (SCL). Features of the I²C Controller include:

- Transmit and Receive Operation in MASTER Mode
- Maximum data rate of 400 kilobit/sec
- 7- and 10-bit addressing modes for Slaves
- Unrestricted number of data bytes transmitted per transfer

The I²C Controller in the Z8 Encore! XP F64xx Series products does not operate in SLAVE Mode.

Architecture

Figure 27 displays the architecture of the I²C Controller.

- The first bit of the byte of an address is shifting out and the RD bit of the I²C Status Register is deasserted.
- The first bit of a 10-bit address shifts out
- The first bit of write data shifts out

Note: Writing to the I²C Data Register always clears the TRDE bit to 0. When TDRE is asserted, the I²C Controller pauses at the beginning of the Acknowledge cycle of the byte currently shifting out. It does not resume until the Data Register is written with the next value to send or until the stop or start bits are set, indicating that the current byte is the last one to send.

The fourth interrupt source is the baud rate generator. If the I²C Controller is disabled (IEN bit in the I2CCTL Register = 0) and the BIRQ bit in the I2CCTL Register = 1, an interrupt is generated when the baud rate generator counts down to 1. This allows the I²C baud rate generator to be used by software as a general purpose timer when IEN = 0.

Software Control of I²C Transactions

Software can control I²C transactions by using the I²C Controller interrupt, by polling the I²C Status Register or by DMA. Note that not all products include a DMA Controller.

To use interrupts, the I²C interrupt must be enabled in the Interrupt Controller. The TXI bit in the I²C Control Register must be set to enable transmit interrupts.

To control transactions by polling, the interrupt bits (TDRE, RDRF and NCKI) in the I²C Status Register should be polled. The TDRE bit asserts regardless of the state of the TXI bit.

Either or both transmit and receive data movement can be controlled by the DMA Controller. The DMA Controller channel(s) must be initialized to select the I²C transmit and receive requests. Transmit DMA requests require that the TXI bit in the I²C Control Register be set.

Caution: A transmit (write) DMA operation hangs if the slave responds with a Not Acknowledge before the last byte has been sent. After receiving the Not Acknowledge, the I²C Controller sets the NCKI bit in the Status Register and pauses until either the stop or start bits in the Control Register are set.

For a receive (read) DMA transaction to send a Not Acknowledge on the last byte, the receive DMA must be set up to receive n-1 bytes, then software must set the NAK bit and receive the last (nth) byte directly.

10. The I2C Controller shifts the rest of the first byte of address and write bit out the SDA signal.
11. If the I2C slave sends an acknowledge by pulling the SDA signal Low during the next High period of SCL the I2C Controller sets the ACK bit in the I2C Status Register. Continue with Step 12.

    If the slave does not acknowledge the first address byte, the I2C Controller sets the NCKI bit and clears the ACK bit in the I2C Status Register. Software responds to the Not Acknowledge interrupt by setting the stop and flush bits 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 following steps).
12. The I2C Controller loads the I2C Shift Register with the contents of the I2C Data Register (2nd byte of address).
13. The I2C Controller shifts the second address byte out the SDA signal. After the first bit has been sent, the transmit interrupt is asserted.
14. Software responds by setting the stop bit in the I2C Control Register. The TXI bit can be cleared at the same time.
15. Software polls the stop bit of the I2C Control Register. Hardware deasserts the stop bit when the transaction is completed (stop condition has been sent).
16. Software checks the ACK bit of the I2C Status Register. If the slave acknowledged, the ACK bit is = 1. If the slave does not acknowledge, the ACK bit is = 0. The NCKI interrupt do not occur because the stop bit was set.

Write Transaction with a 10-Bit Address

Figure 31 displays the data transfer format for a 10-bit addressed slave. Shaded regions indicate data transferred from the I2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I2C Controller.

S	Slave Address 1st 7 bits	W = 0	A	Slave Address 2nd Byte	A	Data	A	Data	A/A	P/S

Figure 31. 10-Bit Addressed Slave Data Transfer Format

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 read/write control bit (=0). The transmit operation is carried out in the same manner as 7-bit addressing.

Observe the following procedure for a transmit operation on a 10-bit addressed slave:

1. Software asserts the IEN bit in the I2C Control Register.

Operation

This section describes the operational aspects of the ADC's power-down and conversion features.

