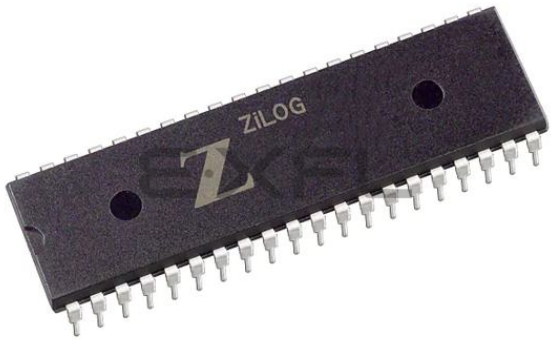


Welcome to E-XFL.COM

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	33
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	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.620", 15.75mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1680pm020eg

Table of Contents

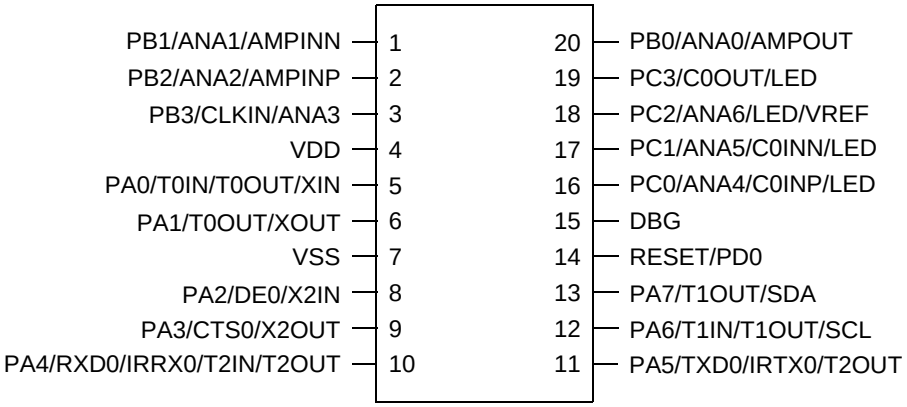


Figure 2. Z8F2480, Z8F1680 and Z8F0880 in 20-Pin SOIC, SSOP or PDIP Packages

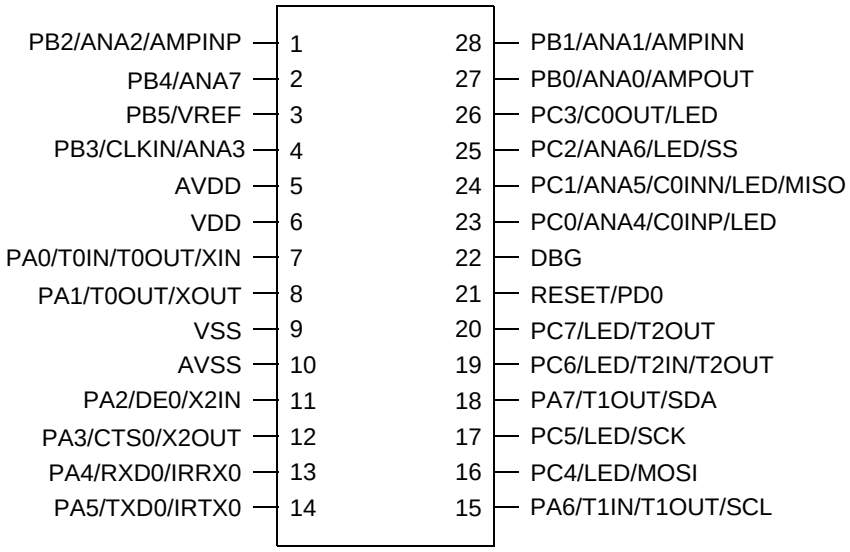


Figure 3. Z8F2480, Z8F1680 and Z8F0880 in 28-Pin SOIC, SSOP or PDIP Packages

Table 48. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bits	7	6	5	4	3	2	1	0
Field	Reserved	MCTENL	U1RENL	U1TENL	C3ENL	C2ENL	C1ENL	C0ENL
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	FC8H							

