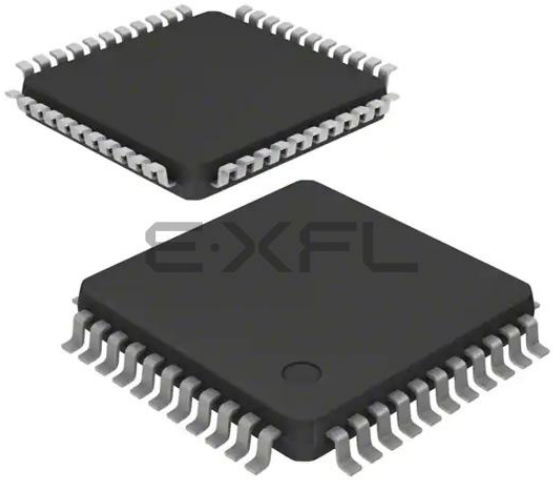


Welcome to E-XFL.COM

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	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	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1680an020sg

Table 5. Pin Characteristics (20-, 28-, 40- and 44-pin Devices) (Continued)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt Trigger Input	Open Drain Output	5V Tolerance
PE[6:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable	Yes, 5V tolerant inputs unless pull-ups are enabled
RESET/ PD0	I/O	I/O (defaults to $\overline{\text{RESET}}$)	Low (in RESET mode)	Yes (PD0 only)	Programmable for PD0; always On for $\overline{\text{RESET}}$	Yes	Programmable for PD0; always On for $\overline{\text{RESET}}$	Yes, 5V tolerant inputs unless pull-ups are enabled
V_{DD}	N/A	N/A	N/A	N/A			N/A	N/A
V_{SS}	N/A	N/A	N/A	N/A			N/A	N/A

Table 18. Port Alternate Function Mapping, 28-Pin Parts[1,2] (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port B	PB0	Reserved		AFS1[0]: 0
		ANA0/AMPOUT	ADC Analog Input/LPO Output	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1/AMPINN	ADC Analog Input/LPO Input (N)	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2/AMPINP	ADC Analog Input/LPO Input (P)	AFS1[2]: 1
	PB3	CLKIN	External Clock Input	AFS1[3]: 0
		ANA3	ADC Analog Input	AFS1[3]: 1
	PB4	Reserved		AFS1[4]: 0
		ANA7	ADC Analog Input	AFS1[4]: 1
	PB5	Reserved		AFS1[5]: 0
		VREF	Voltage Reference	AFS1[5]: 1

Notes:
1. Because there are at most two choices of alternate functions for some pins in Ports A and B, the Alternate Function Set Register (AFS2) is implemented but not used to select the function. The alternate function selection must also be enabled, as described in the Port A–E Alternate Function Subregisters section on page 61.
2. Because there is only one alternate function for each Port D and Port E pin, the Alternate Function Set registers are not implemented for Ports D and E. Enabling the alternate function selections automatically enables the associated alternate function, as described in the Port A–E Alternate Function Subregisters section on page 61.

9.2.4. Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte register is read, the contents of the Timer Low Byte Register are placed in a holding register. A subsequent read from the Timer Low Byte Register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte Register returns the actual value in the counter.

9.2.5. Timer Output Signal Operation

The Timer Output is a GPIO port pin alternate function. Generally, the Timer Output is toggled every time the counter is reloaded.

9.2.6. Timer Noise Filter

A Noise Filter circuit is included which filters noise on a Timer Input signal before the data is sampled by the block.

The Noise Filter has the following features:

- Synchronizes the receive input data to the Timer Clock
- NFEN (Noise Filter Enable) input selects whether the Noise Filter is bypassed (NFEN=0) or included (NFEN=1) in the receive data path
- NFCTL (Noise Filter Control) input selects the width of the up/down saturating counter digital filter. The available widths range from 4 bits to 11 bits
- The digital filter output has hysteresis
- Provides an active Low *saturated state* output (FiltSatB) which is used as an indication of the presence of noise
- Available for operation in STOP Mode

9.2.7. Architecture

Figure 12 displays how the Noise Filter is integrated with the Timer.

