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.
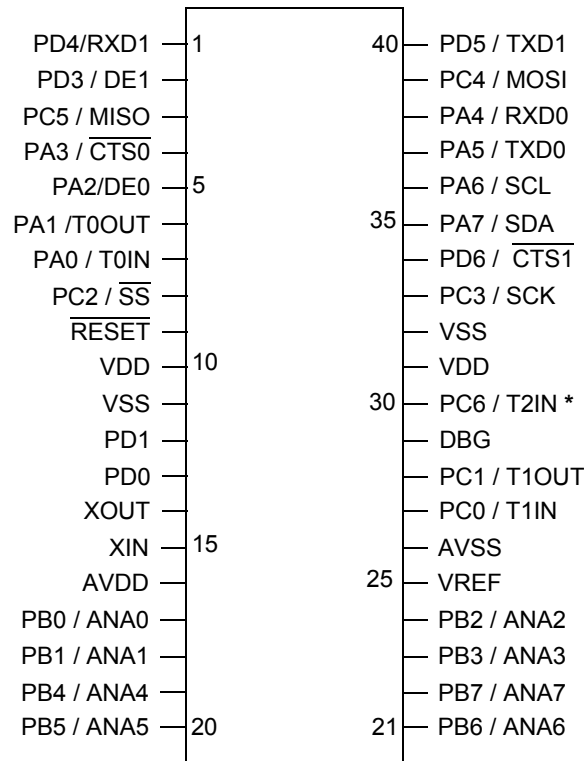
Applications of "Embedded - Microcontrollers"

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	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K 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/z8f4822vs020ec

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

8

Pin Configurations

Figure 2 through Figure 7 on page 13 display the pin configurations for all of the packages available in the Z8 Encore! XP 64K Series Flash Microcontrollers. For description of the signals, see Table 3 on page 14. Timer 3 is not available in the 40-pin and 44-pin packages.


```
PD4/RXD1 ── 1            40 ── PD5 / TXD1
 PD3 / DE1 ──               ── PC4 / MOSI
PC5 / MISO ──               ── PA4 / RXD0
 PA3 / CTS0 ──              ── PA5 / TXD0
  PA2/DE0 ── 5             ── PA6 / SCL
PA1 /T0OUT ──    35        ── PA7 / SDA
 PA0 / T0IN ──             ── PD6 / CTS1
  PC2 / SS ──              ── PC3 / SCK
    RESET ──               ── VSS
      VDD ── 10            ── VDD
      VSS ──     30        ── PC6 / T2IN *
      PD1 ──               ── DBG
      PD0 ──               ── PC1 / T1OUT
     XOUT ──               ── PC0 / T1IN
      XIN ── 15            ── AVSS
     AVDD ──      25       ── VREF
 PB0 / ANA0 ──             ── PB2 / ANA2
 PB1 / ANA1 ──             ── PB3 / ANA3
 PB4 / ANA4 ──             ── PB7 / ANA7
 PB5 / ANA5 ── 20       21 ── PB6 / ANA6
```

Note: Timer 3 is not supported.   ***** T2OUT is not supported.

Figure 2. Z8 Encore! XP 64K Series Flash Microcontrollers in 40-Pin Dual Inline Package (PDIP)

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

20

Program Memory

The eZ8™ CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP 64K Series Flash Microcontrollers contains 16 KB to 64 KB of on-chip Flash in the Program Memory address space, depending upon the device. Reading from Program Memory addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 5 describes the Program Memory maps for the 64K Series products.

Table 5. Z8 Encore! XP 64K Series Flash Microcontrollers Program Memory Maps

Program Memory Address (Hex)	Function
Z8F162x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-3FFF	Program Memory
Z8F242x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-5FFF	Program Memory
Z8F322x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-7FFF	Program Memory
Z8F482x Products	

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

23

Register File Address Map

Table 7 provides the address map for the Register File of the 64K Series products. Not all devices and package styles in the 64K Series support Timer 3 and all of the GPIO Ports. Consider registers for unimplemented peripherals as Reserved.

Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
General-Purpose RAM				
000-EFF	General-Purpose Register File RAM	—	XX	
Timer 0				
F00	Timer 0 High Byte	T0H	00	90
F01	Timer 0 Low Byte	T0L	01	90
F02	Timer 0 Reload High Byte	T0RH	FF	91
F03	Timer 0 Reload Low Byte	T0RL	FF	91
F04	Timer 0 PWM High Byte	T0PWMH	00	92
F05	Timer 0 PWM Low Byte	T0PWML	00	92
F06	Timer 0 Control 0	T0CTL0	00	93
F07	Timer 0 Control 1	T0CTL1	00	94
Timer 1				
F08	Timer 1 High Byte	T1H	00	90
F09	Timer 1 Low Byte	T1L	01	90
F0A	Timer 1 Reload High Byte	T1RH	FF	91
F0B	Timer 1 Reload Low Byte	T1RL	FF	91
F0C	Timer 1 PWM High Byte	T1PWMH	00	92
F0D	Timer 1 PWM Low Byte	T1PWML	00	92
F0E	Timer 1 Control 0	T1CTL0	00	93
F0F	Timer 1 Control 1	T1CTL1	00	94
Timer 2				
F10	Timer 2 High Byte	T2H	00	90
F11	Timer 2 Low Byte	T2L	01	90
F12	Timer 2 Reload High Byte	T2RH	FF	91
F13	Timer 2 Reload Low Byte	T2RL	FF	91
F14	Timer 2 PWM High Byte	T2PWMH	00	92
F15	Timer 2 PWM Low Byte	T2PWML	00	92
F16	Timer 2 Control 0	T2CTL0	00	93
F17	Timer 2 Control 1	T2CTL1	00	94

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

zilog 47

Reset and Stop Mode Recovery

Overview

The Reset Controller within the Z8 Encore! XP 64K Series Flash Microcontrollers controls Reset and Stop Mode Recovery operation. In typical operation, the following events cause a Reset to occur:

- Power-On Reset

- Voltage Brownout

- Watchdog Timer time-out (when configured via 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 64K Series devices are in STOP mode, a Stop Mode Recovery is initiated by either of the following:

- Watchdog Timer time-out

- GPIO Port input pin transition on an enabled Stop Mode Recovery source

- DBG pin driven Low

Reset Types

The 64K Series provides two different 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 64K Series devices and the source of the Reset. Table 8 lists the types of Reset and their operating characteristics.

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

59

GPIO Alternate Functions

Many of the GPIO port pins can be used as both general-purpose I/O and to provide access to on-chip peripheral functions such as the timers and serial communication devices. The Port A–H Alternate Function sub-registers configure these pins for either general-purpose I/O or alternate function operation. When a pin is configured for alternate function, control of the port pin direction (input/output) is passed from the Port A–H Data Direction registers to the alternate function assigned to this pin. Table 12 lists the alternate functions associated with each port pin.

Table 12. Port Alternate Function Mapping

Port	Pin	Mnemonic	Alternate Function Description
Port A	PA0	T0IN	Timer 0 Input
	PA1	T0OUT	Timer 0 Output
	PA2	DE0	UART 0 Driver Enable
	PA3	CTS0	UART 0 Clear to Send
	PA4	RXD0/IRRX0	UART 0/IrDA 0 Receive Data
	PA5	TXD0/IRTX0	UART 0/IrDA 0 Transmit Data
	PA6	SCL	I^2C Clock (automatically open-drain)
	PA7	SDA	I^2C Data (automatically open-drain)
Port B	PB0	ANA0	ADC Analog Input 0
	PB1	ANA1	ADC Analog Input 1
	PB2	ANA2	ADC Analog Input 2
	PB3	ANA3	ADC Analog Input 3
	PB4	ANA4	ADC Analog Input 4
	PB5	ANA5	ADC Analog Input 5
	PB6	ANA6	ADC Analog Input 6
	PB7	ANA7	ADC Analog Input 7

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

66

Port A–H Input Data Registers

Reading from the Port A–H Input Data registers (Table 21) returns the sampled values from the corresponding port pins. The Port A–H Input Data registers are Read-only.

Table 21. Port A–H Input Data Registers (PxIN)

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

PIN[7:0]—Port Input Data
Sampled data from the corresponding port pin input.
0 = Input data is logical 0 (Low).
1 = Input data is logical 1 (High).

Port A–H Output Data Register

The Port A–H Output Data register (Table 22) writes output data to the pins.

Table 22. Port A–H Output Data Register (PxOUT)

BITS	7	6	5	4	3	2	1	0
FIELD	POUT7	POUT6	POUT5	POUT4	POUT3	POUT2	POUT1	POUT0
RESET	0							
R/W	R/W							
ADDR	FD3H, FD7H, FDBH, FDFH, FE3H, FE7H, FEBH, FEFH							

POUT[7:0]—Port Output Data
These bits contain the data to be driven out from the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation.
0 = Drive a logical 0 (Low).
1= Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding Port Output Control register bit to 1.

**Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification**

68

Table 23. Interrupt Vectors in Order of Priority

Priority	Program Memory Vector Address	Interrupt Source
Highest	0002H	Reset (not an interrupt)
	0004H	Watchdog Timer (see Watchdog Timer on page 97)
	0006H	Illegal Instruction Trap (not an interrupt)
	0008H	Timer 2
	000AH	Timer 1
	000CH	Timer 0
	000EH	UART 0 receiver
	0010H	UART 0 transmitter
	0012H	I^{2}C
	0014H	SPI
	0016H	ADC
	0018H	Port A7 or Port D7, rising or falling input edge
	001AH	Port A6 or Port D6, rising or falling input edge
	001CH	Port A5 or Port D5, rising or falling input edge
	001EH	Port A4 or Port D4, rising or falling input edge
	0020H	Port A3 or Port D3, rising or falling input edge
	0022H	Port A2 or Port D2, rising or falling input edge
	0024H	Port A1 or Port D1, rising or falling input edge
	0026H	Port A0 or Port D0, rising or falling input edge
	0028H	Timer 3 *(not available in 44-pin packages)*
	002AH	UART 1 receiver
	002CH	UART 1 transmitter
	002EH	DMA
	0030H	Port C3, both input edges
	0032H	Port C2, both input edges
	0034H	Port C1, both input edges
Lowest	0036H	Port C0, both input edges

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

86

Follow the steps below for configuring a timer for PWM mode and initiating the PWM operation:

1. Write to the Timer Control 1 register to:
   – Disable the timer
   – Configure the timer for PWM mode
   – Set the prescale value
   – Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function

2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of `0001H`.

3. Write to the PWM High and Low Byte registers to set the PWM value.

4. Write to the Timer Reload High and Low Byte registers to set the Reload value (PWM period). The Reload value must be greater than the PWM value.

5. If desired, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.

6. Configure the associated GPIO port pin for the Timer Output alternate function.

7. Write to the Timer Control 1 register to enable the timer and initiate counting.

The PWM period is given by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than `0001H` is loaded into the Timer High and Low Byte registers, the ONE-SHOT mode equation must be used to determine the first PWM time-out period.

If `TPOL` is set to 0, the ratio of the PWM output High time to the total period is given by:

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

If `TPOL` is set to 1, the ratio of the PWM output High time to the total period is given by:

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

CAPTURE Mode

In CAPTURE mode, the current timer count value is recorded when the desired 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.

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

98

Table 47. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value (Hex)	WDT Reload Value (Decimal)	Approximate Time-Out Delay (with 10 kHz typical WDT oscillator frequency)	
		Typical	Description
000004	4	400 μs	Minimum time-out delay
FFFFFF	16,777,215	1677.5 s	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 64K 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 on programming of the WDT_RES Option Bit, see Option Bits on page 195.

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 FFFFFH 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 64K 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 STOP mode. For more information on Stop Mode Recovery, see Reset and Stop Mode Recovery on page 47.

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

112

Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte has been 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 has been 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.

> **Note:** *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 after the valid data has been 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 may not 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

Figure 18 on page 113 displays the recommended procedure for use in UART receiver interrupt service routines.

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

zilog

131

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:

- Master-In/Slave-Out
- Master-Out/Slave-In
- Serial Clock
- Slave Select

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.

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

135

register. The IRQE, PHASE, CLKPOL, WOR bits in the SPICTL register and the NUM-BITS field in the SPIMODE register must be set to be consistent with the other SPI devices. The STR bit in the SPICTL register may be used if desired to force a "startup" interrupt. The BIRQ bit in the SPICTL register and the SSV bit in the SPIMODE register are not used in SLAVE mode. The SPI baud rate generator is not used in SLAVE mode so the SPIBRH and SPIBRL registers need not be initialized.

If the slave has data to send to the master, the data must be written to the SPIDAT register before the transaction starts (first edge of SCK when $\overline{SS}$ is asserted). If the SPIDAT register is not written prior to the slave transaction, the MISO pin outputs whatever value is currently in the SPIDAT register.

Due to the delay resulting from synchronization of the SPI input signals to the internal system clock, the maximum SPICLK baud rate that can be supported in SLAVE mode is the system clock frequency (XIN) divided by 8. This rate is controlled by the SPI master.

Error Detection

The SPI contains error detection logic to support SPI communication protocols and recognize when communication errors have occurred. The SPI Status register indicates when a data transmission error has been detected.