Bit	Description
[7]	Reserved; must be 0.
[6] MCTENL	Multi-Channel Timer Interrupt Request Enable Low Bit (MCTENL)
[5] U1RENL	UART1 Receive Interrupt Request Enable Low Bit (U1RENL)
[4] U1TENL	UART1 Transmit Interrupt Request Enable Low Bit (U1TENL)
[3] C3ENL	Port C3 Interrupt Request Enable Low Bit (C3ENL)
[2] C2ENL	Port C2 Interrupt Request Enable Low Bit (C2ENL)
[1] C1ENL	Port C1 Interrupt Request Enable Low Bit (C1ENL)
[0] C0ENL	Port C0 Interrupt Request Enable Low Bit (C0ENL)

3. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
4. Write to the Timer High and Low Byte registers to set the starting count value. This value only affects the first pass in COUNTER Mode. After the first timer reload in COUNTER Mode, counting always begins at the reset value of 0001H. Generally, in COUNTER Mode the Timer High and Low Byte registers must be written with the value 0001H.
5. Write to the Timer Reload High and Low Byte registers to set the reload value.
6. If required, 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 Input alternate function.
8. When using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
9. Write to the Timer Control 1 Register to enable the timer.

In COUNTER Mode, the number of Timer Input transitions since the timer start is calculated using the following equation:

COUNTER Mode Timer Input Transitions = Current Count Value - Start Value

9.2.3.5. COMPARATOR COUNTER Mode

In COMPARATOR COUNTER Mode, the timer counts output transitions from an analog comparator output. The assignment of a comparator to a timer is based on the TIMTRG bits in the CMP0 and CMP1 registers. The TPOL bit in the Timer Control 1 Register selects whether the count occurs on the rising edge or the falling edge of the comparator output signal. In COMPARATOR COUNTER Mode, the prescaler is disabled.

! Caution: The frequency of the comparator output signal must not exceed one-fourth the timer clock frequency.

Upon reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or High to Low) at timer reload.

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

4. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
5. Write to the Timer Reload High and Low Byte registers to set the reload value.
6. Clear the Timer PWM High and Low Byte registers to 0000H. This allows user software to determine if interrupts were generated by either a capture event or a reload. If the PWM High and Low Byte registers still contain 0000H after the interrupt, then the interrupt was generated by a Reload.
7. If required, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt will be generated for both input capture and reload events. If required, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the Timer Control 0 Register.
8. Configure the associated GPIO port pin for the Timer Input alternate function.
9. Write to the Timer Control 1 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:

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

9.2.3.9. CAPTURE RESTART Mode

In CAPTURE RESTART 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 counts timer clocks up to the 16-bit reload value. The TPOL bit in the Timer Control 1 Register determines if the Capture occurs on a rising edge or a falling edge of the Timer Input signal. When the Capture event occurs, an interrupt is generated and the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in Timer Control 0 Register is set to indicate the timer interrupt is due to an input capture event.

If no Capture event occurs, 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 INPCAP bit in Timer Control 0 Register is cleared to indicate the timer interrupt is not due to an input capture event.

Bit	Description (Continued)
[5:3] PRES	**Prescale Value** The timer input clock is divided by 2PRES, where PRES can be set from 0 to 7. The prescaler is reset each time the Timer is disabled. This insures proper clock division each time the Timer is restarted. 000 = Divide by 1 001 = Divide by 2 010 = Divide by 4 011 = Divide by 8 100 = Divide by 16 101 = Divide by 32 110 = Divide by 64 111 = Divide by 128
[2:0] TMODE[2:0]	**Timer Mode** This field, along with the TMODE[3] bit in the TxCTL0 Register, determines the operating mode of the timer. TMODE[3:0] selects among the following modes: 0000 = ONE-SHOT Mode 0001 = CONTINUOUS Mode 0010 = COUNTER Mode 0011 = PWM SINGLE OUTPUT Mode 0100 = CAPTURE Mode 0101 = COMPARE Mode 0110 = GATED Mode 0111 = CAPTURE/COMPARE Mode 1000 = PWM DUAL OUTPUT Mode 1001 = CAPTURE RESTART Mode 1010 = COMPARATOR COUNTER Mode 1011 = TRIGGERED ONE-SHOT Mode 1100 = DEMODULATION Mode

9.3.5.3. Timer 0–2 Control 2 Register

The Timer 0–2 Control 2 (TxCTL2) registers allow selection of timer clock source and control of timer input polarity in DEMODULATION Mode. See Table 65.