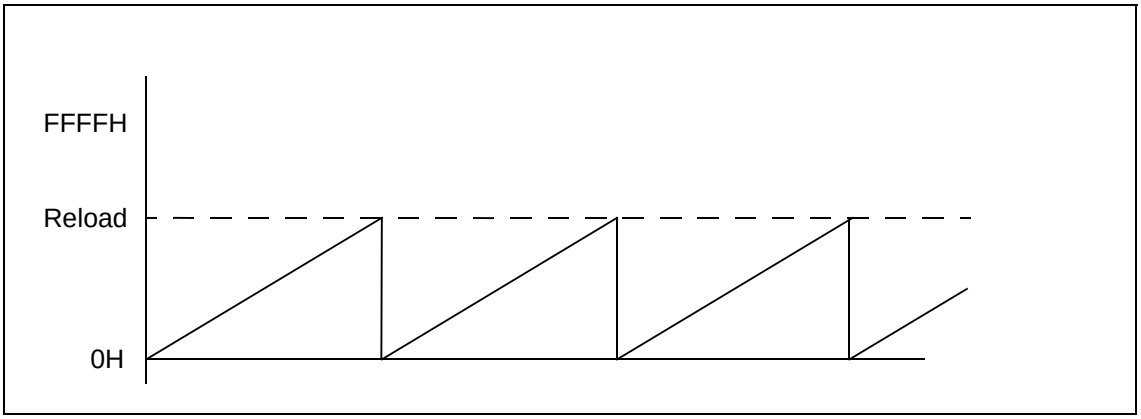


Figure 15. Count Modulo Mode

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.

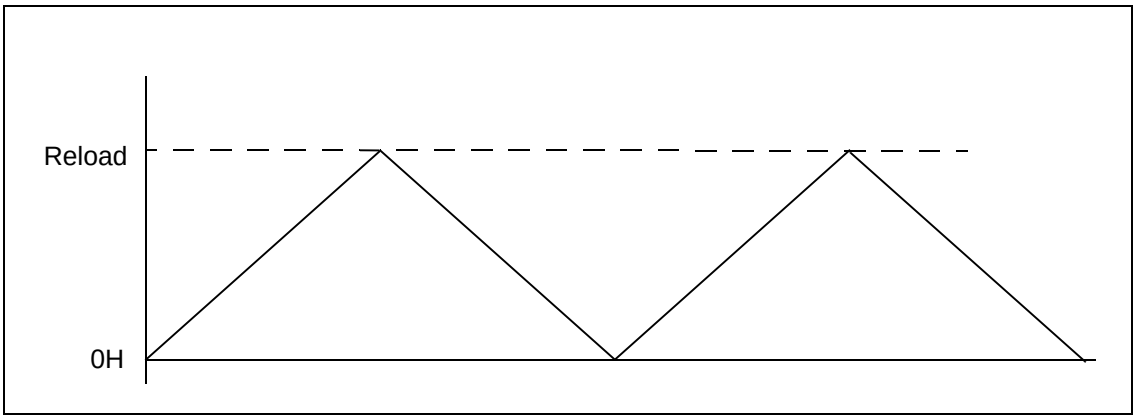


Figure 16. Count Up/Down Mode

12.1.12. LIN-UART Baud Rate Generator

The LIN-UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The LIN-UART Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data-transmission rate (baud rate) of the LIN-UART. The LIN-UART data rate for normal UART operation is calculated using the following equation:

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

The LIN-UART data rate for LIN mode UART operation is calculated using the following equation:

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

When the LIN-UART is disabled, the BRG functions as a basic 16-bit timer with interrupt on time-out. To configure the BRG as a timer with interrupt on time-out, follow the procedure below:

1. Disable the LIN-UART receiver by clearing the REN bit in the LIN-UART Control 0 Register to 0 (i.e., the TEN bit can be asserted; transmit activity can occur).
2. Load the appropriate 16-bit count value into the LIN-UART Baud Rate High and Low Byte registers.
3. Enable the BRG timer function and the associated interrupt by setting the BRGCTL bit in the LIN-UART Control 1 Register to 1.

