



Welcome to [E-XFL.COM](https://www.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	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4822vs020sc00tr



Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2007 by Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.



**ISO 9001:2000
FS 507510**

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

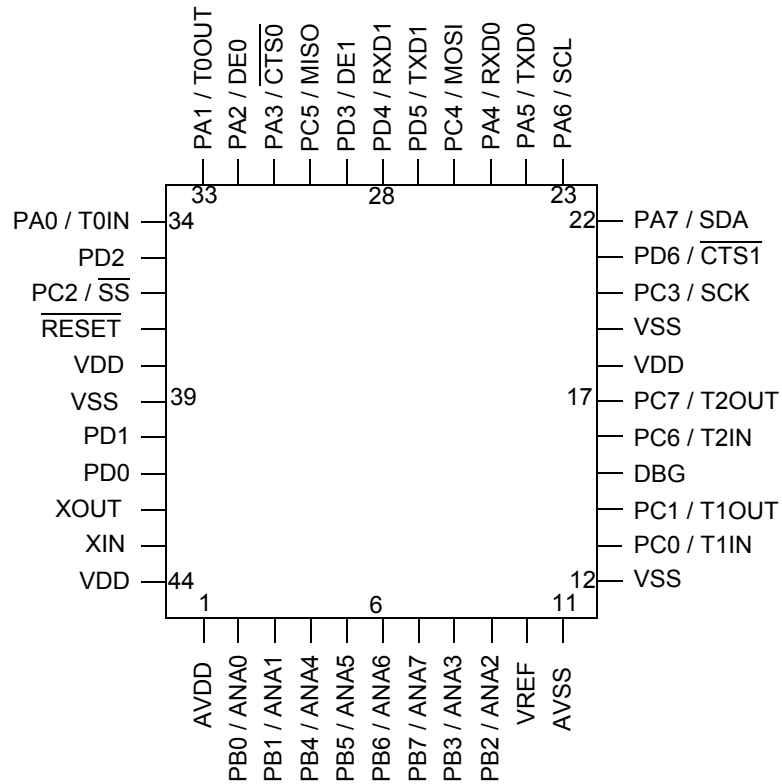


Figure 4. Z8 Encore! XP 64K Series Flash Microcontrollers in 44-Pin Low-Profile Quad Flat Package (LQFP)