Automatic Power-Down

If the ADC is idle (i.e., no conversions are in progress) for 160 consecutive system clock cycles, portions of the ADC are automatically powered down. From this powered-down state, the ADC requires 40 system clock cycles to power up. The ADC powers up when a conversion is requested using the ADC Control Register.

Single-Shot Conversion

When configured for single-shot conversion, the ADC performs a single analog-to-digital conversion on the selected analog input channel. After completion of the conversion, the ADC shuts down. Observe the following procedure for setting up the ADC and initiating a single-shot conversion:

1. Enable the appropriate analog inputs by configuring the general-purpose I/O pins for alternate function. This configuration disables the digital input and output drivers.
2. Write to the ADC Control Register to configure the ADC and begin the conversion. The bit fields in the ADC Control Register can be written simultaneously:
   - Write to the ANAIN[3:0] field to select one of the 12 analog input sources
   - Clear CONT to 0 to select a single-shot conversion
   - Write to the $\overline{\text{VREF}}$ bit to enable or disable the internal voltage reference generator
   - Set CEN to 1 to start the conversion
3. CEN remains 1 while the conversion is in progress. A single-shot conversion requires 5129 system clock cycles to complete. If a single-shot conversion is requested from an ADC powered-down state, the ADC uses 40 additional clock cycles to power up before beginning the 5129 cycle conversion.
4. When the conversion is complete, the ADC control logic performs the following operations:
   - 10-bit data result written to {ADCD_H[7:0], ADCD_L[7:6]}
   - CEN resets to 0 to indicate the conversion is complete
   - An interrupt request is sent to the Interrupt Controller
5. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered down.

ADC Data High Byte Register

The ADC Data High Byte Register, shown in Table 88, contains the upper eight bits of the 10-bit ADC output. During a single-shot conversion, this value is invalid. Access to the ADC Data High Byte Register is read-only. The full 10-bit ADC result is provided by {ADCD_H[7:0], ADCD_L[7:6]}. Reading the ADC Data High Byte Register latches data in the ADC Low Bits Register.

Table 88. 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							

Bit	Description
[7:0] ADCD_H	**ADC Data High Byte** This byte contains the upper eight bits of the 10-bit ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the last conversion output is held in this register. These bits are undefined after a Reset.

Option Bits

Option bits allow user configuration of certain aspects of the Z8 Encore! XP F64xx Series operation. The feature configuration data is stored in the Flash memory and read during Reset. The features available for control via 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 the user code in Flash memory
- Voltage Brown-Out configuration is 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 an external RC oscillator

Operation

This section describes the type and configuration of the programmable Flash option bits.

Option Bit Configuration By Reset

Each time the option bits are programmed or erased, the device must be Reset for the change to take place. 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 F64xx 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.

Option Bit Address Space

The first two bytes of Flash memory at addresses 0000H (see Table 99) and 0001H (see Table 100) are reserved for the user option bits. The byte at Flash memory address 0000H configures user options. The byte at Flash memory address 0001H is reserved for future use and must remain unprogrammed.

Bit	Description (Continued)
[1]	**Reserved** This bit is reserved and must be programmed to 0.
[0] FWP	**Flash Write Protect (Flash version only)** 0 = Programming, Page Erase, and Mass Erase through User Code is disabled. Mass Erase is available through the On-Chip Debugger. 1 = Programming, and Page Erase are enabled for all of Flash program memory.

Flash Memory Address 0001H

Table 100. Options Bits at Flash Memory Address 0001H

Bit	7	6	5	4	3	2	1	0
Field	Reserved							
RESET	U							
R/W	R/W							
Address	Program Memory 0001H							

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

Bit	Description
[7:0]	**Reserved** These option bits are reserved for future use and must always be 1. This setting is the default for unprogrammed (erased) Flash.

Operation

The following section describes the operation of the OCD.

OCD Interface

The On-Chip Debugger uses the DBG pin for communication with an external host. This one-pin interface is a bidirectional open-drain interface that transmits and receives data. Data transmission is half-duplex, meaning that transmit and receive operations cannot occur simultaneously. The serial data on the DBG pin is sent using the standard asynchronous data format defined in RS-232. This pin can interface the Z8 Encore! XP F64xx Series products to the serial port of a host PC using minimal external hardware.Two different methods for connecting the DBG pin to an RS-232 interface are depicted in Figures 37 and 38.

