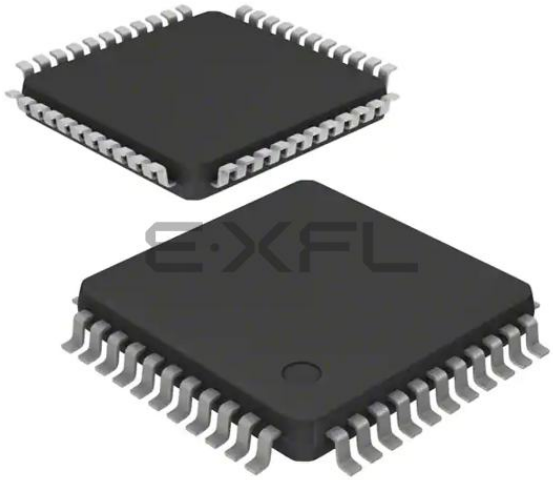


Welcome to E-XFL.COM

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	Active
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	37
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0880an020eg

List of Tables

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)[1]	Page #
Watchdog Timer				
FF2	Watchdog Timer Reload High Byte	WDTH	FF	143
FF3	Watchdog Timer Reload Low Byte	WDTL	FF	143
FF4–FF5	Reserved	—	XX	
Trim Bit Control				
FF6	Trim Bit Address	TRMADR	00	281
FF7	Trim Data	TRMDR	XX	281
Flash Memory Controller				
FF8	Flash Control	FCTL	00	272
	Flash Status	FSTAT	00	272
FF9	Flash Page Select	FPS	00	273
	Flash Sector Protect	FPROT	00	274
FFA	Flash Programming Frequency High Byte	FFREQH	00	275
FFB	Flash Programming Frequency Low Byte	FFREQL	00	275
eZ8 CPU				
FFC	Flags	—	XX	refer to the eZ8 CPU Core User Manual (UM0128)
FFD	Register Pointer	RP	XX	
FFE	Stack Pointer High Byte	SPH	XX	
FFF	Stack Pointer Low Byte	SPL	XX	

Notes:
1. XX=Undefined.
2. The Reserved space can be configured as General-Purpose Register File RAM depending on the user option bits (see the User Option Bits chapter on page 277) and the on-chip PRAM size (see the Ordering Information chapter on page 372). If the PRAM is programmed as General-Purpose Register File RAM on Reserved space, the starting address always begins immediately after the end of General-Purpose Register File RAM.

7.11.14. LED Drive Level Registers

Two LED Drive Level registers consist of the LED Drive Level High Bit Register (LEDLVLH[7:0]) and the LED Drive Level Low Bit Register (LEDLVLL[7:0]), as shown in Tables 34 and 35. Two control bits, LEDLVLH[x] and LEDLVLL[x], are used to select one of four programmable current drive levels for each associated Port C[x] pin. Each Port C pin is individually programmable.

Table 34. LED Drive Level High Bit Register (LEDLVLH)

Bits	7	6	5	4	3	2	1	0
Field	LEDLVLH							
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
Address	F83H							

Table 35. LED Drive Level Low Bit Register (LEDLVLL)

Bits	7	6	5	4	3	2	1	0
Field	LEDLVLL							
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
Address	F84H							

Bit	Description
[7:0] LEDLVLH, LEDLVLL	**LED Drive Level High Bit** **LED Drive Level Low Bit** These bits are used to set the LED drive current. {LEDLVLH[x], LEDLVLL[x]}, in which x=Port C[0] to Port C[7]. Select one of the following four programmable current drive levels for each Port C pin. 00 = 3 mA 01 = 7 mA 10 = 13 mA 11 = 20 mA

9. Counting begins on the first transition of the Timer Input signal. No interrupt is generated by this first edge.

In CAPTURE/COMPARE Mode, the elapsed time from timer start to Capture event is calculated using the following equation:

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

9.2.3.13. DEMODULATION Mode

In DEMODULATION Mode, the timer begins counting on the first external Timer Input transition. The appropriate transition (rising edge or falling edge or both) is set by the TPOL bit in the Timer Control 1 Register and TPOLHI bit in the Timer Control 2 Register. The Timer counts timer clocks up to the 16-bit reload value.

