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, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 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/z8f1680hh020sg

writing this bit does not clear it. The LVD circuit can also generate an interrupt when enabled (see the Interrupt Vectors and Priority section on page 71). The LVD is not latched, so enabling the interrupt is the only way to guarantee detection of a transient low-voltage event.

The LVD circuit is either enabled or disabled by the Power Control Register bit 4. For more details, see the Power Control Register Definitions section on page 44.

5.5. Reset Register Definitions

The following sections define the Reset registers.

5.5.0.1. Reset Status Register

The Reset Status (RSTSTAT) Register, shown in Table 12, is a read-only register that indicates the source of the most recent Reset event, Stop Mode Recovery event and/or WDT time-out. Reading this register resets the upper 4 bits to 0.

This register shares its address with the Reset Status Register, which is write-only.

Table 12. Reset Status Register (RSTSTAT)

Bits	7	6	5	4	3	2	1	0
Field	POR/VBO	STOP	WDT	EXT	Reserved			LVD
Reset	See Table 13			0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	FF0H							

Bit	Description
[7] POR/VBO	**Power-On initiated VBO Reset or general VBO Reset Indicator** If this bit is set to 1, a POR or VBO Reset event occurs. This bit is reset to 0, if a WDT time-out or Stop Mode Recovery occurs. This bit is also reset to 0 when the register is read.
[6] STOP	**Stop Mode Recovery Indicator** If this bit is set to 1, a Stop Mode Recovery occurs. If the STOP and WDT bits are both set to 1, the Stop Mode Recovery occurs because of a WDT time-out. If the stop bit is 1 and the WDT bit is 0, the Stop Mode Recovery is not caused by a WDT time-out. This bit is reset by Power-On Reset or WDT time-out that occurred while not in STOP Mode. Reading this register also resets this bit.
[5] WDT	**Watchdog Timer time-out Indicator** If this bit is set to 1, a WDT time-out occurs. A POR resets this pin. A Stop Mode Recovery from a change in an input pin also resets this bit. Reading this register resets this bit. This Read must occur before clearing the WDT interrupt.

Bit	Description (Continued)
[4] EXT	**External Reset Indicator** If this bit is set to 1, a Reset initiated by the external $\overline{\text{RESET}}$ pin occurs. A POR or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit.
[3:1]	Reserved; must be 0.
[0] LVD	**Low-Voltage Detection Indicator** If this bit is set to 1 the current state of the supply voltage is below the low-voltage detection threshold. This value is not latched but is a real-time indicator of the supply voltage level.

Table 13. Reset Status Per Event

Reset or Stop Mode Recovery Event	POR	STOP	WDT	EXT
Power-On Reset or VBO Reset	1	0	0	0
Reset using $\overline{\text{RESET}}$ pin assertion	0	0	0	1
Reset using Watchdog Timer time-out	0	0	1	0
Reset using the On-Chip Debugger (OCTCTL[1] set to 1)	1	0	0	0
Reset from STOP Mode using DBG Pin driven Low	1	0	0	0
Stop Mode Recovery using GPIO pin transition	0	1	0	0
Stop Mode Recovery using Watchdog Timer time-out	0	1	1	0

Bit	Description (Continued)
[6] TPOL	**Timer Input/Output Polarity** Operation of this field is a function of the current operating modes of the timer. **ONE-SHOT Mode** When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer reload. **CONTINUOUS Mode** When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer reload. **COUNTER Mode** When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer reload. 0 = Count occurs on the rising edge of the Timer Input signal. 1 = Count occurs on the falling edge of the Timer Input signal. **PWM SINGLE OUTPUT Mode** 0 = Timer Output is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced High (1) on PWM count match and forced Low (0) on Reload. 1 = Timer Output is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced Low (0) on PWM count match and forced High (1) on Reload. **CAPTURE Mode** 0 = Count is captured on the rising edge of the Timer Input signal. 1 = Count is captured on the falling edge of the Timer Input signal. **COMPARE Mode** When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented on timer reload. **GATED Mode** 0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input. 1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input. **CAPTURE/COMPARE Mode** 0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal. 1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.

