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	23
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 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1680sj020sg

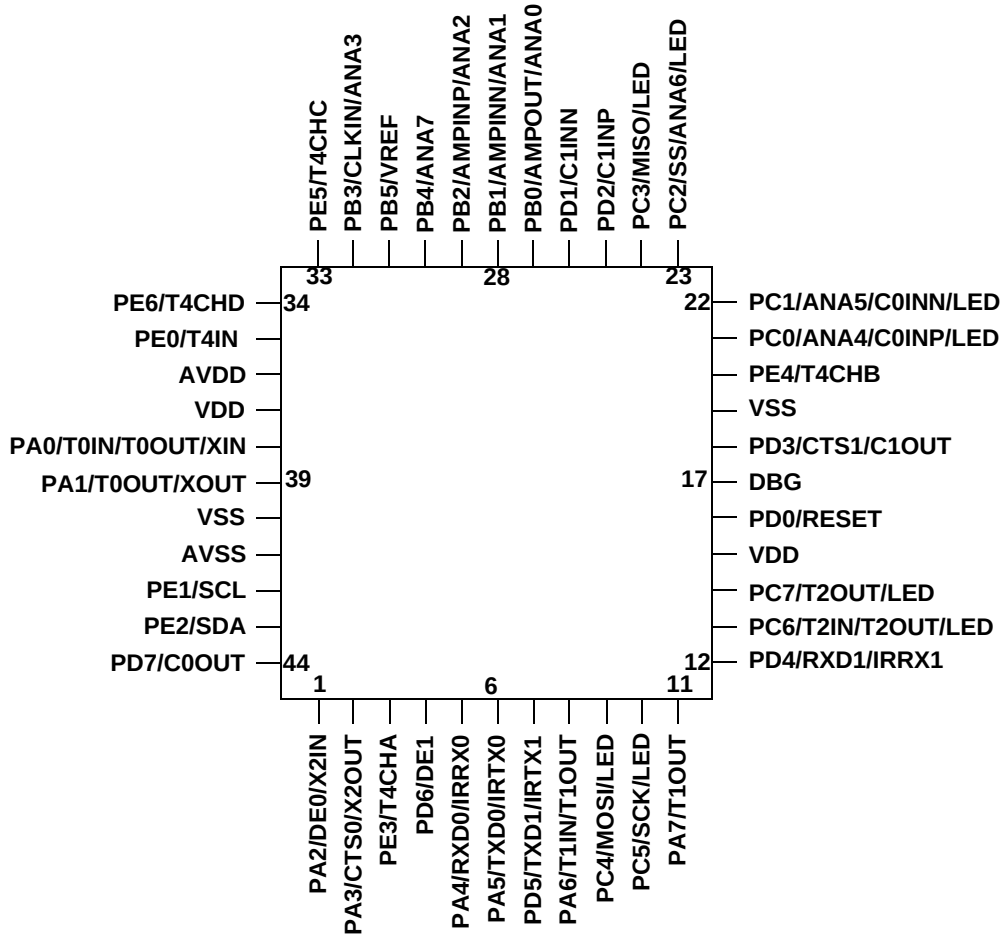


Figure 5. Z8F2480, Z8F1680 and Z8F0880 in 44-Pin Low-Profile Quad Flat Package (LQFP) or Quad Flat No Lead (QFN)

Table 6. F1680 Series MCU Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
Z8F1680 Device	
0000–0001	**Flash option bits**
0002–0003	**Reset vector**
0004–0005	**WDT interrupt vector**
0006–0007	**Illegal instruction trap**
0008–0037	**Interrupt vectors***
0038–003D	**Oscillator fail traps***
003E–3FFF	**Program Flash**
E000–E3FF	**1KB PRAM**
Z8F0880 Device	
0000–0001	**Flash option bits**
0002–0003	**Reset vector**
0004–0005	**WDT interrupt vector**
0006–0007	**Illegal instruction trap**
0008–0037	**Interrupt vectors***
0038–003D	**Oscillator fail traps***
003E–1FFF	**Program Flash**
E000–E3FF	**1KB PRAM**

Note: *See Table 36 on page 69 for a list of interrupt vectors and traps.

3.3. Data Memory

The F1680 Series MCU does not use the eZ8 CPU's 64KB Data Memory address space.