Every subsequent appropriate transition (after the first) of the Timer Input signal captures the current count value. The Capture value is written to the Timer PWM0 High and Low Byte registers for rising input edges of the timer input signal. For falling edges the capture count value is written to the Timer PWM1 High and Low Byte registers. 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. If the TPOLHI bit in the Timer Control 2 Register is set, a Capture is executed on both the rising and falling edges of the input signal.

Whenever the Capture event occurs, an interrupt is generated and the timer continues counting. The corresponding event flag bit in the Timer Status Register, PWMxEF, is set to indicate that the timer interrupt is due to an input Capture event.

The timer counts up to the 16-bit Compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H`, and counting resumes. The RTOEF event flag bit in the Timer Status Register is set to indicate that the timer interrupt is due to a Reload event. Software can use this bit to determine if a Reload occurred prior to a Capture.

Observe the following steps to configure a timer for DEMODULATION Mode and initiate the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer

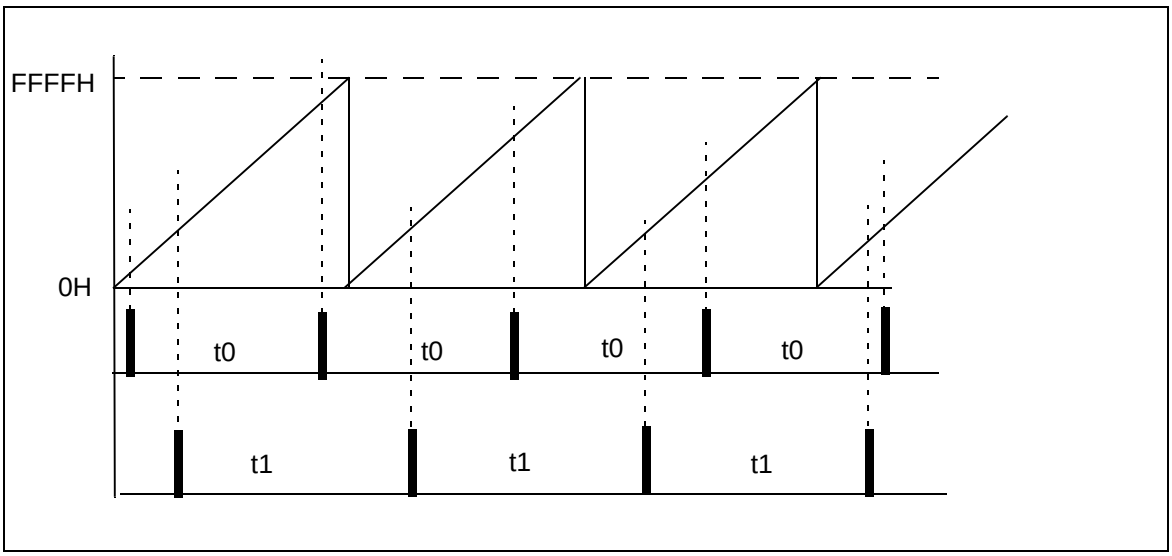


Figure 18. Count Max Mode with Channel Compare

10.7. Multi-Channel Timer Control Register Definitions

This section defines the features of the following Multi-Channel Timer Control registers.

Multi-Channel Timer High and Low Byte Registers: see page 130

Multi-Channel Timer Reload High and Low Byte Registers: see page 130

Multi-Channel Timer Subaddress Register: see page 131

Multi-Channel Timer Subregister x (0, 1, or 2): see page 132

Multi-Channel Timer Control 0, Control 1 Registers: see page 132

Multi-Channel Timer Channel Status 0 and Status 1 Registers: see page 135

Multi-Channel Timer Channel-y Control Registers: see page 137

Multi-Channel Timer Channel-y High and Low Byte Registers: see page 139

10.7.1. Multi-Channel Timer Address Map

Table 69 defines the byte address offsets for the Multi-channel Timer registers. For saving address space, a subaddress is used for the Timer Control 0, Timer Control 1, Channel Status 0, Channel Status 1, Channel-y Control, and Channel-y High and Low byte registers. Only the Timer High and Low Byte registers and the Reload High and Low Byte registers can be directly accessed.