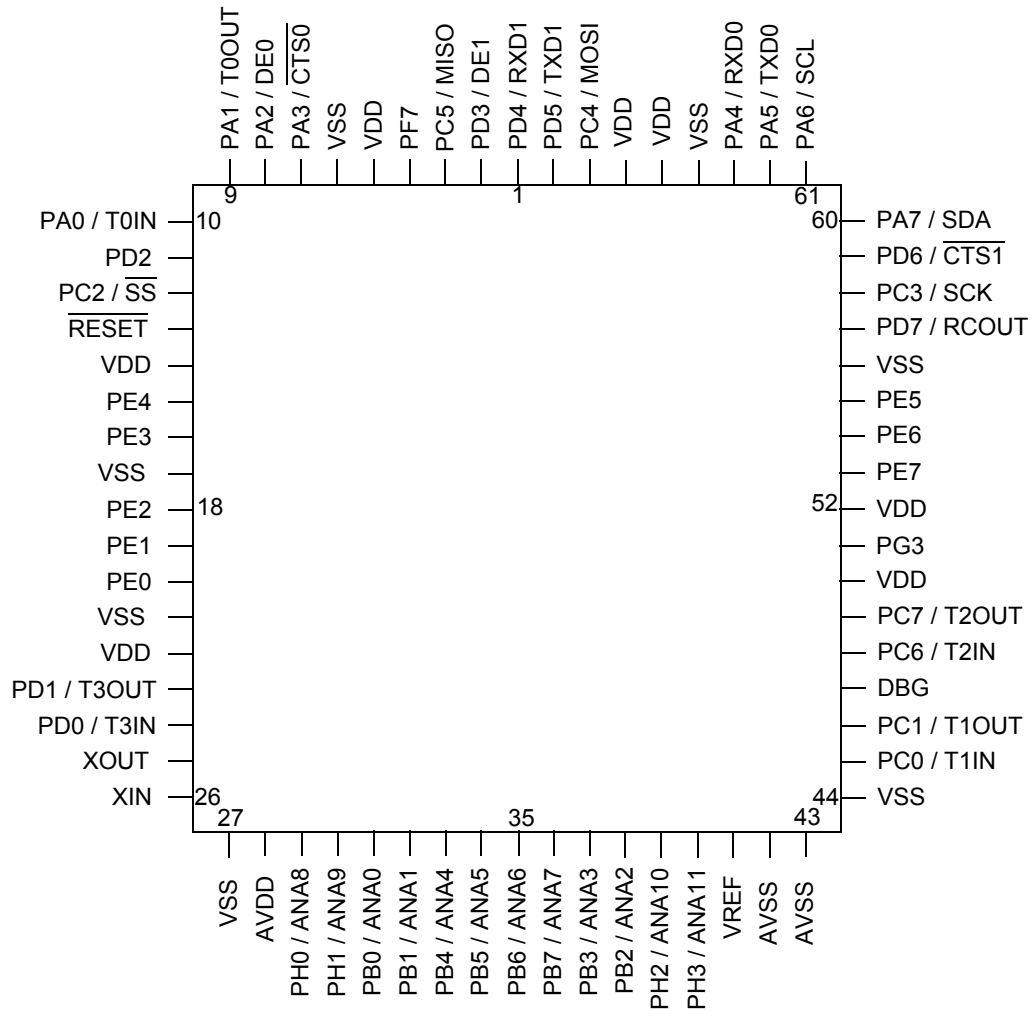
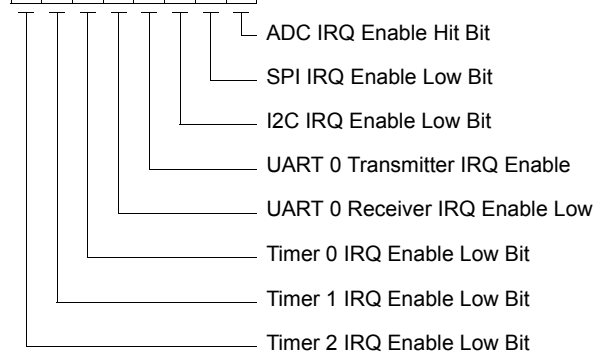


Figure 6. Z8 Encore! XP 64K Series Flash Microcontrollers in 68-Pin Plastic Leaded Chip Carrier (PLCC)

IRQ0 Enable Low Bit

IRQ0ENL (FC2H - Read/Write)

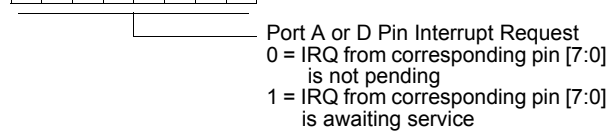
D7 D6 D5 D4 D3 D2 D1 D0



Interrupt Request 1

IRQ1 (FC3H - Read/Write)

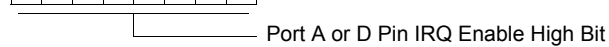
D7 D6 D5 D4 D3 D2 D1 D0



IRQ1 Enable High Bit

IRQ1ENH (FC4H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0



IRQ1 Enable Low Bit

IRQ1ENL (FC5H - Read/Write)

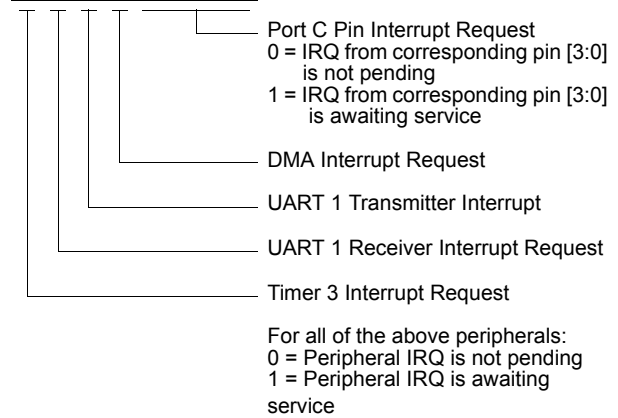
D7 D6 D5 D4 D3 D2 D1 D0



Interrupt Request 2

IRQ2 (FC6H - Read/Write)