10.5. Low-Power Modes

The Z8 Encore! XP F1680 Series of MCUs contains power-saving features. The highest level of power reduction is provided by STOP Mode. The next level of power reduction is provided by HALT Mode.

10.5.1. Operation in HALT Mode

When the eZ8 CPU is operating in HALT Mode, the Multi-Channel Timer will continue to operate if enabled. To minimize current in HALT Mode, the Multi-Channel Timer must be disabled by clearing the TEN control bit.

10.5.2. Operation in STOP Mode

When the eZ8 CPU is operating in STOP Mode, the Multi-Channel Timer ceases to operate because the system clock has stopped. The registers are not reset and operation will resume after Stop Mode Recovery occurs.

10.5.3. Power Reduction During Operation

Deassertion of the TEN bit will inhibit clocking of the entire Multi-Channel Timer block. Deassertion of the CHEN bit of individual channels will inhibit clocking of channel-specific logic to minimize power consumption of unused channels. The CPU can still read and write to the registers when the enable bit(s) are deasserted.

10.6. Multi-Channel Timer Applications Examples

This section provides two brief examples that describe how the the F1680 Series multi-channel timer can be used in your application.

10.6.1. PWM Programmable Deadband Generation

The count up/down mode supports motor control applications that require dead time between output signals. Figure 17 displays dead time generation between two channels operating in count up/down mode.