Chapter 11. Watchdog Timer

The Watchdog Timer (WDT) function helps protect against corrupted or unreliable software and other system-level problems that can place the Z8 Encore! XP F1680 Series MCU into unsuitable operating states. The WDT includes the following features:

- On-chip RC oscillator
- A selectable time-out response: Reset or System Exception
- 16-bit programmable time-out value

11.1. Operation

The WDT is a retriggerable one-shot timer that resets or interrupts the Z8 Encore! XP F1680 Series when the WDT reaches its terminal count. The WDT uses its own dedicated on-chip RC oscillator as its clock source. The WDT has only two modes of operation—ON and OFF. After it is enabled, the WDT always counts and must be refreshed to prevent a time-out. An enable can be performed by executing the WDT instruction or by writing the WDT_AO option bit. When cleared to 0, the WDT_AO bit enables the WDT to operate continuously, even if a WDT instruction has not been executed.

To minimize power consumption, the RC oscillator can be disabled. The RC oscillator is disabled by clearing the WDTEN bit in the Oscillator Control 0 Register (OSCCTL0)[1]. If the RC oscillator is disabled, the WDT will not operate.

The WDT is a 16-bit reloadable downcounter that uses two 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is calculated using the following equation:

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

In the above equation, the WDT reload value is computed using {WDTH[7:0], WDTL[7:0]} and the typical Watchdog Timer RC Oscillator frequency is 10 kHz. Users must consider system requirements when selecting the time-out delay. Table 80 indicates the approximate time-out delays for the default and maximum WDT reload values.

1. For details about this register, see Table 170 on page 319.

11.1.2.3. WDT Reset in Normal Operation

The WDT forces the device into the Reset state if it is configured to generate a Reset when a time-out occurs; the WDT status bit is set to 1 (for details, see the Reset Status Register section on page 40). For more information about Reset and the WDT status bit, see the Reset, Stop Mode Recovery and Low-Voltage Detection section on page 31. Following a Reset sequence, the WDT Counter is initialized with its reset value.

11.1.2.4. WDT Reset in STOP Mode

If enabled in STOP Mode and configured to generate a Reset when a time-out occurs and the device is in STOP Mode, the WDT initiates a Stop Mode Recovery. Both the WDT status bit and the stop bit in the Reset Status Register (RSTSTAT) are set to 1 following a WDT time-out in STOP Mode. For more information, see the Reset, Stop Mode Recovery and Low-Voltage Detection section on page 31.

11.1.3. Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watchdog Timer Reload High (WDTH) Register address unlocks the two Watchdog Timer Reload registers (WDTH and WDTL) to allow changes to the time-out period. These write operations to the WDTH Register address produce no effect on the bits in the WDTH Register. The locking mechanism prevents unwarranted writes to the Reload registers. The following sequence is required to unlock the Watchdog Timer Reload registers (WDTH and WDTL) for write access.

1. Write 55H to the Watchdog Timer Reload High Register (WDTH).
2. Write AAH to the Watchdog Timer Reload High Register (WDTH).
3. Write the appropriate value to the Watchdog Timer Reload High Register (WDTH).
4. Write the appropriate value to the Watchdog Timer Reload Low Register (WDTL). After this write occurs, the Watchdog Timer Reload registers are again locked.

All steps of the WDT Reload Unlock sequence must be written in the sequence defined above. The values in these WDT Reload registers are loaded into the counter every time a WDT instruction is executed.

11.2. Watchdog Timer Register Definitions

The two Watchdog Timer Reload registers (WDTH and WDTL) are described in the following tables.

For a given LIN-UART data rate, the integer baud rate divisor value is calculated using the following equation for LIN mode UART operation:

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

The baud rate error relative to the appropriate baud rate is calculated using the following equation:

$$\text{UART Baud Rate Error (\%)} = 100 \times \left(\frac{\text{Actual Data Rate} - \text{Desired Data Rate}}{\text{Desired Data Rate}}\right)$$

For reliable communication, the LIN-UART baud rate error must never exceed 5 percent. Tables 96 through 100 provide error data for popular baud rates and commonly-used crystal oscillator frequencies for normal UART modes of operation.

