



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	5MHz
Connectivity	IrDA, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, PWM, WDT
Number of I/O	16
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0223hh005sc

 **Warning:** DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2008 by Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

Revision History

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

Date	Revision Level	Description	Page No
March 2008	14	Changed title to Z8 Encore! XP F0823 Series and the contents to match the title.	All
December 2007	13	Updated title from Z8 Encore! 8K and 4K Series to Z8 Encore! XP Z8F0823 Series. Updated Figure 3, Table 15, Table 35, Table 59 through Table 61, Table 119, and Part Number Suffix Designations section.	8, 39, 59, 91, 196, and 226
August 2007	12	Updated Table 1, Table 16, and Program Memory section.	2, 42, and 13
June 2007	11	Updated to combine Z8 Encore! 8K and Z8 Encore! 4K Series.	All
December 2006	10	Updated Ordering Information chapter.	217

Table 5. Pin Characteristics (8-Pin Devices)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt- Trigger Input	Open Drain Output	5 V Tolerance
PA0/DBG	I/O	I (but can change during reset if key sequence detected)	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
PA1	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
$\overline{\text{RESET}}$ / PA2	I/O	I/O (defaults to RESET)	N/A	Yes	Programmable for PA2; always on for RESET	Yes	Programmable for PA2; always on for RESET	Yes, unless pull-ups enabled
PA[5:3]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	Yes, unless pull-ups enabled
VDD	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
VSS	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 9. 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 Internal Precision Oscillator Cycles
Stop Mode Recovery	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	66 Internal Precision Oscillator Cycles + IPO startup time

During a System Reset or Stop Mode Recovery, the IPO requires 4 μ s to start up. Then the Z8 Encore! XP F0823 Series device is held in Reset for 66 cycles of the Internal Precision Oscillator. If the crystal oscillator is enabled in the Flash option bits, this reset period is increased to 5000 IPO cycles. When a reset occurs because of a low voltage condition or Power-On Reset, this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the reset period, the device remains in reset until the pin is deasserted.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address.

When the control registers are re-initialized by a system reset, the system clock after reset is always the IPO. The software must reconfigure the oscillator control block, such that the correct system clock source is enabled and selected.

Reset Sources

Table 10 lists the possible sources of a System Reset.

vector address. Following Stop Mode Recovery, the STOP bit in the Watchdog Timer Control Register is set to 1. Table 11 lists the Stop Mode Recovery sources and resulting actions. The section following the table provides more detailed information on each of the Stop Mode Recovery sources.

Table 11. Stop Mode Recovery Sources and Resulting Action

Operating Mode	Stop Mode Recovery Source	Action
STOP mode	Watchdog Timer time-out when configured for Reset	Stop Mode Recovery
	Watchdog Timer time-out when configured for interrupt	Stop Mode Recovery followed by interrupt (if interrupts are enabled)
	Data transition on any GPIO port pin enabled as a Stop Mode Recovery source	Stop Mode Recovery
	Assertion of external $\overline{\text{RESET}}$ Pin	System Reset
	Debug Pin driven Low	System Reset

Stop Mode Recovery Using Watchdog Timer Time-Out

If the Watchdog Timer times out during STOP mode, the device undergoes a Stop Mode Recovery sequence. In the Watchdog Timer Control register, the WDT and STOP bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and Z8 Encore! XP[®] F0823 Series device is configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery Using a GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery.

► **Note:** *The SMR pulses shorter than specified does not trigger a recovery. When this happens, the STOP bit in the Reset Status (RSTSTAT) register is set to 1.*

! **Caution:** *In STOP mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the port transition only if the signal stays on the port pin through the end of the Stop Mode Recovery delay. As a result, short pulses on the port pin can initiate Stop Mode Recovery without being written to the Port Input Data register or without initiating an interrupt (if enabled for that pin).*

- 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 PWM Control register to set the PWM dead band delay value. The deadband delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM high and low byte registers). It must also be less than the duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).
- 5. 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.
- 6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
- 7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
- 8. Write to the Timer Control register to enable the timer and initiate counting.

The PWM period is represented 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 determines 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 represented 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 represented 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 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 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 INPCAP bit in TxCTL1 register is set to indicate the timer interrupt is because of an input capture event.

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. The INPCAP bit in TxCTL1 register clears indicating the timer interrupt is not because of an input capture event.