3.4. Flash Information Area

Table 7 describes the F1680 Series MCU Flash Information Area. This 512-byte Information Area is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash Information Area is mapped into the Program Memory and overlays the 512bytes at addresses FE00H to FFFFH. When the Information Area access is enabled, all reads from these Program Memory addresses return the Information Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

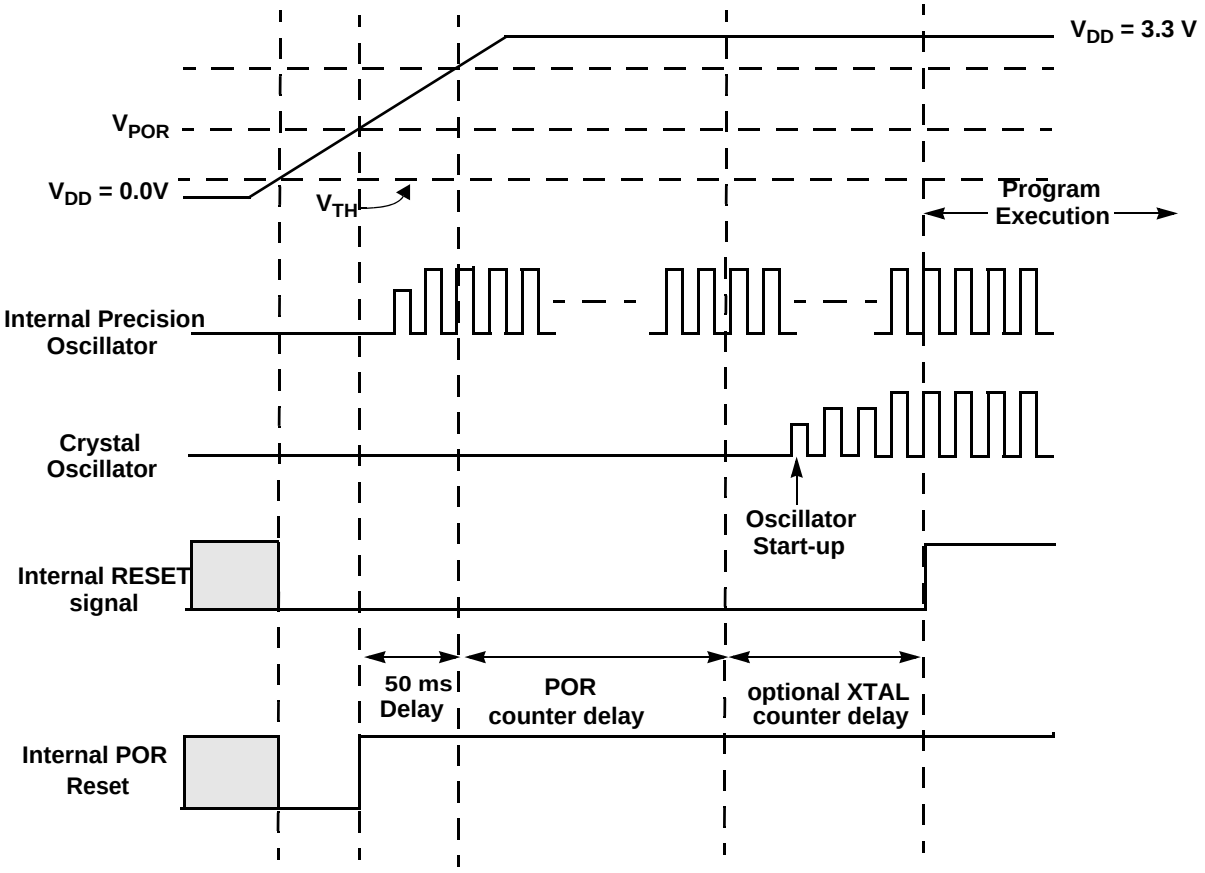
5.2.1. Power-On Reset

Each device in the Z8 Encore! XP F1680 Series contains an internal Power-On Reset (POR) circuit. The POR circuit monitors the supply voltage and holds the whole device in the Reset state until the supply voltage reaches a safe circuit operating level when the device is powered on.

After power on, the POR circuit keeps idle until the supply voltage drops below V_{TH} voltage. Figure 7 on page 35 displays this POR timing.

After the F1680 Series MCU exits the POR state, the eZ8 CPU fetches the Reset vector. Following this POR, the POR/VBO status bit in the Reset Status Register is set to 1.

For the POR threshold voltage (V_{POR}) and POR start voltage V_{TH}, see the Electrical Characteristics chapter on page 349.