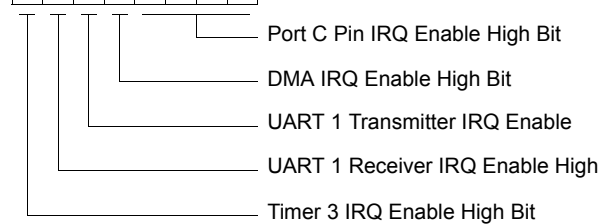
D7 D6 D5 D4 D3 D2 D1 D0



IRQ2 Enable High Bit

IRQ2ENH (FC7H - Read/Write)

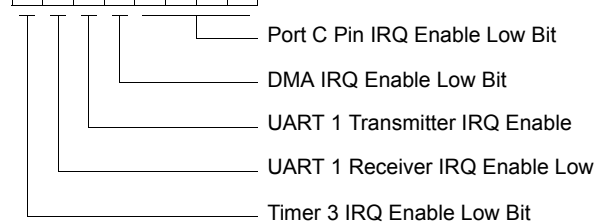
D7 D6 D5 D4 D3 D2 D1 D0



IRQ2 Enable Low Bit

IRQ2ENL (FC8H - Read/Write)

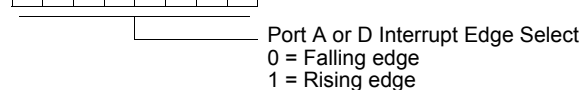
D7 D6 D5 D4 D3 D2 D1 D0



Interrupt Edge Select

IRQES (FCDH - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0



Port F Control

PFCTL (FE5H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port F Control[7:0]
Provides Access to Port Sub-Registers

Port F Input Data

PFIN (FE6H - Read Only)

D7 D6 D5 D4 D3 D2 D1 D0

Port F Input Data [7:0]

Port F Output Data

PFOUT (FE7H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port F Output Data [7:0]

Port G Address

PGADDR (FE8H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port G Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port G Control

PGCTL (FE9H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port G Control[7:0]
Provides Access to Port Sub-Registers

Port G Input Data

PGIN (FEAH - Read Only)

D7 D6 D5 D4 D3 D2 D1 D0

Port G Input Data [7:0]

Port G Output Data

PGOUT (FEBH - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port G Output Data [7:0]

Port H Address

PHADDR (FECH - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port H Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port H Control

PHCTL (FEDH - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port H Control [3:0]
Provides Access to Port Sub-Registers
Reserved

Port H Input Data

PHIN (FEEH - Read Only)

D7 D6 D5 D4 D3 D2 D1 D0

Port H Input Data [3:0]
Reserved

Port H Output Data

PHOUT (FEFH - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

Port H Output Data [3:0]
Reserved

Stop Mode Recovery Using a GPIO Port Pin Transition HALT

Each of the GPIO Port pins may be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. The GPIO Stop Mode Recovery signals are filtered to reject pulses less than 10 ns (typical) in duration. In the Watchdog Timer Control register, the STOP bit is set to 1.



Caution: *In STOP mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the Port pin through the end of the Stop Mode Recovery delay. Thus, short pulses on the Port pin can initiate Stop Mode Recovery without being written to the Port Input Data register or without initiating an interrupt (if enabled for that pin).*



Caution: *The following style of coding to generate software interrupts by setting bits in the Interrupt Request registers is NOT recommended. All incoming interrupts that are received between execution of the first LDX command and the last LDX command are lost.*

Poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, the following style of coding to set bits in the Interrupt Request registers is recommended:

Good coding style that avoids lost interrupt requests:

```
ORX IRQ0, MASK
```

Interrupt Control Register Definitions

For all interrupts other than the Watchdog Timer interrupt, the interrupt control registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) register ([Table 24](#)) stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8[™] CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 0 register to determine if any interrupt requests are pending

Table 24. Interrupt Request 0 Register (IRQ0)

BITS	7	6	5	4	3	2	1	0
FIELD	T2I	T1I	T0I	U0RXI	U0TXI	I2CI	SPII	ADCI
RESET	0							
R/W	R/W							
ADDR	FC0H							

T2I—Timer 2 Interrupt Request

0 = No interrupt request is pending for Timer 2.

1 = An interrupt request from Timer 2 is awaiting service.

REN—Receive Enable

This bit enables or disables the receiver.

0 = Receiver disabled.

1 = Receiver enabled.

CTSE—CTS Enable

0 = The $\overline{\text{CTS}}$ signal has no effect on the transmitter.