Follow the steps below for configuring a timer for CAPTURE mode and initiating the count:

1. Write to the Timer Control 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. Clearing these registers allows the software to determine if interrupts were generated by either a Capture or a Reload event. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt was generated by a Reload.
5. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input Capture and Reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the Reload event by setting TICONFIG field of the TxCTL1 register.
6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control register to enable the timer and initiate counting.

In CAPTURE mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

CAPTURE RESTART Mode

In CAPTURE RESTART mode, the current timer count value is recorded when the acceptable 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 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

Watchdog Timer

The Watchdog Timer (WDT) protects against corrupt or unreliable software, power faults, and other system-level problems which can place Z8 Encore! XP® F0823 Series devices into unsuitable operating states. The features of Watchdog Timer include:

- On-chip RC oscillator
- A selectable time-out response: reset or interrupt
- 24-bit programmable time-out value

Operation

The WDT is a retriggerable one-shot timer that resets or interrupts Z8 Encore! XP F0823 Series devices when the WDT reaches its terminal count. The Watchdog Timer uses a dedicated on-chip RC oscillator as its clock source. The Watchdog Timer operates in only two modes: ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. Perform an enable by executing the WDT instruction or by setting the WDT_AO Flash Option Bit. The WDT_AO bit forces the Watchdog Timer to operate immediately upon reset, even if a WDT instruction has not been executed.

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

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

where the WDT reload value is the decimal value of the 24-bit value given by {WDTU[7:0], WDTM[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10 kHz. The Watchdog Timer cannot be refreshed after it reaches 000002H. The WDT Reload Value must not be set to values below 000004H. Table 57 provides information about approximate time-out delays for the minimum and maximum WDT reload values.

Table 57. 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	28 minutes	Maximum time-out delay

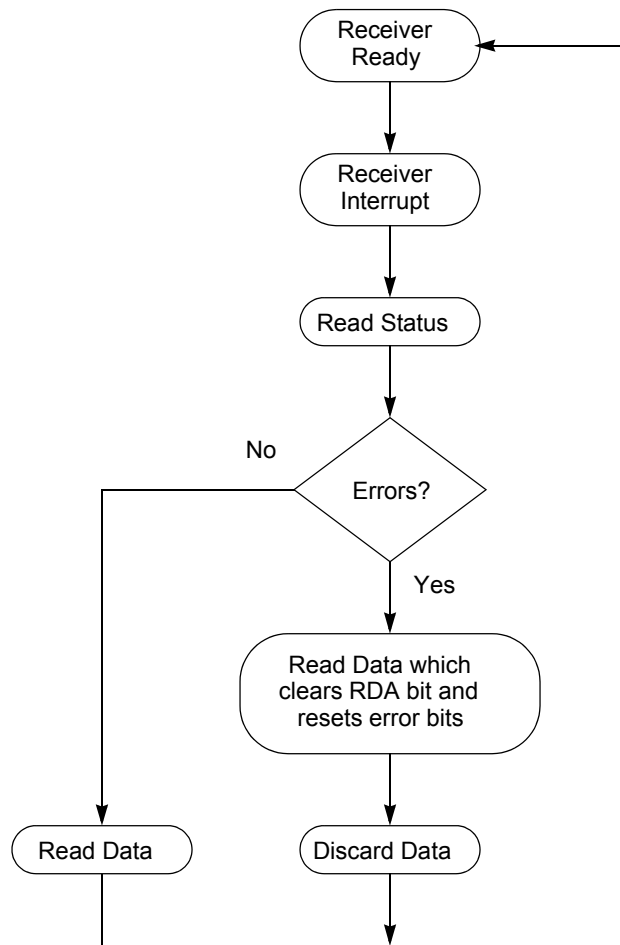


Figure 15. UART Receiver Interrupt Service Routine Flow

Baud Rate Generator Interrupts

If the Baud Rate Generator (BRG) interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This condition allows the Baud Rate Generator to function as an additional counter if the UART functionality is not employed.

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UART Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value

Table 68. UART Address Compare Register (U0ADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	COMP_ADDR							
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F45H							

COMP_ADDR—Compare Address
This 8-bit value is compared to incoming address bytes.

UART Baud Rate High and Low Byte Registers

The UART Baud Rate High and Low Byte registers (Table 69 and Table 70) combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART.

Table 69. UART Baud Rate High Byte Register (U0BRH)

BITS	7	6	5	4	3	2	1	0
FIELD	BRH							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F46H							

Table 70. UART Baud Rate Low Byte Register (U0BRL)

BITS	7	6	5	4	3	2	1	0
FIELD	BRL							
RESET	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	F47H							

The UART data rate is calculated using the following equation:

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

For a given UART data rate, calculate the integer baud rate divisor value using the following equation:

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

a bit of the Sector Protect Register has been set, it cannot be cleared except by powering down the device.