12.2. Noise Filter

A noise filter circuit is included which filters noise on a digital input signal (such as UART Receive Data) before the data is sampled by the block. This noise filter is likely to be a requirement for protocols with a noisy environment.

The noise filter contains the following features:

- Synchronizes the receive input data to the System Clock
- Noise Filter Enable (NFEN) input selects whether the noise filter is bypassed (NFEN = 0) or included (NFEN=1) in the receive data path

12.3.7. Noise Filter Control Register

When MSEL = 001b, the Noise Filter Control Register, shown in Table 91, provides control for the digital noise filter.

Table 91. Noise Filter Control Register (U0CTL1 = F43H with MSEL = 001b)

Bit	7	6	5	4	3	2	1	0
Field	NFEN	NFCTL			—			
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
Address	F43H, F4BH							

Note: R = Read; R/W = Read/Write.

Bit Position	Value	Description
[7] NFEN	**Noise Filter Enable**	
	0	Noise filter is disabled.
	1	Noise filter is enabled. Receive data is preprocessed by the noise filter.
[6:4] NFCTL	**Noise Filter Control** This field controls the delay and noise rejection characteristics of the noise filter. The wider the counter is, the more delay is introduced by the filter and the wider the noise event is filtered.	
	000	4-bit up/down counter.
	001	5-bit up/down counter.
	010	6-bit up/down counter.
	011	7-bit up/down counter.
	100	8-bit up/down counter.
	101	9-bit up/down counter.
	110	10-bit up/down counter.
	111	11-bit up/down counter.
[3:0] Reserved	—	Reserved; must be 0000.

Table 95. LIN-UART Baud Rate Low Byte Register (U0BRL = F47H)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F47H, F4FH							

Note: R/W = Read/Write.

Bit Position	Value	Description
[7:0] BRL	—	**Baud Rate Low** These bits set the Low Byte of the baud rate divisor value.

The LIN-UART data rate is calculated using the following equation for standard UART modes. For the LIN protocol, the Baud Rate registers must be programmed with the baud period rather than 1/16th of the baud period.

Note: The UART must be disabled when updating the Baud Rate registers because the High and Low registers must be written independently.

The LIN-UART data rate is calculated using the following equation for standard UART operation:

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

The LIN-UART data rate is calculated using the following equation for LIN mode UART operation:

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

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

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

16.4.4. ESPI Mode Register

The ESPI Mode Register, shown in Table 112, configures the character bit width and mode of the ESPI I/O pins.

Table 112. ESPI Mode Register (ESPIMODE)

Bits	7	6	5	4	3	2	1	0
Field	SSMD			NUMBITS[2:0]			SSIO	SSPO
Reset	000			0	0	0	0	0
R/W	R/W			R/W	R/W	R/W	R/W	R/W
Address	F63H							