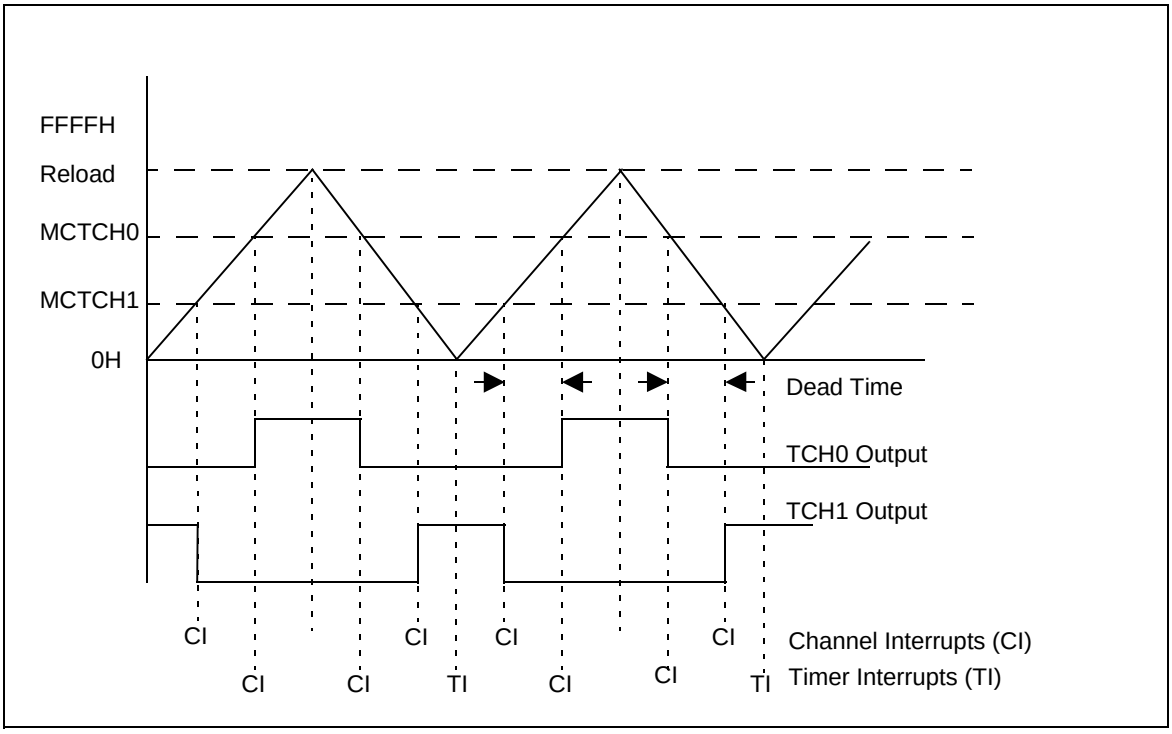

Figure 17. Count Up/Down Mode with PWM Channel Outputs and Deadband

10.6.2. Multiple Timer Intervals Generation

Figure 18 shows a timing diagram featuring two constant time intervals, T0 and T1. The timer is in Count Modulo Mode with reload = FFFFH. Channels 0 and 1 are set up for CONTINUOUS COMPARE operation. After every channel compare interrupt, the channel Capture/Compare registers are updated in the interrupt service routine by adding a constant equal to the time interval required. This operation requires that the Channel Update Enable (CHUE) bit must be set in channels 0 and 1 so that writes to the Capture/ Compare registers take affect immediately.