Byte Programming

The Flash Memory is enabled for byte programming after unlocking the Flash Controller and successfully enabling either Mass Erase or Page Erase. When the Flash Controller is unlocked and Mass Erase is successfully completed, all Program Memory locations are available for byte programming. In contrast, when the Flash Controller is unlocked and Page Erase is successfully enabled, only the locations of the selected page are available for byte programming. An erased Flash byte contains all 1's (FFH). The programming operation can only be used to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires execution of either the Page Erase or Mass Erase commands.

Byte Programming is accomplished using the On-Chip Debugger's Write Memory command or eZ8 CPU execution of the LDC or LDCI instructions. For a description of the LDC and LDCI instructions, refer to *eZ8 CPU Core User Manual (UM0128)* available for download at www.zilog.com. While the Flash Controller programs the Flash memory, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. To exit programming mode and lock the Flash, write any value to the Flash Control register, except the Mass Erase or Page Erase commands.

! Caution: *The byte at each address of the Flash memory cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs. Doing so may result in corrupted data at the target byte.*

Page Erase

The 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 Flash Page Select register identifies the page to be erased. Only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95h to the Flash Control register initiates the Page Erase operation. 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. If the Page Erase operation is performed using the On-Chip Debugger, poll the Flash Status register to determine when the Page Erase operation is complete. When the Page Erase is complete, the Flash Controller returns to its locked state.

Mass Erase

The Flash memory can also be Mass Erased using the Flash Controller, but only by using the On-Chip Debugger. Mass Erasing the Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked and the Mass Erase successfully enabled, writing the

value 63H to the Flash Control register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. Using the On-Chip Debugger, poll the Flash Status register to determine when the Mass Erase operation is complete. When the Mass Erase is complete, the Flash Controller returns to its locked state.

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 Row Programming algorithms by controlling the Flash programming signals directly.

Row programming is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of the Flash memory. Page Erase operations are also supported when the Flash Controller is bypassed.

For more information on bypassing the Flash Controller, refer to *Third-Party Flash Programming Support for Z8 Encore! (AN0117)* available for download at www.zilog.com.

Flash Controller Behavior in DEBUG Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored
- The Flash Sector Protect register is ignored for programming and erase operations
- Programming operations are not limited to the page selected in the Page Select register
- Bits in the Flash Sector Protect register can be written to one or zero
- The second write of the Page Select register to unlock the Flash Controller is not necessary
- The Page Select register can be written when the Flash Controller is unlocked
- The Mass Erase command is enabled through the Flash Control register

! Caution: *For security reasons, Flash controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the Flash controller must go through the unlock sequence again to select another page.*

Table 97. Randomized Lot ID Locations (Continued)

Info Page Address	Memory Address	Usage
5C	FE5C	Randomized Lot ID Byte 23
5D	FE5D	Randomized Lot ID Byte 22
5E	FE5E	Randomized Lot ID Byte 21
5F	FE5F	Randomized Lot ID Byte 20
61	FE61	Randomized Lot ID Byte 19
62	FE62	Randomized Lot ID Byte 18
64	FE64	Randomized Lot ID Byte 17
65	FE65	Randomized Lot ID Byte 16
67	FE67	Randomized Lot ID Byte 15
68	FE68	Randomized Lot ID Byte 14
6A	FE6A	Randomized Lot ID Byte 13
6B	FE6B	Randomized Lot ID Byte 12
6D	FE6D	Randomized Lot ID Byte 11
6E	FE6E	Randomized Lot ID Byte 10
70	FE70	Randomized Lot ID Byte 9
71	FE71	Randomized Lot ID Byte 8
73	FE73	Randomized Lot ID Byte 7
74	FE74	Randomized Lot ID Byte 6
76	FE76	Randomized Lot ID Byte 5
77	FE77	Randomized Lot ID Byte 4
79	FE79	Randomized Lot ID Byte 3
7A	FE7A	Randomized Lot ID Byte 2
7C	FE7C	Randomized Lot ID Byte 1
7D	FE7D	Randomized Lot ID Byte 0 (least significant)

OCD Status Register

The OCD Status register reports status information about the current state of the debugger and the system.