Table 65. Timer 0–2 Control 2 Register (TxCTL2)

Bit	7	6	5	4	3	2	1	0
Field	Reserved		PWM0UE	TPOLHI	Reserved			TCLKS*
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	F22H, F26H, F2AH							

Bit	Description
[7:6]	Reserved; must be 0.
[5] PWM0UE	**PWM0 Update Enable** This bit determines whether writes to the PWM0 High and Low Byte registers are buffered when TEN = 1. Writes to these registers are not buffered when TEN = 0, regardless of the value of this bit. 0 = Writes to the Channel High and Low Byte registers are buffered when TEN = 1 and only take affect on a timer reload to 0001H. 1 = Writes to the Channel High and Low Byte registers are not buffered when TEN = 1.
[4] TPOLHI	**Timer Input/Output Polarity High Bit** This bit determines if timer count is triggered and captured on both edges of the input signal. This applies only to DEMODULATION Mode. 0 = Count is captured only on one edge in DEMODULATION Mode. In this case, edge polarity is determined by TPOL bit in the TxCTL1 Register. 1 = Count is triggered on any edge and captured on both rising and falling edges of the Timer Input signal in DEMODULATION Mode.
[3:1]	Reserved; must be 0.
[0] TCLKS	**Timer Clock Source** 0 = System Clock. 1 = Peripheral Clock.*

Note: *Before selecting the peripheral clock as the timer clock source, the peripheral clock must be enabled and oscillating.

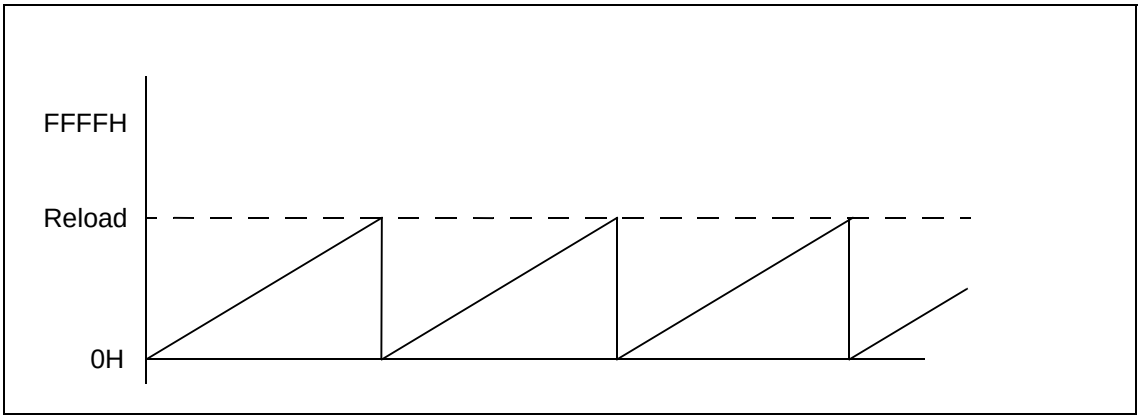


Figure 15. Count Modulo Mode

10.2.7. Count Up/Down Mode

In the Count Up/Down mode, the timer counts up to the Reload Register value and then counts down to 0000H. As shown in Figures 16, the counting cycle continues with twice the reload value as the period. A timer count interrupt is generated when the timer count decrements to zero.