Bit	Description
[7:5] SSMD	**Slave Select Mode** This field selects the behavior of $\overline{SS}$ as a framing signal. For a detailed description of these modes, see Slave Select on page 200. **000 = SPI Mode** When SSIO = 1, the $\overline{SS}$ pin is driven directly from the SSV bit in the Transmit Data Command Register. The Master software should set SSV (or a GPIO output if the $\overline{SS}$ pin is not connected to the appropriate Slave) to the asserted state prior to or on the same clock cycle that the Transmit Data Register is written with the initial byte. At the end of a frame (after the last RDRNE event), SSV will be automatically deasserted by hardware. In this mode, SCK is active only for data transfer (one clock cycle per bit transferred). **001 = Loopback Mode** When ESPI is configured as Master (MMEN = 1), the outputs are deasserted and data is looped from Shift Register Out to Shift Register In. When ESPI is configured as a Slave (MMEN = 0) and $\overline{SS}$ in asserts, MISO (Slave output) is tied to MOSI (Slave input) to provide an asynchronous remote loop back (echo) function. **010 = I2S Mode (Synchronous Framing with SSV)** In this mode, the value from SSV will be output by the Master on the $\overline{SS}$ pin with one SCK period before the data and will remain in that state until the start of the next frame. Typically this mode is used to send back to back frames with $\overline{SS}$ alternating on each frame. A frame boundary is indicated in the Master when SSV changes. A frame boundary is detected in the Slave by $\overline{SS}$ changing state. The $\overline{SS}$ framing signal will lead the frame by one SCK period. In this mode SCK will run continuously, starting with the initial $\overline{SS}$ assertion. Frames will run back-to-back as long as software continues to provide data. An example of this mode is the I^2S protocol (Inter IC Sound) which is used to carry left and right channel audio data with the $\overline{SS}$ signal indicating which channel is being sent. In SLAVE Mode, the change in state of $\overline{SS}$ (Low to High or High to Low) triggers the start of a transaction on the next SCK cycle.