Notes
1. **Not to Scale.**
2. **Internal Reset and POR Reset are active Low.**

Figure 6. Power-On Reset Operation

Bit	Description (Continued)
[4] **EXT**	External Reset Indicator **If this bit is set to 1, a Reset initiated by the external RESET pin occurs. A POR or a Stop Mode Recovery from a change in an input pin sets 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 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**

7.10. GPIO Interrupts

Many of the GPIO port pins can be used as interrupt sources. Some port pins can be configured to generate an interrupt request on either the rising edge or falling edge of the pin input signal. Other port-pin interrupt sources generate an interrupt when any edge occurs (both rising and falling). For more details about interrupts using the GPIO pins, see the Interrupt Controller chapter on page 68.

7.11. GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data and output data. Table 20 lists these port registers. Use Port A–E Address and Control registers together to provide access to subregisters for port configuration and control.

Table 20. GPIO Port Registers and Subregisters

Port Register Mnemonic	Port Register Name
P*x*ADDR	**Port A–E Address Register (Selects subregisters)**
P*x*CTL	**Port A–E Control Register (Provides access to subregisters)**
P*x*IN	**Port A–E Input Data Register**
P*x*OUT	**Port A–E Output Data Register**
Port Subregister Mnemonic	Port Register Name
P*x*DD	**Data Direction**
P*x*AF	**Alternate Function**
P*x*OC	**Output Control (Open-Drain)**
P*x*HDE	**High Drive Enable**
P*x*SMRE	**Stop Mode Recovery Source Enable**
P*x*PUE	**Pull-up Enable**
PxAFS1	**Alternate Function Set 1**
PxAFS2	**Alternate Function Set 2**

9.1. Architecture

Figure 11 displays the architecture of the timers.

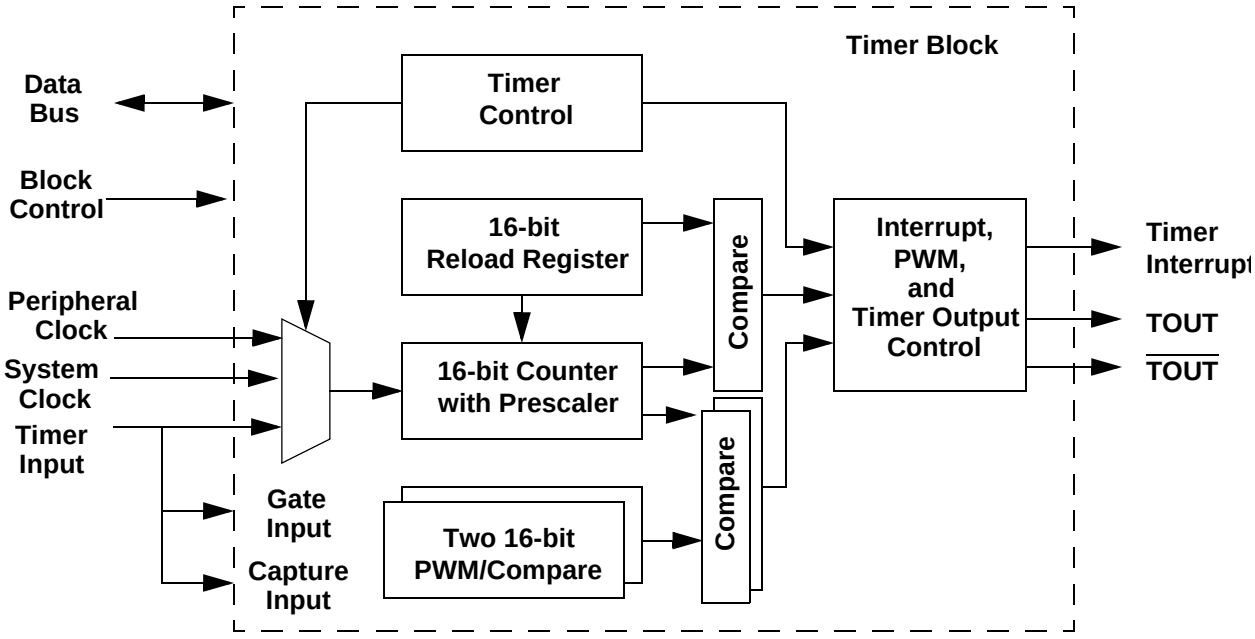


Figure 11. Timer Block Diagram

9.2. Operation

The timers are 16-bit up-counters. Minimum time-out delay is set by loading the value 0001H into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value 0000H into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the Timer reaches FFFFH, the timer rolls over to 0000H and continues counting.