1 = The UART recognizes the $\overline{\text{CTS}}$ signal as an enable control from the transmitter.

PEN—Parity Enable

This bit enables or disables parity. Even or odd is determined by the PSEL bit. It is overridden by the MPEN bit.

0 = Parity is disabled.

1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit.

PSEL—Parity Select

0 = Even parity is transmitted and expected on all received data.

1 = Odd parity is transmitted and expected on all received data.

SBRK—Send Break

This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so ensure that the transmitter has finished sending data before setting this bit.

0 = No break is sent.

1 = The output of the transmitter is zero.

STOP—Stop Bit Select

0 = The transmitter sends one stop bit.

1 = The transmitter sends two stop bits.

LBEN—Loop Back Enable

0 = Normal operation.

1 = All transmitted data is looped back to the receiver.

Table 57. UART Control 1 Register (UxCTL1)

BITS	7	6	5	4	3	2	1	0
FIELD	MPMD[1]	MPEN	MPMD[0]	MPBT	DEPOL	BRGCTL	RDAIRQ	IREN
RESET	0							
R/W	R/W							
ADDR	F43H and F4BH							

MPMD[1:0]—MULTIPROCESSOR Mode

If MULTIPROCESSOR (9-bit) mode is enabled,

00 = The UART generates an interrupt request on all received bytes (data and address).

- 01 = The UART generates an interrupt request only on received address bytes.
- 10 = The UART generates an interrupt request when a received address byte matches the value stored in the Address Compare Register and on all successive data bytes until an address mismatch occurs.
- 11 = The UART generates an interrupt request on all received data bytes for which the most recent address byte matched the value in the Address Compare Register.

MPEN—MULTIPROCESSOR (9-bit) Enable

This bit is used to enable MULTIPROCESSOR (9-bit) mode.

0 = Disable MULTIPROCESSOR (9-bit) mode.

1 = Enable MULTIPROCESSOR (9-bit) mode.

MPBT—MULTIPROCESSOR Bit Transmit

This bit is applicable only when MULTIPROCESSOR (9-bit) mode is enabled.

0 = Send a 0 in the multiprocessor bit location of the data stream (9th bit).

1 = Send a 1 in the multiprocessor bit location of the data stream (9th bit).

DEPOL—Driver Enable Polarity

0 = DE signal is Active High.

1 = DE signal is Active Low.

BRGCTL—Baud Rate Control

This bit causes different UART behavior depending on whether the UART receiver is enabled ($\overline{\text{REN}} = 1$ in the UART Control 0 Register).

When the UART receiver is not enabled, this bit determines whether the Baud Rate Generator issues interrupts.

0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value

1 = The Baud Rate Generator generates a receive interrupt when it counts down to 0.

Reads from the Baud Rate High and Low Byte registers return the current BRG count value.

When the UART receiver is enabled, this bit allows reads from the Baud Rate Registers to return the BRG count value instead of the Reload Value.

0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value.

1 = Reads from the Baud Rate High and Low Byte registers return the current BRG count value. Unlike the Timers, there is no mechanism to latch the High Byte when the Low Byte is read.

$\overline{\text{RDAIRQ}}$ —Receive Data Interrupt Enable

0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller.

1 = Received data does not generate an interrupt request to the Interrupt Controller.
Only receiver errors generate an interrupt request.

IREN—Infrared Encoder/Decoder Enable

0 = Infrared Encoder/Decoder is disabled. UART operates normally operation.

During an SPI transfer, data is sent and received simultaneously by both the Master and the Slave SPI devices. Separate signals are required for data and the serial clock. When an SPI 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 a second data pin. An 8-bit shift register in the Master and another 8-bit shift register in the Slave are connected as a circular buffer. The SPI shift register is single-buffered in the transmit and receive directions. New data to be transmitted cannot be written into the shift register until the previous transmission is complete and receive data (if valid) has been read.

SPI Signals

The four basic SPI signals are:

- [Master-In/Slave-Out](#)
- [Master-Out/Slave-In](#)
- [Serial Clock](#)
- [Slave Select](#)

Each signal is described in both Master and Slave modes.

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. It is one of the two lines that transfer serial data, with the most significant bit sent first. The MISO pin of a Slave device is placed in a high-impedance state if the Slave is not selected. When the SPI is not enabled, this signal is in a high-impedance state.