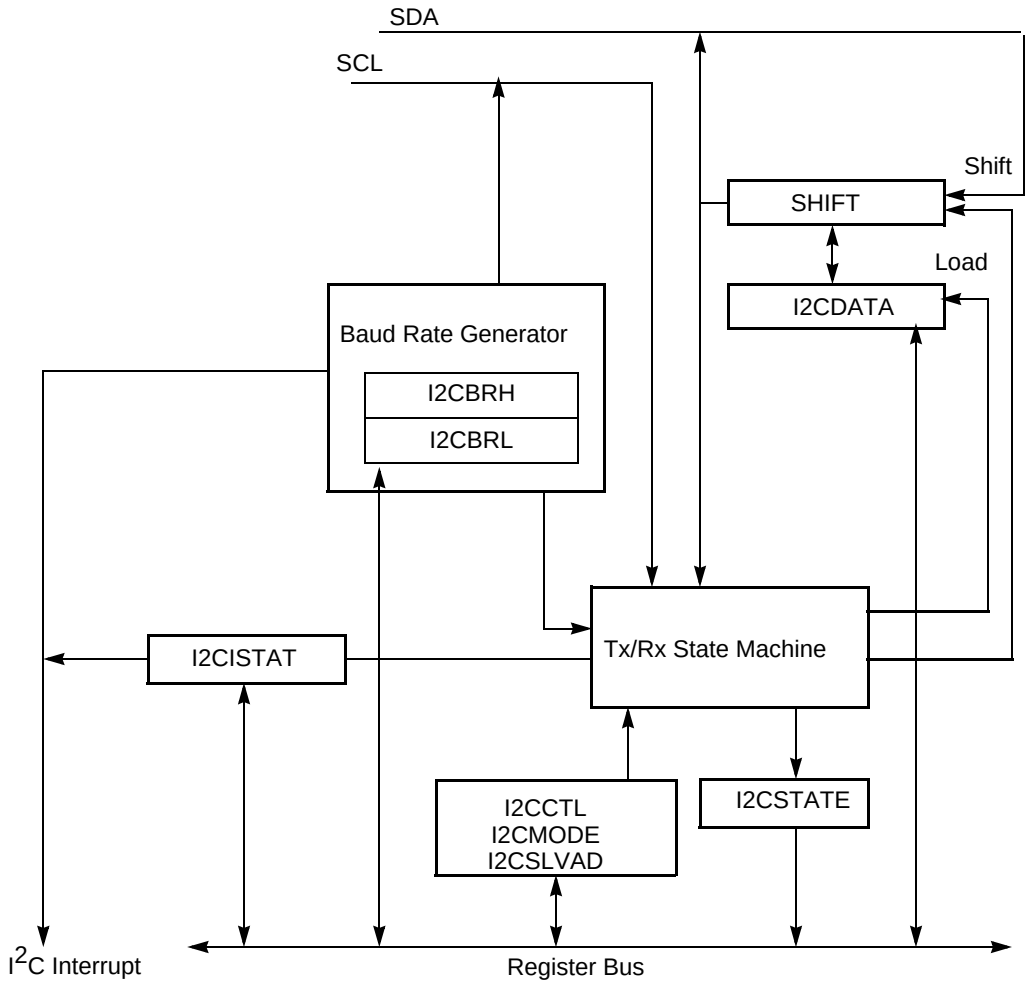


Figure 42. I²C Controller Block Diagram

17.1.1. I²C Master/Slave Controller Registers

Table 118 summarizes the I²C Master/Slave controller's software-accessible registers.

Table 118. I²C Master/Slave Controller Registers

Name	Abbreviation	Description
I²C Data	I2CDATA	Transmit/Receive Data Register.
I²C Interrupt Status	I2CISTAT	Interrupt status register.
I²C Control	I2CCTL	Control Register—basic control functions.

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.

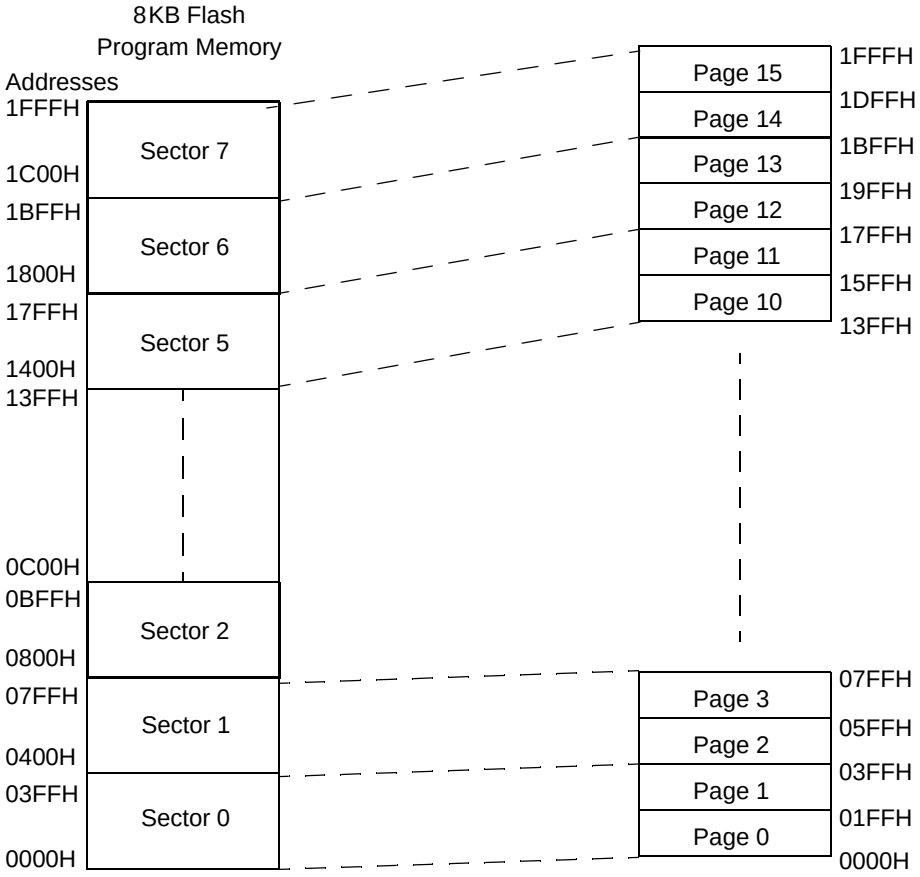


Figure 51. 8KB Flash Memory Arrangement

Table 133. Flash Code Protection Using the Flash Option Bit

FWP	Flash Code Protection Description
0	Programming and erasing disabled for all of Flash Program Memory. In user code programming, Page Erase and Mass Erase are all disabled. Mass Erase is available through the On-Chip Debugger.
1	Programming, Page Erase and Mass Erase are enabled for all of Flash Program Memory.