! Caution: For proper operation of the On-Chip Debugger, all power pins (V_{DD} and AV_{DD}) must be supplied with power, and all ground pins (V_{SS} and AV_{SS}) must be properly grounded. The DBG pin is open-drain and must always be connected to V_{DD} through an external pull-up resistor to ensure proper operation.

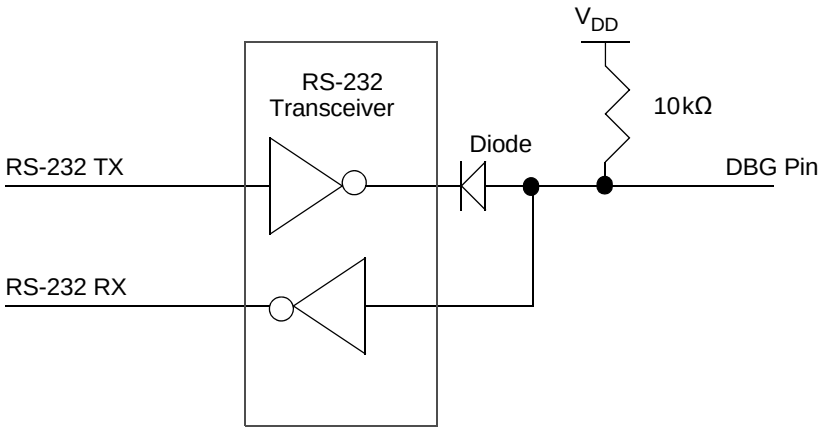

Figure 37. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #1 of 2

Figure 46 displays the maximum HALT Mode current consumption across the full operating temperature range of the device and plotted opposite the system clock frequency. All GPIO pins are configured as outputs and driven High.