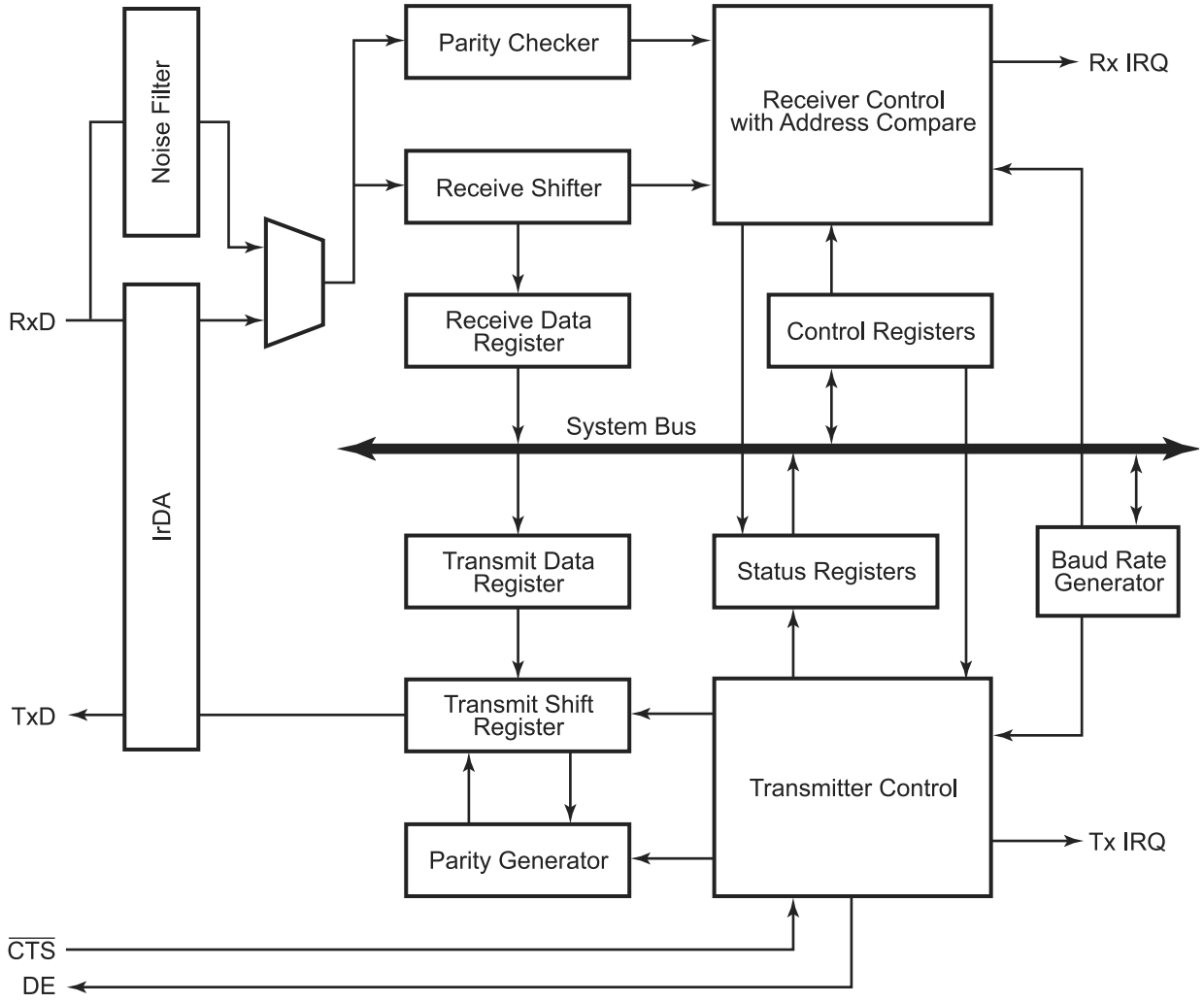

Figure 19. LIN-UART Block Diagram

12.1.1. Data Format for Standard UART Modes

The LIN-UART always transmits and receives data in an 8-bit data format with the least significant bit first. An even-or-odd parity bit or multiprocessor address/data 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 20 and 21 display the asynchronous data format employed by the LIN-UART without parity and with parity, respectively.