9.2.1. Timer Clock Source

The timer clock source can come from either the peripheral clock or the system clock. Peripheral clock is based on a low frequency/low power 32 kHz secondary oscillator that can be used with external watch crystal. Peripheral clock source is only available for driving Timer and Noise Filter operation. It is not supported for other peripherals.

For timer operation in STOP Mode, peripheral clock must be selected as the clock source. Peripheral clock can be selected as source for both ACTIVE and STOP Mode operation.

If system clock is chosen as the clock source, the timer ceases to operate as a system clock and is put into STOP Mode. In this case, the registers are not reset and operation will resume after Stop Mode Recovery occurs.

9.2.2.3. Power Reduction During Operation

Removal of the TEN bit will inhibit clocking of the entire timer block. The CPU can still read/write registers when the enable bit(s) are taken out.

9.2.3. Timer Operating Modes

The timers can be configured to operate in the following modes, each of which is described in this section where indicated in Table 52.

Table 52. Timer Operating Modes

Mode	Page Number
TRIGGERED ONE-SHOT Mode	**88**
CONTINUOUS Mode	**90**
COUNTER Mode	**91**
COMPARATOR COUNTER Mode	**92**
PWM SINGLE OUTPUT Mode	**93**
PWM DUAL Output Mode	**95**
CAPTURE Mode	**97**
CAPTURE RESTART Mode	**98**
COMPARE Mode	**100**
GATED Mode	**100**
CAPTURE/COMPARE Mode	**102**
DEMODULATION Mode	**103**

9.2.3.1. ONE-SHOT Mode

In ONE-SHOT Mode, the timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer counts timer clocks up to the 16-bit reload value. Upon reaching the reload value, the timer generates an interrupt, and the count value in the Timer High and Low Byte registers is reset to 0001H. Then, the timer is automatically disabled and stops counting.

Additionally, if the Timer Output alternate function is enabled, the Timer Output pin changes state for one clock cycle (from Low to High or from High to Low) upon timer reload. If it is appropriate to have the Timer Output make a permanent state change on

Bit	Description (Continued)
[6] (cont'd)	PWM DUAL OUTPUT Mode **0 = Timer Output is forced Low (0) and Timer Output Complement is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon Reload. When enabled, the Timer Output Complement is forced Low (0) upon PWM count match and forced High (1) upon Reload. The PWMD field in Timer Control 0 Register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to High (1).** **1 = Timer Output is forced High (1) and Timer Output Complement is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon Reload. When enabled, the Timer Output Complement is forced High (1) upon PWM count match and forced Low (0) upon Reload. The PWMD field in Timer Control 0 Register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to Low (0).** CAPTURE RESTART 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.** COMPARATOR 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.** TRIGGERED ONE-SHOT Mode **0 = Timer counting is triggered on the rising edge of the Timer Input signal.** **1 = Timer counting is triggered on the falling edge of the Timer Input signal.** DEMODULATION Mode **0 = Timer counting is triggered on the rising edge of the Timer Input signal. The current count is captured into PWM0 High and Low byte registers on subsequent rising edges of the Timer Input signal.** **1 = Timer counting is triggered on the falling edge of the Timer Input signal. The current count is captured into PWM0 High and Low byte registers on subsequent falling edges of the Timer Input signal.** **The above functionality applies only if TPOLHI bit in Timer Control 2 Register is 0. If TPOLHI bit is 1 then timer counting is triggered on any edge of the Timer Input signal and the current count is captured on both edges. The current count is captured into PWM0 registers on rising edges and PWM1 registers on falling edges of the Timer Input signal.**