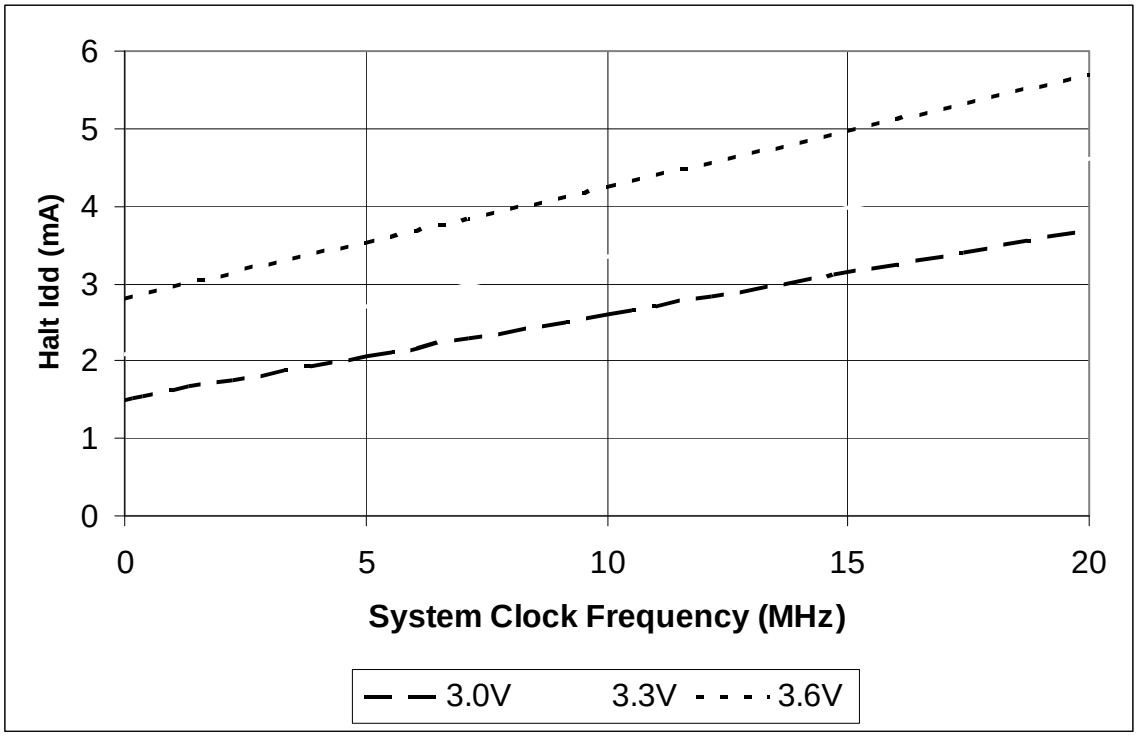

Figure 46. Maximum HALT Mode I_{CC} vs. System Clock Frequency

On-Chip Peripheral AC and DC Electrical Characteristics

Table 108. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

		T_A = –40°C to 125°C				
Symbol	**Parameter**	**Minimum**	**Typical***	**Maximum**	**Units**	**Conditions**
V_{POR}	Power-On Reset Voltage Threshold	2.40	2.70	2.90	V	$V_{DD} = V_{POR}$
V_{VBO}	Voltage Brown-Out Reset Voltage Threshold	2.30	2.60	2.85	V	$V_{DD} = V_{VBO}$
	V_{POR} to V_{VBO} hysteresis	50	100	–	mV	
	Starting V_{DD} voltage to ensure valid Power-On Reset.	–	V_{SS}	–	V	
T_{ANA}	Power-On Reset Analog Delay	–	50	–	µs	$V_{DD} > V_{POR}$; T_{POR} Digital Reset delay follows T_{ANA}
T_{POR}	Power-On Reset Digital Delay	–	6.6	–	ms	66 WDT Oscillator cycles (10kHz) + 16 System Clock cycles (20MHz)
T_{VBO}	Voltage Brown-Out Pulse Rejection Period	–	10	–	µs	$V_{DD} < V_{VBO}$ to generate a Reset.
T_{RAMP}	Time for V_{DD} to transition from V_{SS} to V_{POR} to ensure valid Reset	0.10	–	100	ms	

Note: *Data in the typical column is from characterization at 3.3V and 0°C. These values are provided for design guidance only and are not tested in production.

UART Timing

Figure 56 and Table 121 provide timing information for UART pins for the case where the Clear To Send input pin ($\overline{CTS}$) 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 $\overline{DE}$. The $\overline{CTS}$ to $\overline{DE}$ assertion delay (T1) assumes the UART Transmit Data Register has been loaded with data prior to $\overline{CTS}$ assertion.

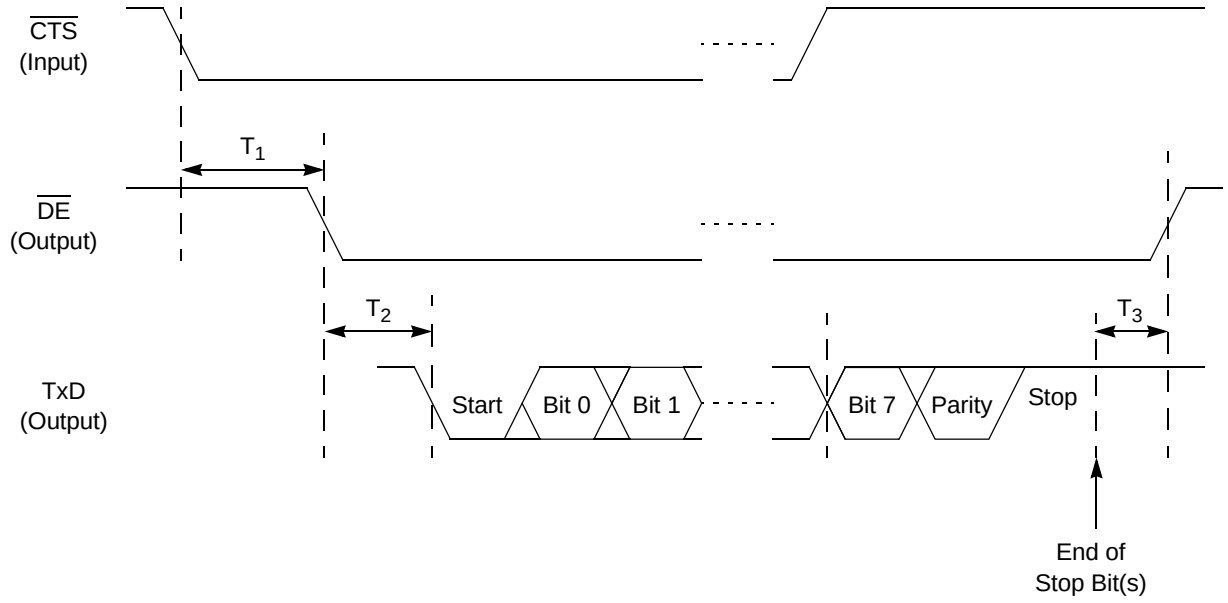

Figure 56. UART Timing with $\overline{CTS}$

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

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

G

H

I