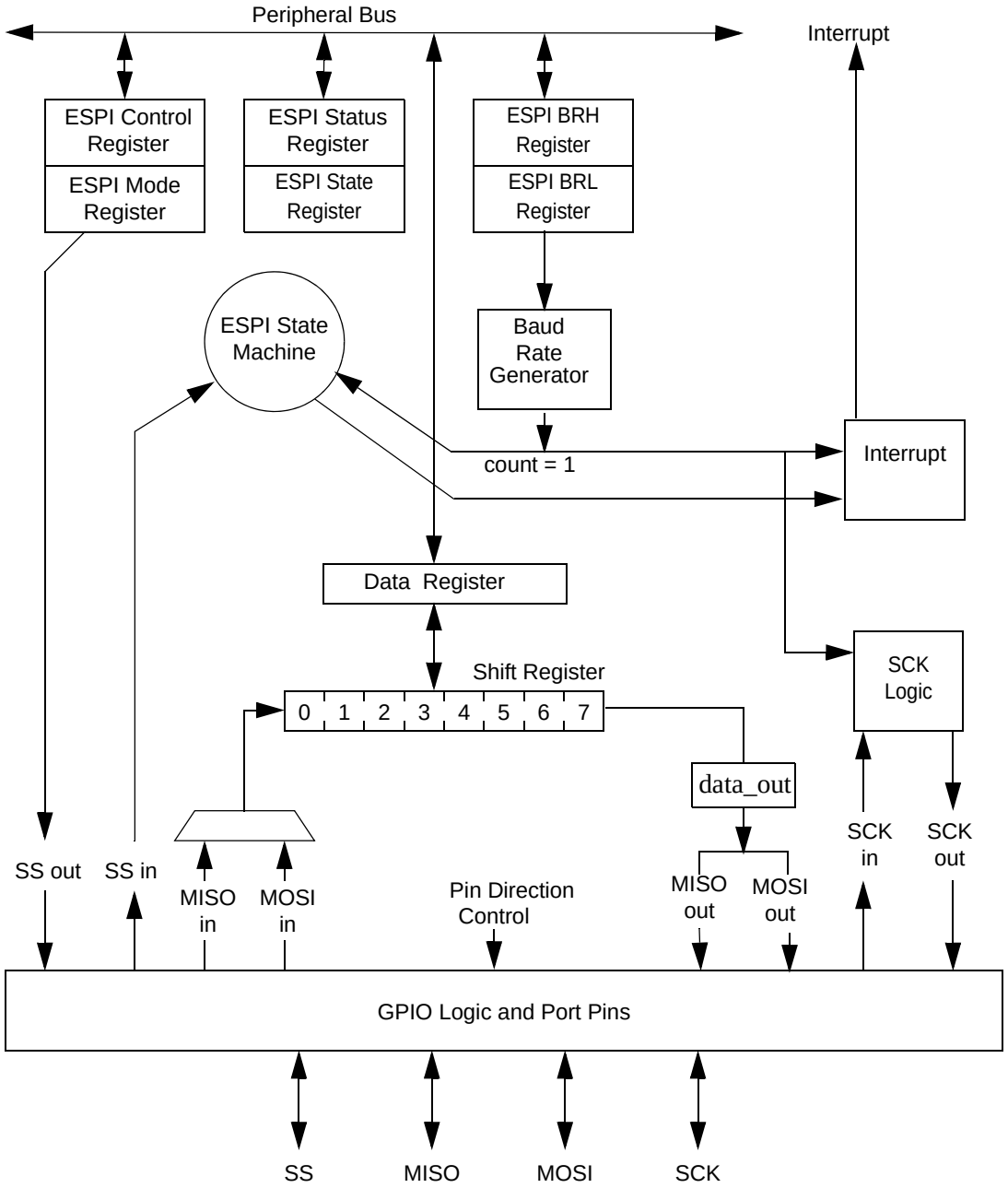

Figure 33. ESPI Block Diagram

transmitted, the hardware will automatically deassert the SSV and TEOF bits. The second method is for software to directly clear the SSV bit after the transaction completes. If software clears the SSV bit directly it is not necessary for software to also set the TEOF bit on the last transmit byte. After writing the last transmit byte, the end of the transaction can be detected by waiting for the last RDRNE interrupt or monitoring the TFST bit in the ESPI Status Register.

The transmit underrun and receive overrun errors will not occur in an SPI mode Master. If the RDRNE and TDRE requests have not been serviced before the current byte transfer completes, SCLK will be paused until the Data Register is read and written. The transmit underrun and receive overrun errors will occur in a Slave if the Slave's software does not keep up with the Master data rate. In this case the Shift Register in the Slave will be loaded with all 1s.

In the SPI mode, the SCK is active only for the data transfer with one SCK period per bit transferred. If the SPI bus has multiple Slaves, the Slave Select lines to all or all but one of the Slaves must be controlled independently by software using GPIO pins. Figure 36 displays multiple character transfer in SPI mode.

Note: When character n is transferred via the Shift Register, software responds to the receive request for character n-1 and the transmit request for character n+1.

State Register. The software responds to the Not Acknowledge interrupt by setting the stop bit and clearing the TXI bit. The I2C controller flushes the Transmit Data Register, sends the stop condition on the bus and clears the stop and NCKI bits. The transaction is complete and the following steps can be ignored.

16. The I2C controller sends a repeated start condition.
17. The I2C controller loads the I2C Shift Register with the contents of the I2C Data Register (the third address transfer).
18. The I2C controller sends `11110b`, followed by the two most-significant bits of the slave read address and a 1 (Read).
19. The I2C slave sends an Acknowledge by pulling the SDA signal Low during the next High period of SCL.
20. The I2C controller shifts in a byte of data from the slave.
21. The I2C controller asserts the Receive interrupt.
22. The software responds by reading the I2C Data Register. If the next data byte is to be the final byte, the software must set the NAK bit of the I2C Control Register.
23. The I2C controller sends an Acknowledge or Not Acknowledge to the I2C Slave, based on the value of the NAK bit.
24. If there are more bytes to transfer, the I2C controller returns to Step 18.
25. The I2C controller generates a NAK interrupt (the NCKI bit in the I2CISTAT Register).
26. The software responds by setting the stop bit of the I2C Control Register.
27. A stop condition is sent to the I2C Slave.

17.2.6. Slave Transactions

The following sections describe Read and Write transactions to the I2C controller configured for 7-bit and 10-bit Slave modes.

17.2.6.1. Slave Address Recognition

The following two slave address recognition options are supported; a description of each follows.

- Slave 7-Bit Address Recognition Mode
- Slave 10-Bit Address Recognition Mode

Slave 7-Bit Address Recognition Mode. If IRM = 0 during the address phase and the controller is configured for MASTER/SLAVE or SLAVE 7-bit address mode, the

Bit	Description (Continued)
[3] TXI	**Enable TDRE Interrupts** This bit enables interrupts when the I2C Data Register is empty.
[2] NAK	**Send NAK** Setting this bit sends a Not Acknowledge condition after the next byte of data has been received. It is automatically deasserted after the Not Acknowledge is sent or the IEN bit is cleared. If this bit is 1, it cannot be cleared to 0 by writing to the register.
[1] FLUSH	**Flush Data** Setting this bit clears the I2C Data Register and sets the TDRE bit to 1. This bit allows flushing of the I2C Data Register when an NAK condition is received after the next data byte is written to the I2C Data Register. Reading this bit always returns 0.
[0] FILTEN	**I2C Signal Filter Enable** Setting this bit enables low-pass digital filters on the SDA and SCL input signals. This function provides the spike suppression filter required in I2C Fast Mode. These filters reject any input pulse with periods less than a full system clock cycle. The filters introduce a 3-system clock cycle latency on the inputs.