Table 96. LIN-UART Baud Rates, 20.0 MHz System Clock

Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)	Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	1	1250.0	0.00	9.60	130	9.62	0.16
625.0	2	625.0	0.00	4.80	260	4.81	0.16
250.0	5	250.0	0.00	2.40	521	2.399	–0.03
115.2	11	113.64	–1.19	1.20	1042	1.199	–0.03
57.6	22	56.82	–1.36	0.60	2083	0.60	0.02
38.4	33	37.88	–1.36	0.30	4167	0.299	–0.01
19.2	65	19.23	0.16				

Table 97. LIN-UART Baud Rates, 10.0 MHz System Clock

Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)	Applicable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	N/A	N/A	N/A	9.60	65	9.62	0.16
625.0	1	625.0	0.00	4.80	130	4.81	0.16
250.0	3	208.33	–16.67	2.40	260	2.40	–0.03
115.2	5	125.0	8.51	1.20	521	1.20	–0.03
57.6	11	56.8	–1.36	0.60	1042	0.60	–0.03

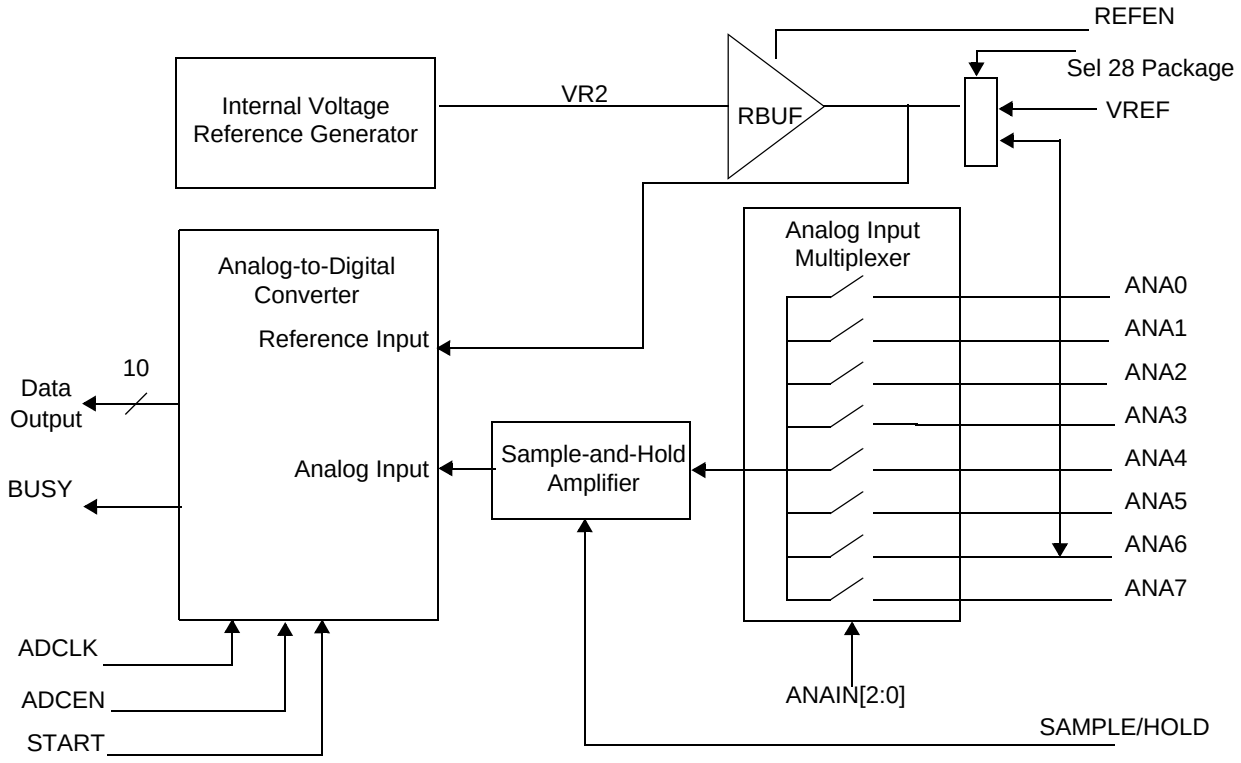


Figure 30. Analog-to-Digital Converter Block Diagram

14.2.1. ADC Timing

Each ADC measurement consists of 3 phases:

1. Input sampling (programmable, minimum of 1.8µs).
2. Sample-and-hold amplifier settling (programmable, minimum of 0.5µs).
3. Conversion is 13 ADCLK cycles.