Table 100. OCD Status Register (OCDSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	DBG	HALT	FRPENB	Reserved				
RESET	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

DBG—Debug Status

0 = NORMAL mode

1 = DEBUG mode

HALT—HALT Mode

0 = Not in HALT mode

1 = In HALT mode

FRPENB—Flash Read Protect Option Bit Enable

0 = FRP bit enabled, that allows disabling of many OCD commands

1 = FRP bit has no effect

Reserved—0 when read

On-Chip Debugger Timing

Figure 31 and Table 129 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4 ns maximum rise and fall time.

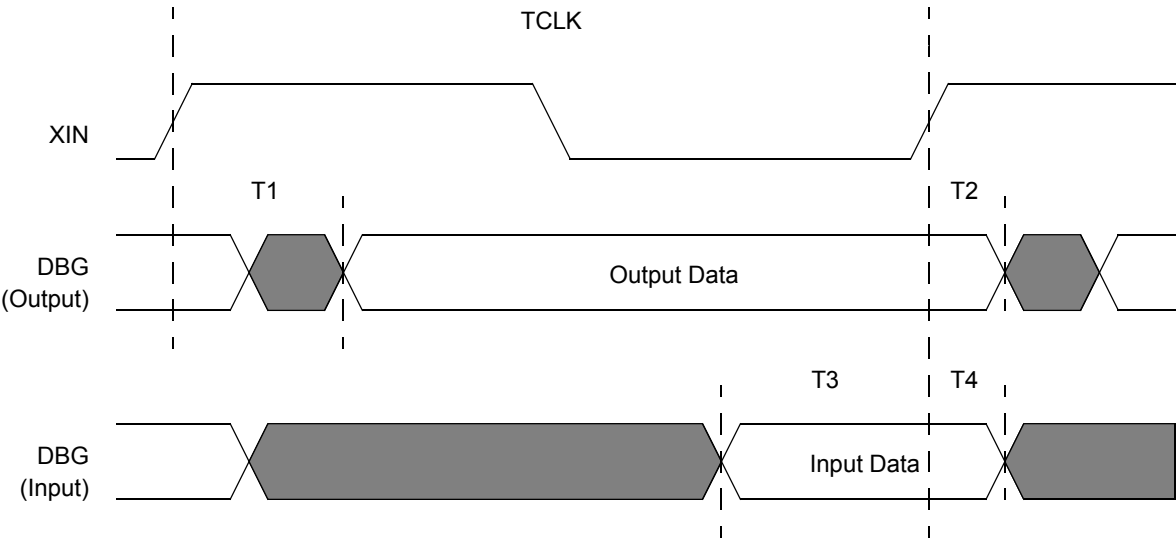


Figure 31. On-Chip Debugger Timing

Table 129. On-Chip Debugger Timing

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

Ordering Information

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Description
Z8 Encore! XP with 8 KB Flash, 10-Bit Analog-to-Digital Converter								
Standard Temperature: 0 °C to 70 °C								
Z8F0823PB005SC	8 KB	1 KB	6	12	2	4	1	PDIP 8-pin package
Z8F0823QB005SC	8 KB	1 KB	6	12	2	4	1	QFN 8-pin package
Z8F0823SB005SC	8 KB	1 KB	6	12	2	4	1	SOIC 8-pin package
Z8F0823SH005SC	8 KB	1 KB	16	18	2	7	1	SOIC 20-pin package
Z8F0823HH005SC	8 KB	1 KB	16	18	2	7	1	SSOP 20-pin package
Z8F0823PH005SC	8 KB	1 KB	16	18	2	7	1	PDIP 20-pin package
Z8F0823SJ005SC	8 KB	1 KB	22	18	2	8	1	SOIC 28-pin package
Z8F0823HJ005SC	8 KB	1 KB	22	18	2	8	1	SSOP 28-pin package
Z8F0823PJ005SC	8 KB	1 KB	22	18	2	8	1	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C								
Z8F0823PB005EC	8 KB	1 KB	6	12	2	4	1	PDIP 8-pin package
Z8F0823QB005EC	8 KB	1 KB	6	12	2	4	1	QFN 8-pin package
Z8F0823SB005EC	8 KB	1 KB	6	12	2	4	1	SOIC 8-pin package
Z8F0823SH005EC	8 KB	1 KB	16	18	2	7	1	SOIC 20-pin package
Z8F0823HH005EC	8 KB	1 KB	16	18	2	7	1	SSOP 20-pin package
Z8F0823PH005EC	8 KB	1 KB	16	18	2	7	1	PDIP 20-pin package
Z8F0823SJ005EC	8 KB	1 KB	22	18	2	8	1	SOIC 28-pin package
Z8F0823HJ005EC	8 KB	1 KB	22	18	2	8	1	SSOP 28-pin package
Z8F0823PJ005EC	8 KB	1 KB	22	18	2	8	1	PDIP 28-pin package
Replace C with G for Lead-Free Packaging								