17.3.4. I2C Baud Rate High and Low Byte Registers

The I2C Baud Rate High and Low Byte registers, shown in Tables 122 and 123, combine to form a 16-bit reload value, BRG[15:0], for the I2C Baud Rate Generator. The I2C baud rate is calculated using the following equation.

$$\text{I}^2\text{C Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{4 \times \text{BRG[15:0]}}$$

Note: If BRG = 0000H, use 10000H in the equation.

Table 122. I2C Baud Rate High Byte Register (I2CBRH = 53H)

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
Address	F53H							

Bit Position	Value	Description
[7:0] BRH		**I2C Baud Rate High Byte** The most significant byte, BRG[15:8], of the I2C Baud Rate Generator's reload value.

Bit	Description (Continued)
[5:2] REFLVL	**Comparator 1 Internal Reference Voltage Level** This reference is independent of the ADC voltage reference. 0000 = 0.0 V 0001 = 0.2 V 0010 = 0.4 V 0011 = 0.6 V 0100 = 0.8 V 0101 = 1.0 V (Default) 0110 = 1.2 V 0111 = 1.4 V 1000 = 1.6 V 1001 = 1.8 V 1010–1111 = Reserved
[1:0] TIMTRG	**Timer Trigger (Comparator Counter Mode)** Enable/disable timer operation. 00 = Disable Timer Trigger. 01 = Comparator 1 output works as Timer 0 Trigger. 10 = Comparator 1 output works as Timer 1 Trigger. 11 = Comparator 1 output works as Timer 2 Trigger.

20.3.4. Flash Sector Protect Register

The Flash Sector Protect Register is shared with the Flash Page Select Register. When the Flash Control Register (see page 271) is written with 5EH, the next write to this address targets the Flash Sector Protect Register. In all other cases, it targets the Flash Page Select Register.

This register selects one of the eight available Flash memory sectors to be protected. The reset state of each Sector Protect bit is an unprotected state. After a sector is protected by setting its corresponding register bit, it can only be unprotected (the register bit can only be cleared) by a System Reset. Please refer to Table 132 on page 262 and to Figures 51 through 53 to review how Flash memory is arranged by sector.

Table 137. Flash Sector Protect Register (FPROT)

Bits	7	6	5	4	3	2	1	0
Field	SPROT7	SPROT6	SPROT5	SPROT4	SPROT3	SPROT2	SPROT1	SPROT0
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	FF9H							

Bit	Description
[7:0] SPROT*x*	**Sector Protection** • On Z8F2480 devices, each bit corresponds to a 3KB Flash sector. • On Z8F1680 devices, each bit corresponds to a 2KB Flash sector. • On Z8F0880 devices, each bit corresponds to a 1KB Flash sector.

20.3.5. Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte registers, shown in Tables 138 and 139, 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) and is calculated using the following equation:

$$FFREQ[15:0] = \{FFREQH[7:0],FFREQL[7:0]\} = \frac{\text{System Clock Frequency}}{1000}$$

Caution: Flash programming and erasure is not supported for system clock frequencies below 32 kHz or above 20 MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct values to ensure proper operation of the device.

the return from the subroutine, the read byte resides in working register R0 and the read status byte resides in working register R1. The bit fields of this status byte are defined in Table 160. Also, the user code should pop the address byte off the stack.

The read routine uses 16 bytes of stack space in addition to the 1 byte of address pushed by you. Sufficient memory must be available for this stack usage. Because of the Flash memory architecture, NVDS reads exhibit a nonuniform execution time. A read operation takes between 71 µs and 258 µs (assuming a 20 MHz system clock). Slower system clock speeds result in proportionally higher execution times.