Figure 31 displays the timing of an ADC conversion.

SPI BRG Interrupt Interval (s) = System Clock Period (s) × BRG[15:0]

16.4. ESPI Control Register Definitions

This section defines the features of the following ESPI Control registers.

ESPI Data Register: see page 213

ESPI Transmit Data Command and Receive Data Buffer Control Register: see page 214

ESPI Control Register: see page 215

ESPI Mode Register: see page 217

ESPI Status Register: see page 219

ESPI State Register: see page 220

ESPI Baud Rate High and Low Byte Registers: see page 221

16.4.1. ESPI Data Register

The ESPI Data Register, shown in Table 109, addresses both the outgoing Transmit Data Register and the incoming Receive Data Register. Reads from the ESPI Data Register return the contents of the Receive Data Register. The Receive Data Register is updated with the contents of the Shift Register at the end of each transfer. Writes to the ESPI Data Register load the Transmit Data Register unless TDRE = 0. Data is shifted out starting with bit 7. The last bit received resides in bit position 0.

With the ESPI configured as a Master, writing a data byte to this register initiates the data transmission. With the ESPI configured as a Slave, writing a data byte to this register loads the Shift Register in preparation for the next data transfer with the external Master. In either the MASTER or SLAVE modes, if TDRE = 0, writes to this register are ignored.

When the character length is less than 8 bits (as set by the NUMBITS field in the ESPI Mode Register), the transmit character must be left justified in the ESPI Data Register. A received character of less than 8 bits is right justified (last bit received is in bit position 0). For example, if the ESPI is configured for 4-bit characters, the transmit characters must be written to ESPIDATA[7:4] and the received characters are read from ESPIDATA[3:0].

17.3.7. I²C Slave Address Register

The I²C Slave Address Register, shown in Table 129, provides control over the lower order address bits used in 7 and 10 bit slave address recognition.

Table 129. I²C Slave Address Register (I2CSLVAD = 57H)

Bits	7	6	5	4	3	2	1	0
Field	SLA[7:0]							
Reset	00H							
R/W	R/W							
Address	F57H							

Bit	Description
[7:0] SLA[7:0]	**Slave Address Bits** Initialize with the appropriate Slave address value. When using 7-bit Slave addressing, SLA[9:7] are ignored.

20.2.1. Flash Operation Timing Using Flash Frequency Registers

Before performing either a program or erase operation on Flash memory, you must first configure the Flash frequency High and Low Byte registers. The Flash frequency registers allow programming and erasing of the Flash with system clock frequencies ranging from 32 kHz (32768 Hz) through 20 MHz.

The Flash frequency High and Low Byte registers combine to form a 16-bit value, FFREQ, to control timing for flash program and erase operations. The 16-bit binary Flash frequency value must contain the system clock frequency (in kHz). This value is calculated using the following equation:

$$FFREQ[15:0] = \frac{\text{System Clock Frequency (Hz)}}{1000}$$

! Caution: Flash programming and erasure are not supported for system clock frequencies below 32 kHz (32768 Hz) or above 20 MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure operation of the Z8 Encore! XP F1680 Series devices.

20.2.2. Flash Code Protection Against External Access

The user code contained within Flash memory can be protected against external access with the On-Chip Debugger. Programming the FRP Flash option bit prevents reading of the user code with the On-Chip Debugger. For more details, see the Flash Option Bits chapter on page 276 and the On-Chip Debugger chapter on page 294.

20.2.3. Flash Code Protection Against Accidental Program and Erasure

The Z8 Encore! XP F1680 Series provides several levels of protection against accidental program and erasure of the contents of Flash memory. This protection is provided by a combination of the Flash Option bits, the register locking mechanism, the page select redundancy and the sector level protection control of the Flash Controller.

20.2.3.1. Flash Code Protection Using the Flash Option Bits

The FWP Flash option bit provides Flash Program Memory protection as listed in Table 133. For more details, see the Flash Option Bits chapter on page 276.