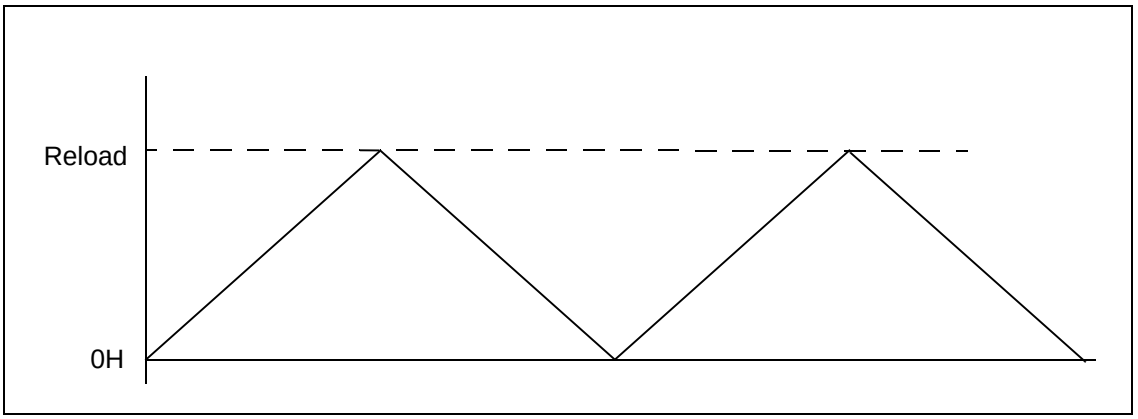


Figure 16. Count Up/Down Mode

rate. To avoid an autobaud overrun error, the system clock must not be greater than 2^{19} times the baud rate (16 bit counter following 3-bit prescaler when counting the 8 bit times of the Autobaud sequence).

Following the Synch character, the LIN-UART hardware transits to the Active state, in which the identifier character is received and the characters of the response section of the message are sent or received. The slave remains in this Active state until a break is received or software forces a state change. After it is in an Active state (i.e., autobaud has completed), a break of 10 or more bit times is recognized and causes a transition to the Autobaud state.

If the identifier character indicates that this slave device is not participating in the message, software sets the LinState[1:0] = 01b (Wait for Break state) to ignore the rest of the message. No further receive interrupts will occur until the next break.

12.1.11. LIN-UART Interrupts

The LIN-UART features separate interrupts for the transmitter and receiver. In addition, when the LIN-UART primary functionality is disabled, the Baud Rate Generator can also function as a basic timer with interrupt capability.

12.1.11.1. Transmitter Interrupts

The transmitter generates a single interrupt when the Transmit Data Register Empty bit (TDRE) is set to 1. This indicates that the transmitter is ready to accept new data for transmission. The TDRE interrupt occurs when the transmitter is initially enabled and after the Transmit Shift Register has shifted out the first bit of a character. At this point, the Transmit Data Register can be written with the next character to send. This provides 7 bit periods of latency to load the Transmit Data Register before the Transmit Shift Register completes shifting the current character. Writing to the LIN-UART Transmit Data Register clears the TDRE bit to 0.

12.1.11.2. Receiver Interrupts

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

- A data byte has been received and is available in the LIN-UART Receive Data Register. This interrupt can be disabled independent of the other receiver interrupt sources via the RDAIRQ bit (this feature is useful in devices which support DMA). The received data interrupt occurs after the receive character has been placed in the Receive Data Register. Software must respond to this received data available condition before the next character is completely received to avoid an overrun error.

14.3.2. ADC Raw Data High Byte Register

The ADC Raw Data High Byte Register, shown in Table 102, contains the upper 8 bits of raw data from the ADC output. Access to the ADC Raw Data High Byte register is read-only. This register is used for test only.

Table 102. ADC Raw Data High Byte Register (ADCRD_H)

Bits	7	6	5	4	3	2	1	0
Field	ADCRDH							
Reset	X							
R/W	R							
Address	F71H							

Bit Position	Value (H)	Description
[7:0]	00–FF	**ADC Raw Data High Byte** The data in this register is the raw data coming from the SAR Block. It will change as the conversion is in progress. This register is used for testing only.

14.3.3. ADC Data High Byte Register

The ADC Data High Byte Register, shown in Table 103, contains the upper eight bits of the ADC output. Access to the ADC Data High Byte Register is read-only. Reading the ADC Data High Byte Register latches data in the ADC Low Bits Register.

Table 103. ADC Data High Byte Register (ADCD_H)

Bits	7	6	5	4	3	2	1	0
Field	ADCDH							
Reset	X							
R/W	R							
Address	F72H							

Bit Position	Value (H)	Description
[7:0]	00–FF	**ADC High Byte** The last conversion output is held in the data registers until the next ADC conversion has completed.

14.3.5. Sample Settling Time Register

The Sample Settling Time Register, shown in Table 105, is used to program the length of time from the SAMPLE/HOLD signal to the start signal, when the conversion can begin. The number of clock cycles required for settling will vary from system to system depending on the system clock period used. The system designer should program this register to contain the number of clocks required to meet a 0.5µs minimum settling time.

Table 105. Sample Settling Time (ADCSST)

Bits	7	6	5	4	3	2	1	0
Field	Reserved				SST			
Reset	0				1	1	1	1
R/W	R				R/W			
Address	F74H							

Bit Position	Value (H)	Description
[7:4]	0	Reserved; must be 0.
[3:0] SST	0–F	Sample settling time in number of system clock periods to meet 0.5µs minimum.