NVDS byte reads from invalid addresses (those exceeding the NVDS array size) return 0xff. Illegal read operations have a 6 µs execution time. The status byte returned by the NVDS read routine is zero for successful read. If the status byte is nonzero, there is a corrupted value in the NVDS array at the location being read. In this case, the value returned in R0 is the byte most recently written to the array that does not have an error.

Table 160. Read Status Byte

Bits	7	6	5	4	3	2	1	0
Field	Reserved			DE	Reserved	FE	IGADDR	Reserved
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:5]	Reserved; must be 0.
[4] DE	**Data Error** When reading a NVDS address, if an error is found in the latest data corresponding to the NVDS address, this bit is set to 1. NVDS source code steps forward until it finds valid data at this address.
[3]	Reserved; must be 0.
[2] FE	**Flash Error** If a Flash error is detected, this bit is set to 1.
[1] IGADDR	**Illegal Address** When NVDS byte reads occur from invalid addresses (those exceeding the NVDS array size), this bit is set to 1. Note: When the NVDS array size is 256 bytes, there is no address exceeding the size: therefore the IGADDR bit cannot be used.
[0]	Reserved; must be 0.

22.2.3. Power Failure Protection

The NVDS routines employ error checking mechanisms to ensure a power failure endangers only the most recently written byte. Bytes previously written to the array are not perturbed. For this protection to function, the VBO must be enabled (see the Low-Power

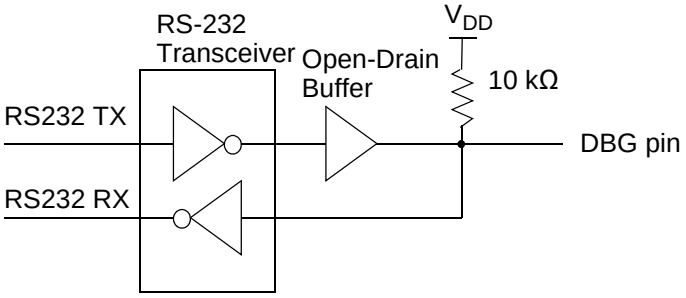

Figure 58. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, #2 of 2

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

23.4.2. OCD Status Register

The OCD Status Register, shown in Table 165, reports status information about the current state of the debugger and the system.

Table 165. OCD Status Register (OCDSTAT)

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

Bit	Description
[7] IDLE	**CPU Idle** This bit is set if the part is in Debug mode (DBGMODE is 1) or if a BRK instruction has occurred since the last time OCDCTL was written. This can be used to determine if the CPU is running or if it is idle. 0 = The eZ8 CPU is running. 1 = The eZ8 CPU is either stopped or looping on a BRK instruction.
[6] HALT	**HALT Mode** 0 = The device is not in HALT Mode. 1 = The device is in HALT Mode.
[5] RPEN	**Read Protect Option Bit Enable** 0 = The Read Protect option bit is disabled (Flash option bit is 1). 1 = The Read Protect option bit is enabled (Flash option bit is 0), disabling many OCD commands.
[4:0]	Reserved; must be 0.

27.3. eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary section on page 336, the operands, condition codes, status flags and address modes are represented by the notational shorthand provided in Table 176.

Table 176. Notational Shorthand

Notation	Description	Operand	Range
b	Bit	b	b represents a value from 0 to 7 (000B to 111B)
cc	Condition Code	—	See the Condition Codes overview in the eZ8 CPU Core User Manual (UM0128)
DA	Direct Address	Addrs	Addrs. represents a number in the range of 0000H to FFFFH
ER	Extended Addressing Register	Reg	Reg. represents a number in the range of 000H to FFFH
IM	Immediate Data	#Data	Data is a number between 00H to FFH
Ir	Indirect Working Register	@Rn	n = 0 –15
IR	Indirect Register	@Reg	Reg. represents a number in the range of 00H to FFH
Irr	Indirect Working Register Pair	@RRp	p = 0, 2, 4, 6, 8, 10, 12 or 14
IRR	Indirect Register Pair	@Reg	Reg. represents an even number in the range 00H to FEH
p	Polarity	p	Polarity is a single bit binary value of either 0B or 1B.
r	Working Register	Rn	n = 0–15
R	Register	Reg	Reg. represents a number in the range of 00H to FFH
RA	Relative Address	X	X represents an index in the range of +127 to –128, which is an offset relative to the address of the next instruction
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12 or 14
RR	Register Pair	Reg	Reg. represents an even number in the range of 00H to FEH
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to –128 range.