Chapter 21. Flash Option Bits

Programmable Flash option bits allow user configuration of certain aspects of Z8 Encore! XP F1680 Series MCU operation. The feature configuration data is stored in Flash program memory and are read during Reset. The features available for control through the Flash option bits include:

- Watchdog Timer time-out response selection–interrupt or System Reset
- Watchdog Timer enabled at Reset
- The ability to prevent unwanted read access to user code in Program Memory
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Program Memory
- VBO configuration-always enabled or disabled during STOP Mode to reduce STOP Mode power consumption
- LVD voltage threshold selection
- Oscillator mode selection for high, medium and low-power crystal oscillators or an external RC oscillator
- Factory trimming information for the IPO and Temperature Sensor

21.1. Operation

This section describes the types of option bits and their configuration in the Option Configuration registers.

21.1.1. Option Bit Configuration by Reset

Each time the Flash option bits are programmed or erased, the device must be Reset for the change to take effect. During any Reset operation (System Reset or Stop Mode Recovery), the Flash option bits are automatically read from Flash Program Memory and written to the Option Configuration registers. These Option Configuration registers control operation of the devices within the Z8 Encore! XP F1680 Series MCU. 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.

Table 150. LVD_Trim Values (Continued)

	LVD Threshold (V)			
LVD_TRIM	**Minimum**	**Typical**	**Maximum**	**Description**
01101		2.55		
01110		2.50		
01111		2.45		
10000		2.40		
10001		2.35		
10010		2.30		
10011		2.25		
10100		2.20		
10101		2.15		
10110		2.10		
10111		2.05		
11000		2.00		
11001		1.95		
11010		1.90		
11011		1.85		
11100		1.80		
11101		1.75		
11110		1.70		
11111		1.65		Minimum LVD threshold, default on Reset.

21.2.4.5. Trim Bit Address 0005H

In the Trim Option Bits Register at address 0005H and shown in Table 153, all bits are reserved.

Table 153. Trim Option Bits at 0005H (TVREF)

Bits	7	6	5	4	3	2	1	0
Field	Reserved			Reserved				
Reset	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0025H							

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

Bit	Description
[7:5]	Reserved; must be 1.
[4:0]	Reserved; must be 1.

21.2.4.6. Trim Bit Address 0006H

The Trim Option Bits Register at address 0006H, shown in Table 154, governs crystal oscillator trim signals.

Table 154. Trim Option Bits at 0006H (TBG)

Bits	7	6	5	4	3	2	1	0
Field	X1_TRIM				X0_TRIM			
Reset	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Information Page Memory 0026H							

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

Bit	Description
[7:4] X1_TRIM	4-bit trimming signal for the 20M crystal oscillator.
[3:0] X0_TRIM	4-bit trimming signal for the 32K second crystal oscillator.

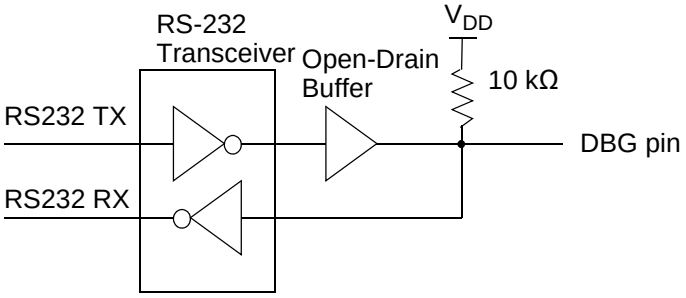


Figure 58. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #2 of 2

23.2.1. DEBUG Mode

The operating characteristics of the Z8 Encore! XP F1680 Series device in DEBUG mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions
- The system clock operates unless in STOP Mode
- All enabled on-chip peripherals operate unless in STOP Mode or otherwise defined by the on-chip peripheral to disable in DEBUG mode
- Automatically exits HALT Mode
- Constantly refreshes the Watch-Dog Timer, if enabled

23.2.1.1. Entering DEBUG Mode

The device enters DEBUG mode following any of the these operations:

- Writing the DBGMODE bit in the OCD Control Register to 1 using the OCD interface
- eZ8 CPU execution of a breakpoint (BRK) instruction (when enabled)
- Match of PC to OCDCNTR Register (when enabled)
- OCDCNTR Register decrements to `0000H` (when enabled)
- The DBG pin is Low when the device exits Reset

23.2.1.2. Exiting DEBUG Mode

The device exits DEBUG mode following any of these operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-on reset

enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

If breakpoints are enabled, the OCD can be configured to automatically enter DEBUG mode, or to loop on the break instruction. If the OCD is configured to loop on the BRK instruction, then the CPU remains able to service interrupt requests.