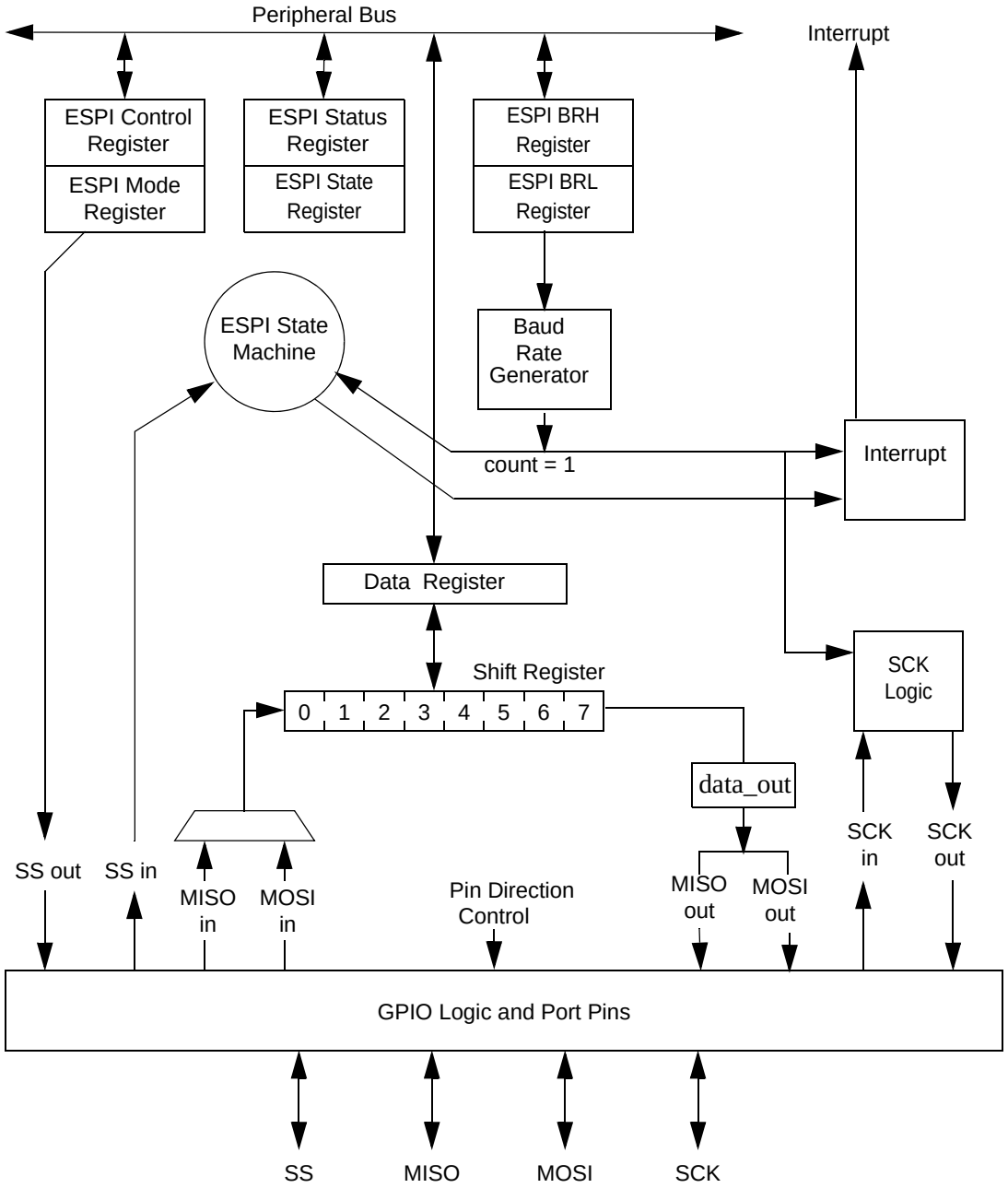


Figure 33. ESPI Block Diagram

be configured with MMEN = 1, SSIO = 0 ($\overline{SS}$ is an input) and $\overline{SS}$ input = 0. A mode fault sets the COL bit in the ESPI Status Register to 1. Writing a 1 to COL clears this error flag.