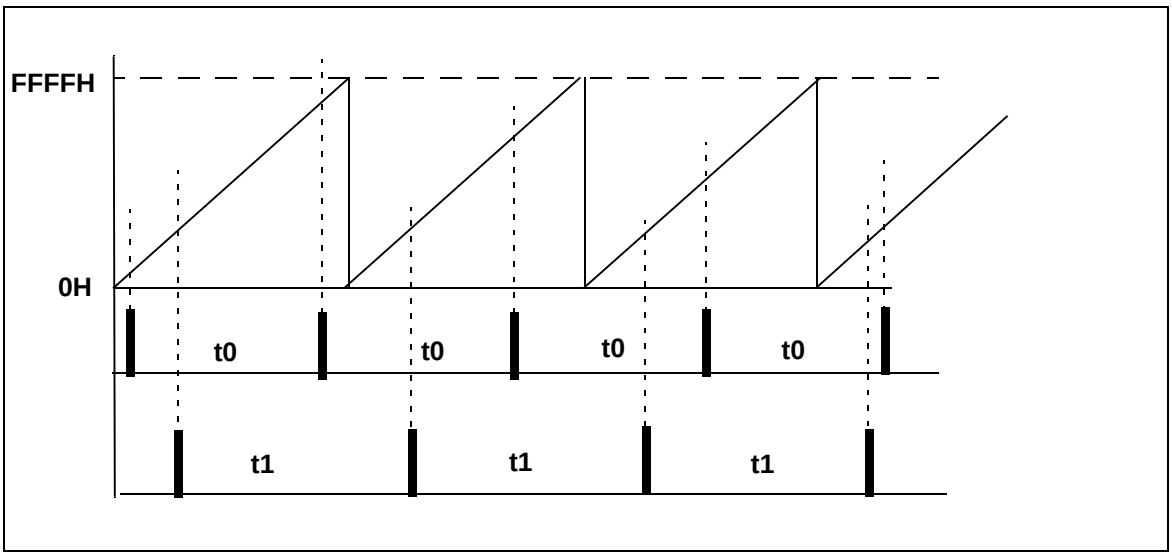


Figure 18. Count Max Mode with Channel Compare

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 Registers - 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.

10.7.8. Multi-Channel Timer Channel-y Control Registers

Each channel has a control register to enable the channel, select the input/output polarity, enable channel interrupts and select the channel mode of operation.

Table 78. Multi-Channel Timer Channel Control Register (MCTCHyCTL) [1]

Bit	7	6	5	4	3	2	1	0
Field	**CHEN**	**CHPOL**	**CHIEN**	**CHUE**	**Reserved**	**CHOP**		
Reset	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
R/W	**R/W**	**R/W**	**R/W**	**R/W**	**R**	**R/W**	**R/W**	**R/W**
Address	**See note 2.**							

Notes:
1. **y = A, B, C, D.**
2. **If 02H, 03H, 04H and 05H are in the Subaddress Register, they are accessible through Subregister 2.**

Bit	Description
[7] **CHEN**	Channel Enable **0 = Channel is disabled.** **1 = Channel is enabled.**
[6] **CHPOL**	Channel Input/Output Polarity **Operation of this bit is a function of the current operating method of the channel.** ONE-SHOT Operation **When the channel is disabled, the Channel Output signal is set to the value of this bit. When the channel is enabled, the Channel Output signal toggles for one system clock on reaching the Channel Capture/Compare Register value.** CONTINUOUS COMPARE Operation **When the channel is disabled, the Channel Output signal is set to the value of this bit. When the channel is enabled, the Channel Output signal toggles (from Low to High or High to Low) on reaching the Channel Capture/Compare Register value.** PWM OUTPUT Operation **0 = Channel Output is forced Low when the channel is disabled. When enabled, the Channel Output is forced High on Channel Capture/Compare Register value match and forced Low on reaching the Timer Reload Register value (modulo mode) or counting down through the channel Capture/Compare register value (count up/down mode).** **1 = Channel Output is forced Low when the channel is disabled. When enabled, the Channel Output is forced High on Channel Capture/Compare Register value match and forced Low on reaching the Timer Reload Register value (modulo mode) or counting down through the channel Capture/Compare register value (count up/down mode).** CAPTURE Operation **0 = Count is captured on the rising edge of the Channel Input signal.** **1 = Count is captured on the falling edge of the Channel Input signal.**

16.2. ESPI Signals

The four ESPI signals are:

- Master-In/Slave-Out (MISO)
- Master-Out/Slave-In (MOSI)
- Serial Clock (SCK)
- Slave Select ($\overline{SS}$)

The following paragraphs discuss these signals as they operate in both MASTER and SLAVE modes.

16.2.1. 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. Data is transferred most significant bit first. The MISO pin of a Slave device is placed in a high-impedance state if the Slave is not selected. When the ESPI is not enabled, this signal is in a high-impedance state. The direction of this pin is controlled by the MMEN bit of the ESPI Control Register.

16.2.2. 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. Data is transferred most significant bit first. When the ESPI is not enabled, this signal is in a high-impedance state. The direction of this pin is controlled by the MMEN bit of the ESPI Control Register.

16.2.3. Serial Clock

The Serial Clock (SCK) synchronizes data movement both in and out of the Shift Register via the MOSI and MISO pins. In MASTER Mode (MMEN = 1), the ESPI s Baud Rate Generator creates the serial clock and drives it out on its SCK pin to the slave devices. In SLAVE Mode, the SCK pin is an input. Slave devices ignore the SCK signal, unless their $\overline{SS}$ pin is asserted.