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. It is one of the two lines that transfer serial data, with the most significant bit sent first. When the SPI is not enabled, this signal is in a high-impedance state.

Serial Clock

The Serial Clock (SCK) synchronizes data movement both in and out of the device through its MOSI and MISO pins. In MASTER mode, the SPI's Baud Rate Generator creates the serial clock. The Master drives the serial clock out its own SCK pin to the Slave's SCK pin. When the SPI is configured as a Slave, the SCK pin is an input and the clock signal from the Master synchronizes the data transfer between the Master and Slave devices. Slave devices ignore the SCK signal, unless the \overline{SS} pin is asserted. When configured as a slave, the SPI block requires a minimum SCK period of greater than or equal to 8 times the system (XIN) clock period.

The Master and Slave are each capable of exchanging a character of data during a sequence of NUMBITS clock cycles (see NUMBITS field in the [SPI Mode Register](#) on page 140). In both Master and Slave SPI devices, data is shifted on one edge of the SCK and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI phase and polarity control.

Slave Select

The active Low Slave Select (\overline{SS}) input signal selects a Slave SPI device. \overline{SS} must be Low prior to all data communication to and from the Slave device. \overline{SS} must stay Low for the full duration of each character transferred. The \overline{SS} signal may stay Low during the transfer of multiple characters or may deassert between each character.

When the SPI is configured as the only Master in an SPI system, the \overline{SS} pin can be set as either an input or an output. For communication between the Z8F642x family Z8R642x family device's SPI Master and external Slave devices, the \overline{SS} signal, as an output, can assert the \overline{SS} input pin on one of the Slave devices. Other GPIO output pins can also be employed to select external SPI Slave devices.

When the SPI is configured as one Master in a multi-master SPI system, the \overline{SS} pin must be set as an input. The \overline{SS} input signal on the Master must be High. If the \overline{SS} signal goes Low (indicating another Master is driving the SPI bus), a Collision error Flag is set in the SPI Status register.

SPI Clock Phase and Polarity Control

The SPI supports four combinations of serial clock phase and polarity using two bits in the SPI Control register. The clock polarity bit, CLKPOL, selects an active high or active Low clock and has no effect on the transfer format. [Table 62](#) lists the SPI Clock Phase and Polarity Operation parameters. The clock phase bit, PHASE, selects one of two fundamentally different transfer formats. For proper data transmission, the clock phase and polarity must be identical for the SPI Master and the SPI Slave. The Master always places data on the MOSI line a half-cycle before the receive clock edge (SCK signal), in order for the Slave to latch the data.

Table 62. SPI Clock Phase (PHASE) and Clock Polarity (CLKPOL) Operation

PHASE	CLKPOL	SCK Transmit Edge	SCK Receive Edge	SCK Idle State
0	0	Falling	Rising	Low
0	1	Rising	Falling	High
1	0	Rising	Falling	Low
1	1	Falling	Rising	High

- Master receives from a 7-bit slave
- Master receives from a 10-bit slave

SDA and SCL Signals

I²C sends all addresses, data and acknowledge signals over the SDA line, most-significant bit first. SCL is the common clock for the I²C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I²C) is responsible for driving the SCL clock signal, although the clock signal can become skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a high level. When the slave releases the clock, the I²C Controller continues the transaction. All data is transferred in bytes and there is no limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the high period of SCL.

I²C Interrupts

The I²C Controller contains four sources of interrupts—Transmit, Receive, Not Acknowledge and baud rate generator. These four interrupt sources are combined into a single interrupt request signal to the Interrupt Controller. The Transmit interrupt is enabled by the IEN and TXI bits of the Control register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the Control register. The baud rate generator interrupt is enabled by the BIRQ and IEN bits of the Control register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I²C Controller and neither the *START* or *STOP* bit is set. The Not Acknowledge event sets the NCKI bit of the I²C Status register and can only be cleared by setting the *START* or *STOP* bit in the I²C Control register. When this interrupt occurs, the I²C Controller waits until either the *STOP* or *START* bit is set before performing any action. In an interrupt service routine, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

Receive interrupts occur when a byte of data has been received by the I²C Controller (master reading data from slave). This procedure sets the RDRF bit of the I²C Status register. The RDRF bit is cleared by reading the I²C Data register. The RDRF bit is set during the acknowledge phase. The I²C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.