Overrun (Write Collision)

An overrun error (write collision) indicates a write to the SPI Data register was attempted while a data transfer is in progress (in either MASTER or SLAVE modes). An overrun sets the OVR bit in the SPI Status register to 1. Writing a 1 to OVR clears this error Flag. The data register is not altered when a write occurs while data transfer is in progress.

Mode Fault (Multi-Master Collision)

A mode fault indicates when more than one Master is trying to communicate at the same time (a multi-master collision). The mode fault is detected when the enabled Master's $\overline{SS}$ pin is asserted. A mode fault sets the COL bit in the SPI Status register to 1. Writing a 1 to COL clears this error Flag.

Slave Mode Abort

In SLAVE mode of operation if the $\overline{SS}$ pin deasserts before all bits in a character have been transferred, the transaction is aborted. When this condition occurs the ABT bit is set in the SPISTAT register as well as the IRQ bit (indicating the transaction is complete). 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

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

151

7. The I²C Controller loads the I²C Shift register with the contents of the I²C Data register.

8. After one bit of address is shifted out by the SDA signal, the Transmit interrupt is asserted.

9. Software responds by writing the second byte of address into the contents of the I²C Data register.

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/$\overline{A}$	P/S

Figure 31. 10-Bit Addressed Slave Data Transfer Format

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

152

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.

Follow the steps below for a transmit operation on a 10-bit addressed slave:

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

2. Software asserts the TXI bit of the I²C Control register to enable Transmit interrupts.

3. The I²C interrupt asserts because the I²C Data register is empty.

4. Software responds to the TDRE interrupt by writing the first slave address byte to the I²C Data register. The least-significant bit must be 0 for the write operation.

5. Software asserts the START bit of the I²C Control register.

6. The I²C Controller sends the START condition to the I²C slave.

7. The I²C Controller loads the I²C Shift register with the contents of the I²C Data register.

8. After one bit of address is shifted out by the SDA signal, the Transmit interrupt is asserted.

9. Software responds by writing the second byte of address into the contents of the I²C Data register.

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 acknowledges the first address byte 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 the following steps).

12. The I²C Controller loads the I²C Shift register with the contents of the I²C Data register.

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 writing a data byte to the I²C Data register.

15. The I²C Controller completes shifting the contents of the shift register on the SDA signal.

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

184

Table 90. Flash Memory Sector Addresses

Sector Number	Flash Sector Address Ranges				
	Z8F162x	**Z8F242x**	**Z8F322x**	**Z8F482x**	**Z8F642x**
0	0000H-07FFH	0000H-0FFFH	0000H-0FFFH	0000H-1FFFH	0000H-1FFFH
1	0800H-0FFFH	1000H-1FFFH	1000H-1FFFH	2000H-3FFFH	2000H-3FFFH
2	1000H-17FFH	2000H-2FFFH	2000H-2FFFH	4000H-5FFFH	4000H-5FFFH
3	1800H-1FFFH	3000H-3FFFH	3000H-3FFFH	6000H-7FFFH	6000H-7FFFH
4	2000H-27FFH	4000H-4FFFH	4000H-4FFFH	8000H-9FFFH	8000H-9FFFH
5	2800H-2FFFH	5000H-5FFFH	5000H-5FFFH	A000H-BFFFH	A000H-BFFFH
6	3000H-37FFH	N/A	6000H-6FFFH	N/A	C000H-DFFFH
7	3800H-3FFFH	N/A	7000H-7FFFH	N/A	E000H-FFFFH