16.3.5.3. Receive Overrun

A receive overrun error occurs when a transfer completes and the RDRNE bit is still set from the previous transfer. A receive overrun sets the ROVR bit in the ESPI Status Register to 1. Writing a 1 to ROVR clears this error flag. The Receive Data Register is not overwritten and will contain the data from the transfer which initially set the RDRNE bit. Subsequent received data is lost until the RDRNE bit is cleared.

In SPI MASTER Mode, a receive overrun will not occur. Instead, the SCK will be paused until software responds to the previous RDRNE/TDRE requests.

16.3.5.4. SLAVE Mode Abort

In SLAVE Mode, if the $\overline{SS}$ pin deasserts before all bits in a character have been transferred, the transaction is aborted. When this condition occurs the ABT bit is set in the ESPI Status Register. A Slave abort error resets the Slave control logic to idle state.

A Slave abort error is also asserted in SLAVE Mode, if BRGCTL = 1 and a baud rate generator time-out occurs. When BRGCTL = 1 in SLAVE Mode, the baud rate generator functions as a Watchdog Timer monitoring the SCK signal. The BRG counter is reloaded every time a transition on SCK occurs while $\overline{SS}$ is asserted. The Baud Rate Reload registers must be programmed with a value longer than the expected time between the $\overline{SS}$ assertion and the first SCK edge, between SCK transitions while $\overline{SS}$ is asserted and between the last SCK edge and $\overline{SS}$ deassertion. A time-out indicates the Master is stalled or disabled. Writing a 1 to ABT clears this error flag.

16.3.6. ESPI Interrupts

ESPI has a single interrupt output which is asserted when any of the TDRE, TUND, COL, ABT, ROVR or RDRNE bits are set in the ESPI Status Register. The interrupt is a pulse which is generated when any one of the source bits initially sets. The TDRE and RDRNE interrupts can be enabled/disabled via the Data Interrupt Request Enable (DIRQE) bit of the ESPI Control Register.

A transmit interrupt is asserted by the TDRE status bit when the ESPI block is enabled and the DIRQE bit is set. The TDRE bit in the Status register is cleared automatically when the Data Register is written or the ESPI block is disabled. After the Data Register is loaded into the Shift Register to start a new transfer, the TDRE bit will be set again, causing a new transmit interrupt. In SLAVE Mode, if information is being received but not transmitted the transmit interrupts can be eliminated by selecting Receive Only mode (ESPIEN1,0 = 01). A Master cannot operate in Receive Only mode since a write to the ESPI (Transmit) Data Register is still required to initiate the transfer of a character even if information is being received but not transmitted by the software application.