20.2.3.2. Flash Code Protection Using the Flash Controller

At Reset, the Flash Controller locks to prevent accidental program or erasure of the contents of Flash memory. Follow the steps below to unlock the Flash Controller from user code:

1. Write the Page Select Register with the target page.
2. Write the first unlock command 73H to the Flash Control Register.
3. Write the second unlock command 8CH to the Flash Control Register.
4. Rewrite the Page Select Register with the same page previously stored there.

If the two Page Select writes do not match, the controller reverts to a locked state. If the two writes match, the selected page becomes active. For details, see the flowchart in Figure 54 on page 266.

Note: Byte Programming, Page Erase and Mass Erase will not be allowed if the FWP bit is cleared or if the page resides in a protected block.

After unlocking a specific page, Byte Programing or Page Erase can be performed. At the conclusion of a Page Erase, the Flash Controller is automatically locked. To lock the Flash Controller after Byte Programming, write to the Flash Control Register with any value other than the Page Erase or Mass Erase comands.

20.2.3.3. Sector Based Flash Protection

The final protection mechanism is implemented on a per-sector basis. The Flash memories of Z8 Encore! devices are divided into a maximum number of 8 sectors. A sector is 1/8 of the total size of Flash memory unless this value is smaller than the page size, in which case the sector and page sizes are equal. On the Z8 Encore! XP F1680 Series devices, the sector size is 3KB, 2KB or 1KB depending on available on-chip Flash size of 24KB, 16KB and 8KB.

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.

Bit	Description (Continued)
[5:4] OSC_SEL[1:0]	**Oscillator Mode Selection** 00 = On-chip oscillator configured for use with external RC networks or external clock input (<4MHz). 01 = Reserved. 10 = Medium power for use with medium frequency crystals or ceramic resonators (1.0MHz to 8.0MHz). 11 = Maximum power for use with high-frequency crystals (8.0MHz to 20.0MHz). This setting is default for unprogrammed (erased) Flash.
[3] VBO_AO	**Voltage Brown-Out Protection Always ON** 0 = Voltage Brown-Out Protection is disabled in STOP Mode to reduce total power consumption. And it is controlled by VBO/LVD control bit of Power Control Register in ACTIVE and HALT Mode. 1 = Voltage Brown-Out Protection is always enabled including during STOP Mode. And it cannot be disabled by VBO/LVD control bit of Power Control Register in any mode. This setting is default for unprogrammed (erased) Flash.
[2] FRP	**Flash Read Protect** 0 = User program code is inaccessible. Limited control features are available through the On-Chip Debugger. 1 = User program code is accessible. All On-Chip Debugger commands are enabled. This setting is default for unprogrammed (erased) Flash.
[1] PRAM_M	**On-Chip Program RAM Mode Select** 0 = Program RAM is used as on-chip Register RAM and it begins at the first available Register File address space. See the Register Map chapter on page 23. 1 = Program RAM is used as on-chip Program RAM and it begins at address E000H in the Program Memory address space. This setting is default for unprogrammed (erased) Flash.
[0] FWP	**Flash Write Protect** This option bit provides Flash program memory protection: 0 = Programming and erasure disabled for all of Flash program memory. Programming, Page Erase and Mass Erase through User Code are disabled. Mass Erase is available using the On-Chip Debugger. 1 = Programming, Page Erase and Mass Erase are enabled for all of Flash program memory.

18H. The register becomes locked upon successful completion of a register write to the OSCCTL0.