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
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	dst ← dst AND src	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	dst[bit] ← 0	r		E2	–	*	*	0	–	–	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	–	*	*	0	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	–	*	*	0	–	–	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
CALL dst	SP ← SP -2 @SP ← PC PC ← dst	IRR		D4	–	–	–	–	–	–	2	6
		DA		D6							3	3
CCF	C ← ~C			EF	*	–	–	–	–	–-	1	2

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

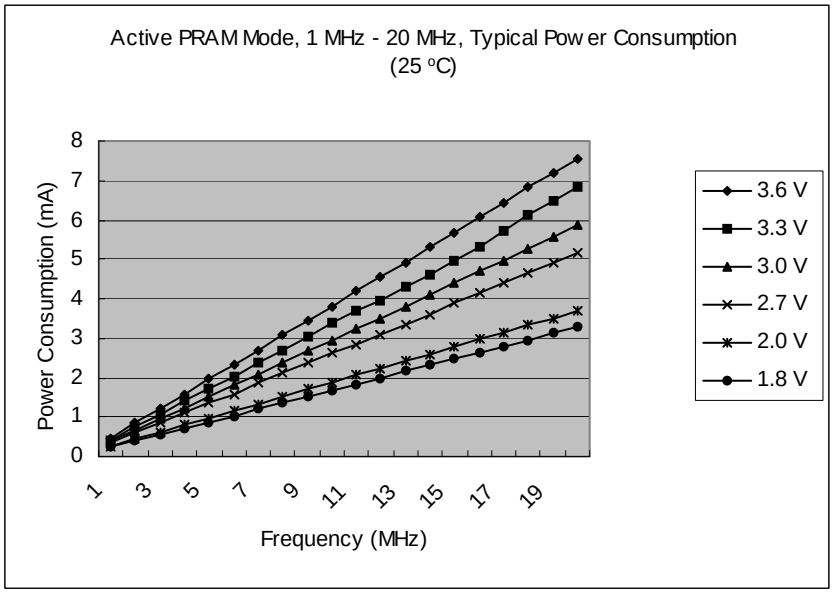

Figure 70. Typical Active PRAM Mode Supply Current (1–20MHz)

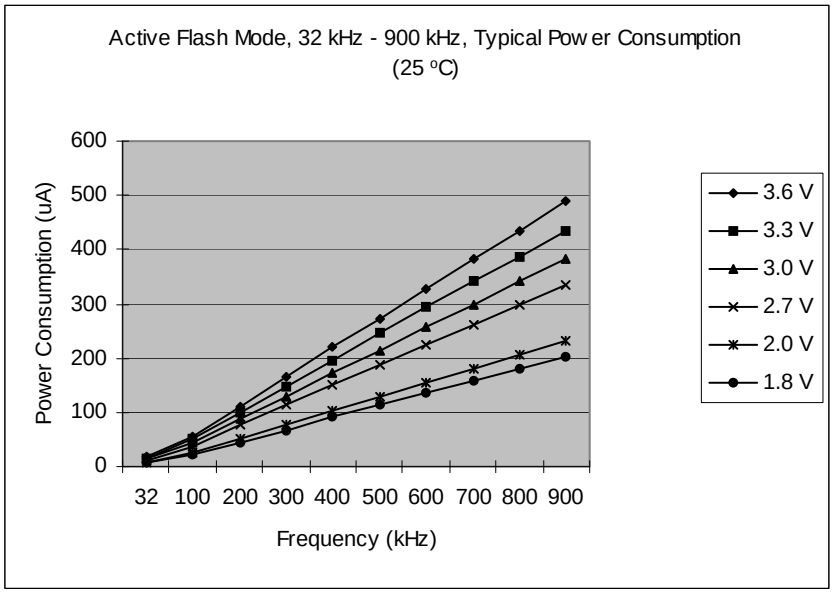

Figure 71. Typical Active Flash Mode Supply Current (32–900kHz)

Index