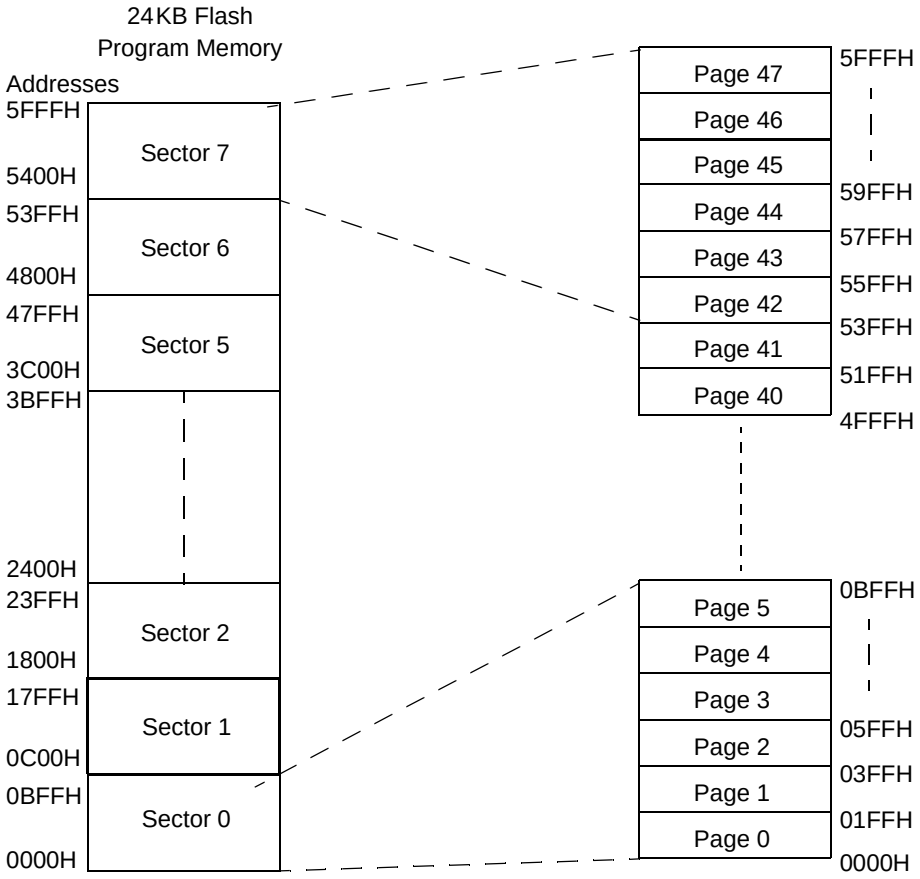


Figure 53. 24KB Flash Memory Arrangement

20.2. Operation

The Flash Controller programs and erases Flash memory. The Flash Controller provides the proper Flash controls and timing for byte programming, Page Erase and Mass Erase of Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanisms operate on the page, sector and full-memory levels.

The Flow Chart in Figure 54 displays basic Flash Controller operation. The sections that follow provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select Page Erase and Mass Erase) shown in Figure 54.

Table 162. OCD Baud-Rate Limits

System Clock Frequency	Maximum Asynchronous Baud Rate (bits/s)	Minimum Baud Rate (bits/s)
20.0 MHz	2.5 M	39.1 k
1.0MHz	125 k	1.96K
32kHz	4096	64

If the OCD receives a Serial Break (ten or more continuous bits Low), the Autobaud Detector/Generator resets. The Autobaud Detector/Generator can then be reconfigured by sending 80H. If the Autobaud Detector overflows while measuring the Autobaud character, the Autobaud Detector will remain reset.

23.2.4. High Speed Synchronous

It is possible to operate the serial On-Chip Debugger at high speeds. To operate at high speeds, data must be synchronized with an external clock. High speed synchronous communication will only work when using an external clock source. To operate in high-speed synchronous mode, simply Autobaud to the appropriate speed. The Autobaud generator will automatically run at the appropriate baud rate.

Slow bus rise times due to the pullup resistor become a limiting factor when operating at high speeds. To compensate for slow rise times, the output driver can be configured to drive the line High. If the TXD (Transmit Drive) bit is set, the line will be driven both High and Low during transmission. The line starts being driven at the beginning of the start bit and stops being driven at the middle of the stop bit. If the TXDH (Transmit Drive High) bit is set, the line will be driven High until the input is High or the center of the bit occurs, whichever is first. If both TXD and TXDH are set, the pin will be driven High for one clock period at the beginning of each 0 to 1 transition. An example of a high-speed synchronous interface is displayed in Figure 60.

```
DBG ← 03H
DBG → ~OCDCNTR[15:8]
DBG → ~OCDCNTR[7:0]
```

Write OCD Control Register (04H). The Write OCD Control Register command writes the data that follows to the OCDCTL register.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

Read OCD Control Register (05H). The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

Write Program Counter (06H). The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Read Protect Option bit is enabled, the Program Counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

Read Program Counter (07H). The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG mode or if the Read Protect option bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

Write Register (08H). The Write Register command writes data to the Register File. Data can be written 1–256 bytes at a time (256 bytes can be written by setting size to 0). If the device is not in DEBUG mode, the address and data values are discarded. If the Read Protect option bit is enabled, then only writes to the on-chip peripheral registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0,Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```

Read Register (09H). The Read Register command reads data from the Register File. Data can be read 1–256 bytes at a time (256 bytes can be read by setting size to zero). If the device is not in DEBUG mode or if the Read Protect option bit is enabled and on-chip RAM is being read from, this command returns FFH for all the data values.

```
DBG ← 09H
DBG ← {4'h0,Register Address[11:8]
DBG ← Register Address[7:0]
```

An external resistance value of 45KΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40KΩ. The typical oscillator frequency can be estimated from the values of the resistor (*R* in KΩ) and capacitor (*C* in pF) elements using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1\times10^{6}}{(0.4 \times R \times C) + (4 \times C)}$$

Figure 64 displays the typical (3.3V and 25°C) oscillator frequency as a function of the capacitor (*C* in pF) employed in the RC network assuming a 45KΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator X_{IN} pin and the printed circuit board should be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20pF are recommended.