Figure 35. Flash Memory Arrangement

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

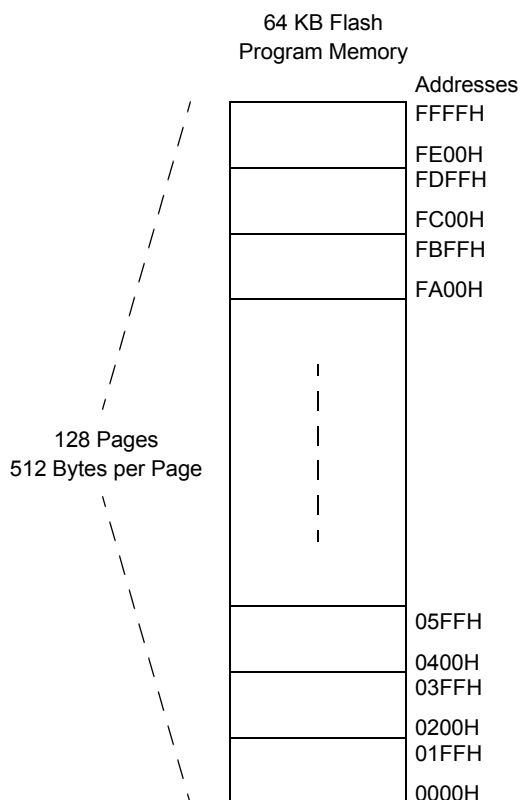194

Z8 Encore! XP® 64K Series Flash Microcontrollers
**Product Specification**

200

Operation

OCD Interface

The On-Chip Debugger uses the DBG pin for communication with an external host. This one-pin interface is a bi-directional open-drain interface that transmits and receives data. Data transmission is half-duplex, in that transmit and receive 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 64K 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 Figure 37 and Figure 38 on page 201.

⚠️ **Caution:** *For 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.
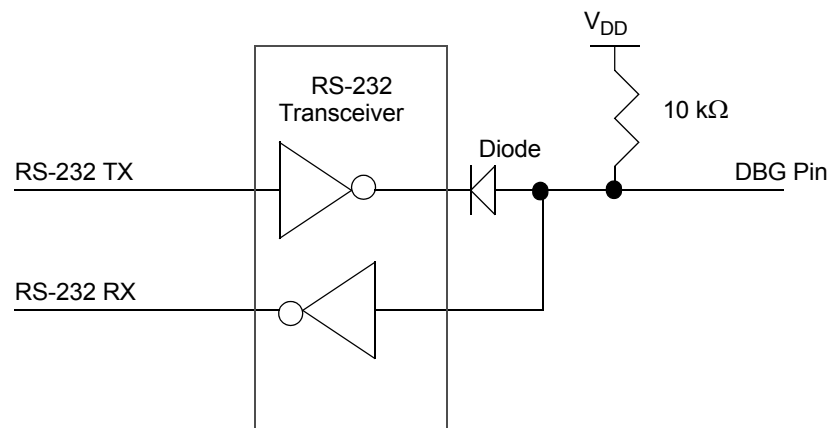
Figure 37. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface (1)

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

211

On-Chip Oscillator

Overview

The products in the 64K Series 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

The 64K Series products support four 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 195.

Crystal Oscillator Operation

Figure 40 on page 212 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 104 on page 212. Resistor R1 is optional and limits total power dissipation by the crystal. The printed circuit board layout

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

226

On-Chip Peripheral AC and DC Electrical Characteristics

Table 107. Power-On Reset and Voltage Brownout Electrical Characteristics and Timing

| Symbol | Parameter | T_A = –40 °C to 125 °C | | | Units | Conditions |
		Minimum	Typical[1]	Maximum		
V_{POR}	Power-On Reset Voltage Threshold	2.40	2.70	2.90	V	V_{DD} = V_{POR}
V_{VBO}	Voltage Brownout 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 (10 kHz) + 16 System Clock cycles (20 MHz)
T_{VBO}	Voltage Brownout Pulse Rejection Period	–	10	–	µs	V_{DD} < V_{VBO} to generate a Reset.
T_{RAMP}	Time for VDD to transition from V_{SS} to V_{POR} to ensure valid Reset	0.10	–	100	ms	

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

Z8 Encore! XP® 64K Series Flash Microcontrollers
Product Specification

274

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	I²C	SPI	UARTs with IrDA	Description
Z8F162x with 16 KB Flash, 10-Bit Analog-to-Digital Converter										
Standard Temperature: 0 °C to 70 °C										
Z8F1621PM020SC	16 KB	2 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F1621AN020SC	16 KB	2 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F1621VN020SC	16 KB	2 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F1622AR020SC	16 KB	2 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F1622VS020SC	16 KB	2 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Extended Temperature: –40 °C to +105 °C										
Z8F1621PM020EC	16 KB	2 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F1621AN020EC	16 KB	2 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F1621VN020EC	16 KB	2 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F1622AR020EC	16 KB	2 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F1622VS020EC	16 KB	2 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Automotive/Industrial Temperature: –40 °C to +125 °C										
Z8F1621PM020AC	16 KB	2 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F1621AN020AC	16 KB	2 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F1621VN020AC	16 KB	2 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F1622AR020AC	16 KB	2 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F1622VS020AC	16 KB	2 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F64200100KITG										Development Kit
ZUSBSC00100ZACG										USB Smart Cable Accessory Kit
ZUSBOPTSC01ZACG										Opto-Isolated USB Smart Cable Accessory Kit

Note: Replace C with G for lead-free packaging.