TXRXSTATE	State Description
1_1101	10-bit addressing: Bit 3 of 2nd address byte 7-bit addressing: Bit 3 of address byte
1_1110	10-bit addressing: Bit 2 of 2nd address byte 7-bit addressing: Bit 2 of address byte
1_1111	10-bit addressing: Bit 1 of 2nd address byte 7-bit addressing: Bit 1 of address byte

I²C Diagnostic Control Register

The I²C Diagnostic register (Table 76) provides control over diagnostic modes. This register is a read/write register used for I²C diagnostics.

Table 76. I²C Diagnostic Control Register (I2CDIAG)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved							DIAG
RESET	0							
R/W	R							R/W
ADDR	F56H							

DIAG = Diagnostic Control Bit - Selects read back value of the Baud Rate Reload registers.

0 = NORMAL mode. Reading the Baud Rate High and Low Byte registers returns the baud rate reload value.

1 = DIAGNOSTIC mode. Reading the Baud Rate High and Low Byte registers returns the baud rate counter value.

DMAx Start/Current Address Low Byte Register

The DMAx Start/Current Address Low register, in conjunction with the DMAx Address High Nibble register, forms a 12-bit Start/Current Address. Writes to this register set the Start Address for DMA operations. Each time the DMA completes a data transfer, the 12-bit Start/Current Address increments by either 1 (single-byte transfer) or 2 (two-byte word transfer). Reads from this register return the low byte of the Current Address to be used for the next DMA data transfer.

Table 80. DMAx Start/Current Address Low Byte Register (DMAxSTART)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_START							
RESET	X							
R/W	R/W							
ADDR	FB3H, FBBH							

DMA_START—DMAx Start/Current Address Low

These bits, with the four lower bits of the DMAx_H register, form the 12-bit Start/Current address. The full 12-bit address is given by {DMA_START_H[3:0], DMA_START[7:0]}.

DMAx End Address Low Byte Register

The DMAx End Address Low Byte register ([Table 80](#)), in conjunction with the DMAx_H register ([Table 81](#)), forms a 12-bit End Address.

Table 81. DMAx End Address Low Byte Register (DMAxEND)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_END							
RESET	X							
R/W	R/W							
ADDR	FB4H, FBCH							

DMA_END—DMAx End Address Low

These bits, with the four upper bits of the DMAx_H register, form a 12-bit address. This address is the ending location of the DMAx transfer. The full 12-bit address is given by {DMA_END_H[3:0], DMA_END[7:0]}.

Option Bits

Overview

Option Bits allow user configuration of certain aspects of the 64K Series operation. The feature configuration data is stored in the Flash Memory and read during Reset. The features available for control via the Option Bits are:

- Watchdog Timer time-out response selection—interrupt or Reset.
- Watchdog Timer enabled at Reset.
- The ability to prevent unwanted read access to user code in Flash Memory.
- The ability to prevent accidental programming and erasure of the user code in Flash Memory.
- Voltage Brownout configuration—always enabled or disabled during STOP mode to reduce STOP mode power consumption.
- Oscillator mode selection—for high, medium, and low power crystal oscillators, or external RC oscillator.

Operation

Option Bit Configuration By Reset

Each time the Option Bits are programmed or erased, the device must be Reset for the change to take place. During any reset operation (System Reset, Reset, or Stop Mode Recovery), the Option Bits are automatically read from the Flash Memory and written to Option Configuration registers. The Option Configuration registers control operation of the devices within the 64K Series. 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.

Option Bit Address Space

The first two bytes of Flash Memory at addresses 0000H (see [Table 98](#) on page 196) and 0001H (see [Table 99](#) on page 197) are reserved for the user Option Bits. The byte at Flash Memory address 0000H configures user options. The byte at Flash Memory address 0001H is reserved for future use and must remain unprogrammed.

Flash Memory Address 0000H

Table 98. Flash Option Bits At Flash Memory Address 0000H

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RE S	WDT_AO	OSC_SEL[1:0]		VBO_AO	RP	Reserved	FWP
RESET	U							
R/W	R/W							
ADDR	Program Memory 0000H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

WDT_RES—Watchdog Timer Reset

0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request.

1 = Watchdog Timer time-out causes a Short Reset. This setting is the default for unprogrammed (erased) Flash.