The Master and Slave are each capable of exchanging a character of data during a sequence of NUMBITS clock cycles (see Table 112 on page 217). In both Master and Slave ESPI devices, data is shifted on one edge of the SCK and is sampled on the opposite edge where data is stable. SCK phase and polarity is determined by the PHASE and CLKPOL bits in the ESPI Control Register.

16.2.4. Slave Select

The Slave Select signal is a bidirectional framing signal with several modes of operation to support SPI and other synchronous serial interface protocols. The Slave Select mode is selected by the SSMD field of the ESPI Mode Register. The direction of the SS signal is controlled by the SSIO bit of the ESPI Mode Register. The SS signal is an input on slave devices and is an output on the active Master device. Slave devices ignore transactions on the bus unless their Slave Select input is asserted. In SPI MASTER Mode, additional GPIO pins are required to provide Slave Select if there is more than one slave device.

16.3. Operation

During a transfer, data is sent and received simultaneously by both the Master and Slave devices. Separate signals are required for transmit data, receive data and the serial clock. When a transfer occurs, a multi-bit (typically 8-bit) character is shifted out one data pin and a multi-bit character is simultaneously shifted in on second data pin. An 8-bit shift register in the Master and an 8-bit shift register in the Slave are connected as a circular buffer. The ESPI Shift Register is buffered to support back-to-back character transfers in high-performance applications.

A transaction is initiated when the Data Register is written in the Master device. The value from the Data Register is transferred into the Shift Register and the SPI transaction begins. At the end of each character transfer, if the next transmit value has been written to the Data Register, the data and shift register values are swapped, which places the new transmit data into the Shift Register and the Shift Register contents (receive data) into the Data Register. At that point the Receive Data Register Not Empty signal is asserted (RDRNE bit set in the Status Register). After software reads the receive data from the Data Register, the Transmit Data Register Empty signal is asserted (TDRE bit set in the Status Register) to request the next transmit byte. To support back-to-back transfers without an intervening pause, the receive and transmit interrupts must be serviced when the current character is being transferred.

The Master sources the Serial Clock (SCK) and Slave Select signal (SS) during the transfer.

Internal data movement (by software) to/from the ESPI block is controlled by the Transmit Data Register Empty (TDRE) and Receive Data Register Not Empty (RDRNE) signals. These signals are read-only bits in the ESPI Status Register. When either the TDRE or RDRNE bits assert, an interrupt is sent to the interrupt controller. In many cases the software application is only moving information in one direction. In this case either the TDRE or RDRNE interrupts can be disabled to minimize software overhead. Unidirectional data transfer is supported by setting the ESPIEN1,0 bits in the Control Register to 10 or 01.

transmitted, the hardware will automatically de-assert the SSV and TEOF bits. The second method is for software to explicitly clear the SSV bit after the transaction completes. If software clears the SSV bit during the last byte, 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.

$\overline{SS}$ pin on the selected slave. Then, the active Master drives the clock and transmits data on the SCK and MOSI pins to the SCK and MOSI pins on the Slave (including those Slaves which are not enabled). The enabled slave device drives data out its MISO pin to the MISO Master pin.

When the ESPI is configured as a Master in a Multi-Master SPI system, the $\overline{SS}$ pin must be configured as an input. The $\overline{SS}$ input signal on a device configured as a Master should remain High. If the $\overline{SS}$ signal on the active Master goes Low (indicating another Master is accessing this device as a Slave), a Collision error flag is set in the ESPI Status Register. The Slave select outputs on a Master in a Multi-Master system must come from GPIO pins.

16.3.4.3. SPI Slave Operation

The ESPI block is configured for SLAVE Mode operation by setting the MMEN bit = 0 in the ESPICTL register and setting the SSIO bit = 0 in the ESPIMODE register. The SSMD field of the ESPI Mode Register is set to 00 for SPI protocol mode. The PHASE, CLKPOL and WOR bits in the ESPICTL register and the NUMBITS field in the ESPIMODE register must be set to be consistent with the other SPI devices. Typically for an SPI Slave, SSPO = 0.

If the Slave has data to send to the Master, the data must be written to the Data Register before the transaction starts (first edge of SCK when $\overline{SS}$ is asserted). If the Data Register is not written prior to the Slave transaction, the MISO pin outputs all 1s.

Due to the delay resulting from synchronization of the $\overline{SS}$ and SCK input signals to the internal system clock, the maximum SCK baud rate that can be supported in SLAVE Mode is the system clock frequency divided by 4. This rate is controlled by the SPI Master. Figure 41 displays the ESPI configuration in SPI SLAVE Mode.