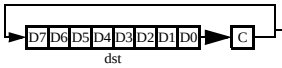
Table 170. Oscillator Control 0 Register (OSCCTL0)

Bits	7	6	5	4	3	2	1	0
Field	INTEN*	XTLEN	WDTEN	POFEN	WDFEN	SCKSEL		
Reset	1	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	F86H							

Bit	Description
[7] INTEN	**Internal Precision Oscillator Enable** 1 = Internal precision oscillator is enabled. 0 = Internal precision oscillator is disabled.
[6] XTLEN	**Crystal Oscillator Enable** This setting overrides the GPIO register control for PA0 and PA1. 1 = Crystal oscillator is enabled. 0 = Crystal oscillator is disabled.
[5] WDTEN	**Watchdog Timer Oscillator Enable** 1 = Watchdog Timer oscillator is enabled. 0 = Watchdog Timer oscillator is disabled.
[4] POFEN	**Primary Oscillator Failure Detection Enable** 1 = Failure detection and recovery of primary oscillator is enabled. 0 = Failure detection and recovery of primary oscillator is disabled.
[3] WDFEN	**Watchdog Timer Oscillator Failure Detection Enable** 1 = Failure detection of Watchdog Timer oscillator is enabled. 0 = Failure detection of Watchdog Timer oscillator is disabled.
[2:0] SCKSEL	**System Clock Oscillator Select** 000 = Internal precision oscillator functions as system clock. 001 = Reserved. 010 = Crystal oscillator or external RC oscillator functions as system clock. 011 = Watchdog Timer oscillator functions as system. 100 = External clock signal on PB3 functions as system clock. 101 = Reserved. 110 = Reserved. 111 = Reserved.

Note: The INTEN bit should be disabled when you use another clock as a system clock.

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.

Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7	3, 2 **PUSH** IM															
8																
9																
A			3.3 **CPC** r1,r2	3.4 **CPC** r1,Ir2	4.3 **CPC** R2,R1	4.4 **CPC** IR2,R1	4.3 **CPC** R1,IM	4.4 **CPC** IR1,IM	5.3 **CPCX** ER2,ER1	5.3 **CPCX** IM,ER1						
B																
C	3.2 **SRL** R1	3.3 **SRL** IR1														
D																
E									5, 4 **LDWX** ER2,ER1							
F																

Figure 68. Second Op Code Map after 1FH

Table 193. Flash Memory Electrical Characteristics and Timing

	VDD = 2.7V to 3.6V TA = 0°C to +70°C TA = –40°C to +105°C				
Parameter	**Min**	**Typ**	**Max**	**Units**	**Conditions**
Flash Byte Read Time	100	–	–	ns	V_{DD} = 1.8 V to 3.6V
Flash Byte Program Time	20	–	40	µs	
Flash Page Erase Time	50	–	–	ms	
Flash Mass Erase Time	50	–	–	ms	
Writes to Single Address Before Next Erase	–	–	2		
Data Retention	20	–	–	years	25°C
Endurance	5,000	–	–	cycles	Program/erase cycles

Table 194. Watchdog Timer Electrical Characteristics and Timing

		VDD = 1.8V to 3.6V TA = 0°C to +70°C TA = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Unit**	**Conditions**
$T_{STARTUP}$		–	–	10	ms	After pd disable only
I_{DD}WDT	WDT Active Current	–	–	5	µA	
I_{DDQ}WDT	WDT Quiescent Current	–	5	–	nA	
F_{WDT}	WDT Oscillator Frequency	2.5	5	20	kHz	

Table 195. Non-Volatile Data Storage

	VDD = 2.7V to 3.6V TA = 0°C to +70°C TA = –40°C to +105°C				
Parameter	**Min**	**Typ**	**Max**	**Units**	**Conditions**
NVDS Byte Read Time	34	–	519	µs	With system clock at 20MHz
NVDS Byte Program Time	0.171	–	39.7	ms	With system clock at 20MHz
Data Retention	20	–	–	years	25°C
Endurance	50,000	–	–	cycles	Cumulative write cycles for entire memory