WDT_AO—Watchdog Timer Always On

0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled except during STOP Mode (if configured to power down during STOP Mode).

1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

OSC_SEL[1:0]—Oscillator Mode Selection

00 = On-chip oscillator configured for use with external RC networks (<4 MHz).

01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).

10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 10.0 MHz).

11 = Maximum power for use with high frequency crystals (8.0 MHz to 20.0 MHz). This setting is the default for unprogrammed (erased) Flash.

VBO_AO—Voltage Brownout Protection Always On

0 = Voltage Brownout Protection is disabled in STOP mode to reduce total power consumption.

1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

RP—Read Protect

0 = User program code is inaccessible. Limited control features are available through

Table 123. Additional Symbols

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
B	Binary Number Suffix
%	Hexadecimal Number Prefix
H	Hexadecimal Number Suffix

Assignment of a value is indicated by an arrow. For example,

$$\text{dst} \leftarrow \text{dst} + \text{src}$$

indicates the source data is added to the destination data and the result is stored in the destination location.

Condition Codes

The C, Z, S and V Flags control the operation of the conditional jump (JP cc and JR cc) instructions. Sixteen frequently useful functions of the Flag settings are encoded in a 4-bit field called the condition code (cc), which forms Bits 7:4 of the conditional jump instructions. The condition codes are summarized in [Table 124](#). Some binary condition codes can be created using more than one assembly code mnemonic. The result of the Flag test operation decides if the conditional jump is executed.

Table 124. Condition Codes

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0000	0	F	Always False	—
0001	1	LT	Less Than	(S XOR V) = 1
0010	2	LE	Less Than or Equal	(Z OR (S XOR V)) = 1

Figure 67 displays the 80-pin Quad Flat Package (QFP) available for the Z8X4823 and Z8X6423 devices.

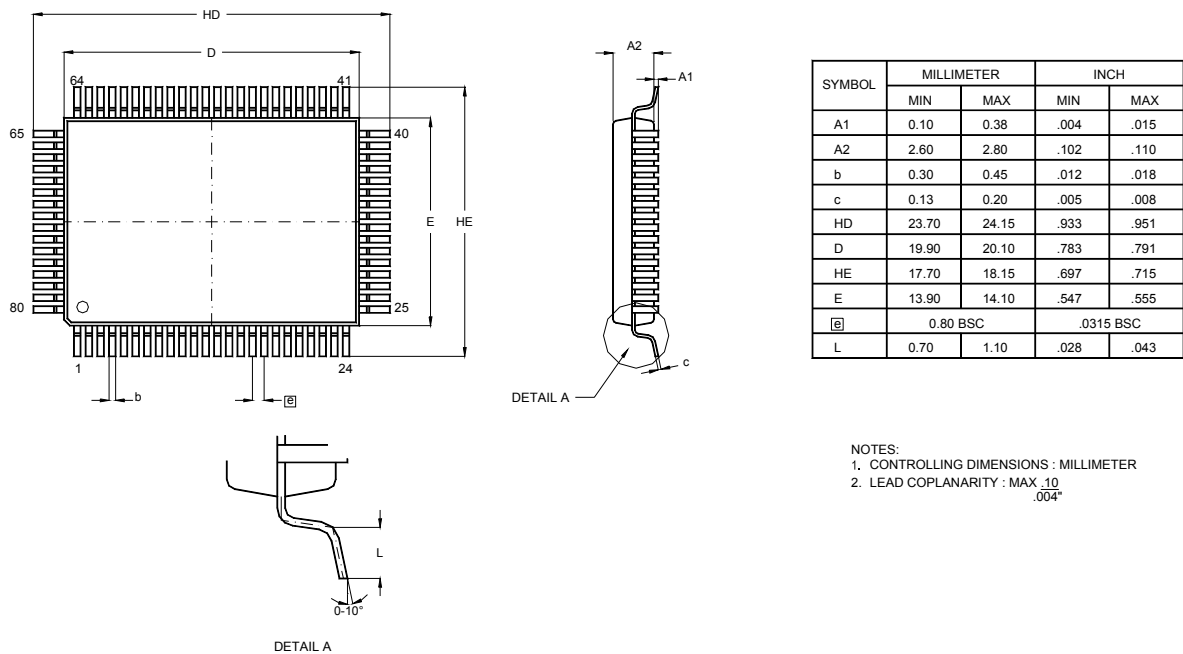


Figure 67. 80-Lead Quad-Flat Package (QFP)