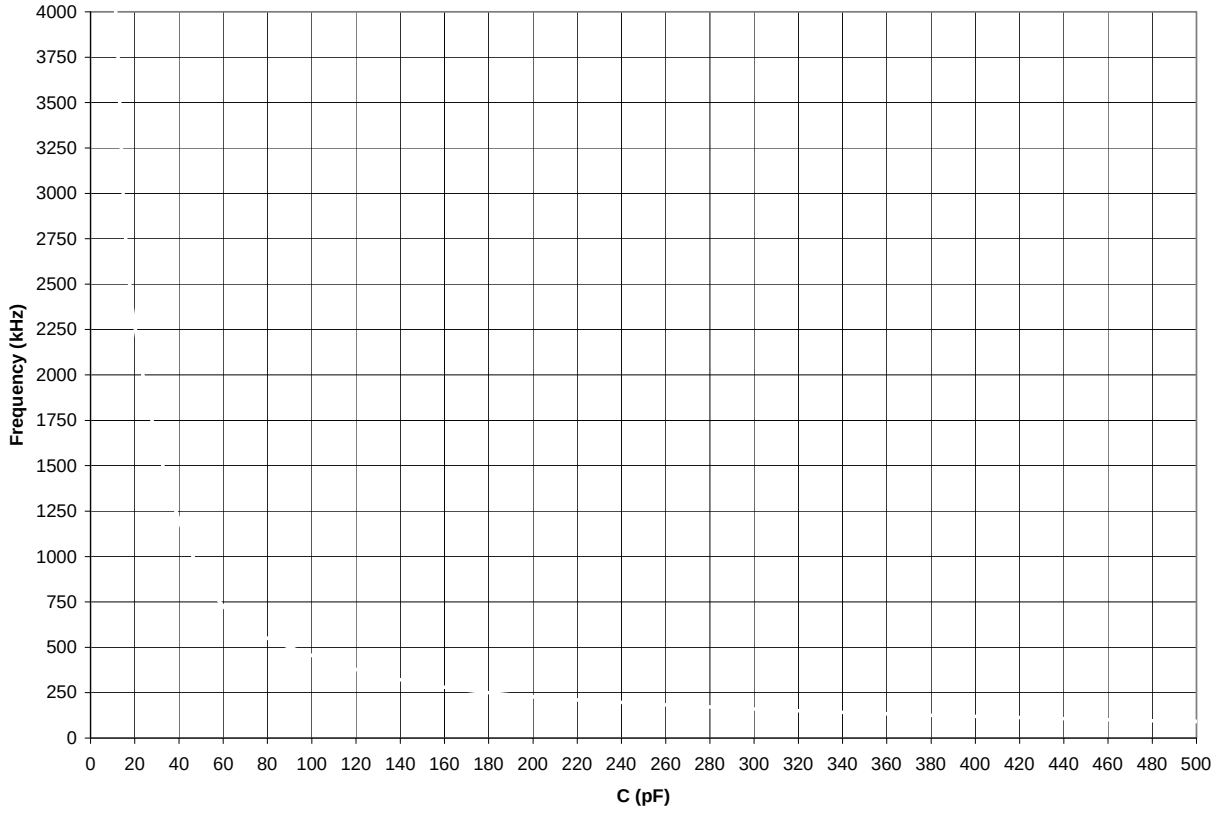
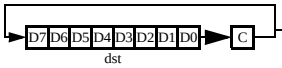


Figure 64. Typical RC Oscillator Frequency as a Function of External Capacitance

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 210. Package and Pin Count Description

	Pin Count			
Package	**20**	**28**	**40**	**44**
SOIC	√	√		
SSOP	√	√		
PDIP	√	√	√	
LQFP				√
QFN				√

31.1.0.1. Precharacterization Product

The product represented by this document is newly introduced and Zilog® has not completed the full characterization of the product. The document states what Zilog knows about this product at this time, but additional features or nonconformance with some aspects of the document might be found, either by Zilog or its customers in the course of further application and characterization work. In addition, Zilog cautions that delivery might be uncertain at times, because of start-up yield issues.

S