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Description
Z8 Encore! XP with 2 KB Flash								
Standard Temperature: 0 °C to 70 °C								
Z8F0213PB005SC	2 KB	512 B	6	12	2	0	1	PDIP 8-pin package
Z8F0213QB005SC	2 KB	512 B	6	12	2	0	1	QFN 8-pin package
Z8F0213SB005SC	2 KB	512 B	6	12	2	0	1	SOIC 8-pin package
Z8F0213SH005SC	2 KB	512 B	16	18	2	0	1	SOIC 20-pin package
Z8F0213HH005SC	2 KB	512 B	16	18	2	0	1	SSOP 20-pin package
Z8F0213PH005SC	2 KB	512 B	16	18	2	0	1	PDIP 20-pin package
Z8F0213SJ005SC	2 KB	512 B	24	18	2	0	1	SOIC 28-pin package
Z8F0213HJ005SC	2 KB	512 B	24	18	2	0	1	SSOP 28-pin package
Z8F0213PJ005SC	2 KB	512 B	24	18	2	0	1	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C								
Z8F0213PB005EC	2 KB	512 B	6	12	2	0	1	PDIP 8-pin package
Z8F0213QB005EC	2 KB	512 B	6	12	2	0	1	QFN 8-pin package
Z8F0213SB005EC	2 KB	512 B	6	12	2	0	1	SOIC 8-pin package
Z8F0213SH005EC	2 KB	512 B	16	18	2	0	1	SOIC 20-pin package
Z8F0213HH005EC	2 KB	512 B	16	18	2	0	1	SSOP 20-pin package
Z8F0213PH005EC	2 KB	512 B	16	18	2	0	1	PDIP 20-pin package
Z8F0213SJ005EC	2 KB	512 B	24	18	2	0	1	SOIC 28-pin package
Z8F0213HJ005EC	2 KB	512 B	24	18	2	0	1	SSOP 28-pin package
Z8F0213PJ005EC	2 KB	512 B	24	18	2	0	1	PDIP 28-pin package
Replace C with G for Lead-Free Packaging								

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Description
Z8 Encore! XP with 1 KB Flash								
Standard Temperature: 0 °C to 70 °C								
Z8F0113PB005SC	1 KB	256 B	6	12	2	0	1	PDIP 8-pin package
Z8F0113QB005SC	1 KB	256 B	6	12	2	0	1	QFN 8-pin package
Z8F0113SB005SC	1 KB	256 B	6	12	2	0	1	SOIC 8-pin package
Z8F0113SH005SC	1 KB	256 B	16	18	2	0	1	SOIC 20-pin package
Z8F0113HH005SC	1 KB	256 B	16	18	2	0	1	SSOP 20-pin package
Z8F0113PH005SC	1 KB	256 B	16	18	2	0	1	PDIP 20-pin package
Z8F0113SJ005SC	1 KB	256 B	24	18	2	0	1	SOIC 28-pin package
Z8F0113HJ005SC	1 KB	256 B	24	18	2	0	1	SSOP 28-pin package
Z8F0113PJ005SC	1 KB	256 B	24	18	2	0	1	PDIP 28-pin package
Extended Temperature: -40 °C to 105 °C								
Z8F0113PB005EC	1 KB	256 B	6	12	2	0	1	PDIP 8-pin package
Z8F0113QB005EC	1 KB	256 B	6	12	2	0	1	QFN 8-pin package
Z8F0113SB005EC	1 KB	256 B	6	12	2	0	1	SOIC 8-pin package
Z8F0113SH005EC	1 KB	256 B	16	18	2	0	1	SOIC 20-pin package
Z8F0113HH005EC	1 KB	256 B	16	18	2	0	1	SSOP 20-pin package
Z8F0113PH005EC	1 KB	256 B	16	18	2	0	1	PDIP 20-pin package
Z8F0113SJ005EC	1 KB	256 B	24	18	2	0	1	SOIC 28-pin package
Z8F0113HJ005EC	1 KB	256 B	24	18	2	0	1	SSOP 28-pin package
Z8F0113PJ005EC	1 KB	256 B	24	18	2	0	1	PDIP 28-pin package
Replace C with G for Lead-Free Packaging								