4. If this operation is a single-byte transfer, the software asserts the NAK bit of the I2C Control Register so that after the first byte of data has been read by the I2C controller, a Not Acknowledge instruction is sent to the I2C slave.
5. The I2C controller sends a start condition.
6. The I2C controller sends the address and Read bit out via the SDA signal.
7. The I2C slave acknowledges the address by pulling the SDA signal Low during the next High period of SCL.

   If the slave does not acknowledge the address byte, the I2C controller sets the NCKI bit in the I2C Status Register, sets the ACKV bit and clears the ACK bit in the I2C 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 a stop condition on the bus and clears the stop and NCKI bits. The transaction is complete and the following steps can be ignored.
8. The I2C controller shifts in the first byte of data from the I2C slave on the SDA signal.
9. The I2C controller asserts the receive interrupt.
10. 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.
11. The I2C controller sends a Not Acknowledge to the I2C slave if the next byte is the final byte; otherwise, it sends an Acknowledge.
12. If there are more bytes to transfer, the I2C controller returns to Step 7
13. A NAK interrupt (NCKI bit in I2CISTAT) is generated by the I2C controller.
14. The software responds by setting the stop bit of the I2C Control Register.
15. A stop condition is sent to the I2C slave.

17.2.5.7. Master Read Transaction with a 10-Bit Address

Figure 46 displays the read transaction format for a 10-bit addressed Slave.

S	Slave Address 1st Byte	W=0	A	Slave Address 2nd Byte	A	S	Slave Address 1st Byte	R=1	A	Data	A	Data	$\overline{A}$	P

Figure 46. Data Transfer Format—Master Read Transaction with a 10-Bit Address

The first 7 bits transmitted in the first byte are 11110XX. The two XX bits are the two most-significant bits of the 10-bit address. The last bit of the first byte transferred is the write control bit.

Observe the following data transfer procedure for a Read operation to a 10-bit addressed slave:

from the Shift Register after it is received from the I2C bus. The I2C Shift Register is not accessible in the Register File address space; data is used only to buffer incoming and outgoing data.

Writes by the software to the I2CDATA Register are blocked if a slave Write transaction is underway (the I2C controller is in SLAVE Mode and data is being received).

Table 119. I2C Data Register (I2CDATA = F50H)

Bits	7	6	5	4	3	2	1	0
Field	**Data 7**	**Data 6**	**Data 5**	**Data 4**	**Data 3**	**Data 2**	**Data 1**	**Data 0**
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	**F50H**							

Bit Position	Value	Description
[7:0] DATA	**—**	**I2C Data Byte**

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.**

Figures 69 through 72 display the typical current consumption at voltages of 1.8V, 2.0V, 2.7V, 3.0V, 3.3V and 3.6V, respectively, versus different system clock frequencies while operating at a temperature of 25”C.

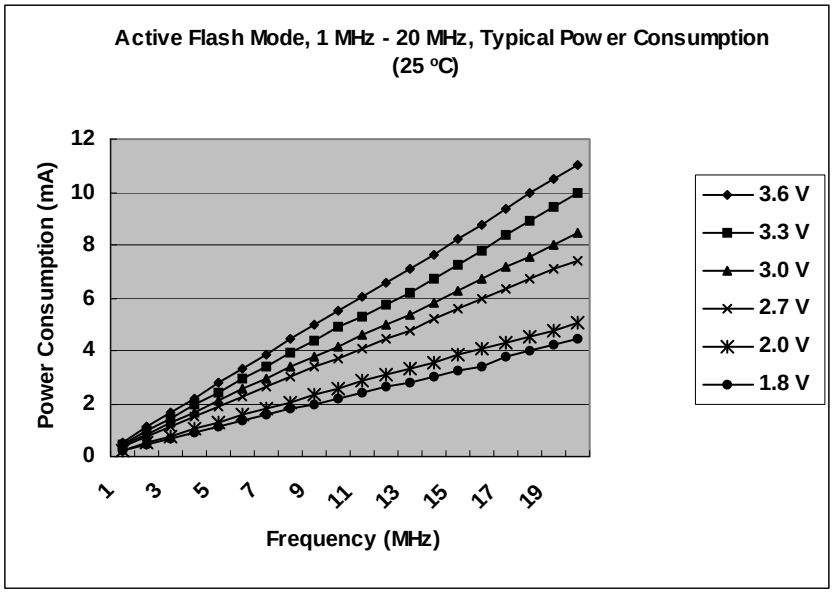


Figure 69. Typical Active Flash Mode Supply Current (1–20MHz)