The loop on a BRK instruction can service interrupts in the background. For interrupts to be serviced in the background, there cannot be any breakpoints in the interrupt service routine. Otherwise, the CPU stops on the breakpoint in the interrupt routine. For interrupts to be serviced in the background, interrupts must also be enabled. Interrupts are typically disabled during critical sections of code where interrupts do not occur (such as adjusting the stack pointer or modifying shared data).

Through the OCD, host debugger software can poll the IDLE bit of the OCDSTAT Register to determine if the OCD is looping on a BRK instruction. When the host must stop the CPU on the BRK instruction on which it is looping, the host must not set the DBGMODE bit of the OCDCTL register. The CPU may have vectored to an interrupt service routine. Instead, the host clears the BRKLOOP bit, thereby allowing the CPU to finish the interrupt service routine and return to the BRK instruction. When the CPU returns to the BRK instruction on which it was previously looping, it automatically sets the DBGMODE bit and enters DEBUG mode.

The majority of the OCD commands remain disabled when the eZ8 CPU is looping on a BRK instruction. The eZ8 CPU must be in DEBUG mode before these commands can be issued.

23.2.8.1. Breakpoints in Flash Memory

The BRK instruction is op code `00H`, which corresponds to the fully programmed state of a byte in Flash memory. To implement a breakpoint, write `00H` to the appropriate address, overwriting the current instruction. To remove a breakpoint, erase the corresponding page of Flash memory and reprogram with the original data.

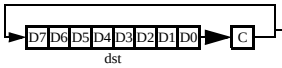
23.2.9. OCDCNTR Register

The On-Chip Debugger contains a multipurpose 16-bit Counter Register. It can be used for the following:

- Count system clock cycles between breakpoints
- Generate a BRK when it counts down to 0
- Generate a BRK when its value matches the Program Counter

When configured as a counter, the OCDCNTR Register starts counting when the On-Chip Debugger exits DEBUG mode and stops counting when it enters DEBUG mode again or

Table 186. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
POPX dst	dst ← @SP SP ← SP + 1	ER		D8	–	–	–	–	–	–	3	2
PUSH src	SP ← SP – 1 @SP ← src	R		70	–	–	–	–	–	–	2	2
		IR		71							2	3
		IM		IF70							3	2
PUSHX src	SP ← SP – 1 @SP ← src	ER		C8	–	–	–	–	–	–	3	2
RCF	C ← 0			CF	0	–	–	–	–	–	1	2
RET	PC ← @SP SP ← SP + 2			AF	–	–	–	–	–	–	1	4
RL dst		R		90	*	*	*	*	–	–	2	2
		IR		91							2	3
RLC dst		R		10	*	*	*	*	–	–	2	2
		IR		11							2	3
RR dst		R		E0	*	*	*	*	–	–	2	2
		IR		E1							2	3
RRC dst		R		C0	*	*	*	*	–	–	2	2
		IR		C1							2	3
SBC dst, src	dst ← dst – src – C	r	r	32	*	*	*	*	1	*	2	3
		r	Ir	33							2	4
		R	R	34							3	3
		R	IR	35							3	4
		R	IM	36							3	3
		IR	IM	37							3	4
SBCX dst, src	dst ← dst – src – C	ER	ER	38	*	*	*	*	1	*	4	3
		ER	IM	39							4	3

Flags notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 186. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
TM dst, src	dst AND src	r	r	72	–	*	*	0	–	–	2	3
		r	Ir	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	–	*	*	0	–	–	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	–	–	–	–	–	–	2	6
WDT				5F	–	–	–	–	–	–	1	2
XOR dst, src	dst ← dst XOR src	r	r	B2	–	*	*	0	–	–	2	3
		r	Ir	B3							2	4
		R	R	B4							3	3
		R	IR	B5							3	4
		R	IM	B6							3	3
		IR	IM	B7							3	4
XORX dst, src	dst ← dst XOR src	ER	ER	B8	–	*	*	0	–	–	4	3
		ER	IM	B9							4	3

Flags notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

V

W

X

Z