I2CBRL register 161	CPU control 247
I2CCTL register 158	CPX 246
I2CDATA register 157	DA 246
I2CSTAT register 157	DEC 246
IM 243	DECW 246
immediate data 243	DI 247
immediate operand prefix 244	DJNZ 249
INC 246	EI 247
increment 246	HALT 247
increment word 246	INC 246
INCW 246	INCW 246
indexed 243	IRET 249
indirect address prefix 244	JP 249
indirect register 243	LD 248
indirect register pair 243	LDC 248
indirect working register 243	LDCI 247, 248
indirect working register pair 243	LDE 248
infrared encoder/decoder (IrDA) 125	LDEI 247
instruction set, ez8 CPU 241	LDX 248
instructions	LEA 248
ADC 246	load 248
ADCX 246	logical 248
ADD 246	MULT 246
ADDX 246	NOP 247
AND 248	OR 248
ANDX 248	ORX 248
arithmetic 246	POP 248
BCLR 246	POPX 248
BIT 246	program control 249
bit manipulation 246	PUSH 248
block transfer 247	PUSHX 248
BRK 249	RCF 247
BSET 246	RET 249
BSWAP 247, 249	RL 249
BTJ 249	RLC 249
BTJNZ 249	rotate and shift 249
BTJZ 249	RR 249
CALL 249	RRC 249
CCF 247	SBC 246
CLR 248	SCF 247
COM 248	SRA 249
CP 246	SRL 250
CPC 246	SRP 247
CPCX 246	STOP 248

- flash page select (FPS) 191
- flash status (FSTAT) 190
- GPIO port A-H address (PxADDR) 61
- GPIO port A-H alternate function sub-registers 63
- GPIO port A-H control address (PxCTL) 62
- GPIO port A-H data direction sub-registers 63
- I2C baud rate high (I2CBRH) 160, 161, 163
- I2C control (I2CCTL) 158
- I2C data (I2CDATA) 157
- I2C status 157
- I2C status (I2CSTAT) 157
- I2Cbaud rate low (I2CBRL) 161
- mode, SPI 140
- OCD control 209
- OCD status 210
- SPI baud rate high byte (SPIBRH) 142
- SPI baud rate low byte (SPIBRL) 142
- SPI control (SPICCTL) 138
- SPI data (SPIDATA) 137
- SPI status (SPISTAT) 139
- status, I2C 157
- status, SPI 139
- UARTx baud rate high byte (UxBRH) 121
- UARTx baud rate low byte (UxBRL) 121
- UARTx Control 0 (UxCTL0) 117, 120
- UARTx control 1 (UxCTL1) 118
- UARTx receive data (UxRXD) 115
- UARTx status 0 (UxSTAT0) 115
- UARTx status 1 (UxSTAT1) 117
- UARTx transmit data (UxTXD) 114
- watch-dog timer control (WDTCTL) 100
- watch-dog timer reload high byte (WDTH) 102
- watch-dog timer reload low byte (WDTL) 102
- watch-dog timer reload upper byte (WDTU) 102
- register file 19
- register file address map 23
- register pair 243
- register pointer 244
- reset
 - and STOP mode characteristics 48
 - carry flag 247
 - controller 5
 - sources 48
- RET 249
- return 249
- RL 249
- RLC 249
- rotate and shift instructions 249
- rotate left 249
- rotate left through carry 249
- rotate right 249
- rotate right through carry 249
- RP 244
- RR 243, 249
- rr 243
- RRC 249
- S**
- SBC 246
- SCF 247
- SDA and SCL (IrDA) signals 145
- second opcode map after 1FH 264
- serial clock 131
- serial peripheral interface (SPI) 129
- set carry flag 247
- set register pointer 247
- shift right arithmetic 249
- shift right logical 250
- signal descriptions 14
- single-shot conversion (ADC) 177
- SIO 5
- slave data transfer formats (I2C) 151
- slave select 132
- software trap 249
- source operand 244
- SP 244
- SPI
 - architecture 129
 - baud rate generator 136
 - baud rate high and low byte register 142
 - clock phase 132
 - configured as slave 130
 - control register 137
 - control register definitions 137
 - data register 137