Table 200. IPO Electrical Characteristics (Continued)

		V_{DD} = 1.8 to 3.6V T_A = –40°C to +105°C			V_{DD} = 2.7 to 3.6V T_A = 0°C to +70°C				
Symbol	Parameter	Min	Typ	Max	Min	Typ	Max	Units	Conditions
	Output Frequency	**10.6168**	**11.0592**	**11.5016**	**10.78272**	**11.0592**	**11.33568**		
	Divided-by-2 Output Frequency	**5.3084**	**5.5296**	**5.7508**	**5.39136**	**5.5296**	**5.66784**		
	Divided-by-4 Output Frequency	**2.6542**	**2.7648**	**2.8754**	**2.69568**	**2.7648**	**2.83392**		**±2.5% 2.7 to 3.6 V, 0–70°C;**
	Divided-by-8 Output Frequency	**1.3271**	**1.3824**	**1.4377**	**1.34784**	**1.3824**	**1.41696**		
F_{IPO}	**Divided-by-16 Output Frequency**	**0.6636**	**0.6912**	**0.7188**	**0.67392**	**0.6912**	**0.70848**	**MHz**	**±4% 1.8 to 2.7 V, 0–70°C**
	Divided-by-32 Output Frequency	**0.3318**	**0.3456**	**0.3594**	**0.33696**	**0.3456**	**0.35424**		**±4% 1.8 to 3.6 V, –40–105°C**
	Divided-by-128 Output Frequency	**0.0829**	**0.0864**	**0.0899**	**0.08424**	**0.0864**	**0.08856**		
	Divided-by-256 Output Frequency	**0.0415**	**0.0432**	**0.0449**	**0.04212**	**0.0432**	**0.04428**		
	Duty Cycle of Output	**45**		**55**	**45**		**55**	**%**	

Table 201. Low Voltage Detect Electrical Characteristics

		T_A = 0°C to +70°C T_A = –40°C to +105°C V_{DD} = 1.8 to 3.6 V				
Symbol	Parameter	Min	Typ	Max	Units	Conditions
I_{DD}^{LVD}	**LVD Active Current**	**–**	**–**	**50**	**µA**	
I_{DDQ}^{LVD}	**LVD Quiescent Current**	**–**	**5**	**–**	**nA**	
V_{TH}	**Detected Source Voltage**	**V_{TP} – 10%**	**V_{TP}[1]**	**V_{TP} + 10%**	**V**	

Table 201. Low Voltage Detect Electrical Characteristics (Continued)

		T_A = 0°C to +70°C T_A = –40°C to +105°C V_{DD} = 1.8 to 3.6 V				
Symbol	Parameter	Min	Typ	Max	Units	Conditions
V_{TH_PRO}	**Detected Source Voltage for Flash Protection**	**2.4**	**2.5**	**2.6**	**V**	
T_{DELAY}	**Delay from source voltage falling lower than V_{TP} to I_{VD_OUT} output logic High**	**50**	**1000**	**–**	**ns**	
Note: [1] V_{TP} is a user-set threshold voltage to be detected.						

Table 202. Crystal Oscillator Characteristics

		T_A = 0°C to +70°C T_A = –40°C to +105°C							
		V_{DD} = 2.7 to 3.6V			V_{DD} = 1.8 to 2.7V				
Symbol	Parameter	Min	Typ	Max	Min	Typ	Max	Units	Conditions
I_{DD}XTAL	**Crystal Oscillator Active Supply Current**	**–**	**–**	**500**	**–**	**–**	**300**	**µA**	
I_{DDQ}XTAL	**Crystal Oscillator Quiescent Current**	**–**	**5**	**–**	**–**	**5**	**–**	**nA**	
S_{CLK}	**Clk_out State in Crystal Disable**	**1**	**1**	**1**	**1**	**1**	**1**		
F_{XTAL}	**External Crystal Oscillator Frequency**	**1**	**–**	**20**	**1**	**–**	**20**	**MHz**	**See Figure 74.**
T_{SET}	**Startup Time After Enable**	**–**	**10,000**	**30,000**	**–**	**10,000**	**30,000**	**Cycle**	
	Clk_out Duty Cycle	**40**	**50**	**60**	**40**	**50**	**60**	**%**	
	Clk_out Jitter	**–**	**1**	**–**	**–**	**1**	**–**	**%**	