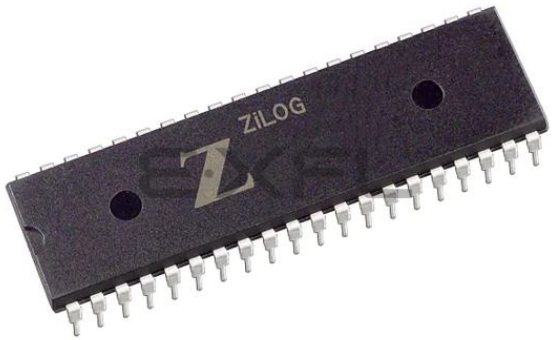


Welcome to E-XFL.COM

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	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	29
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°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/z8f2421pm020sc

Block Diagram

Figure 1 displays the block diagram of the architecture of the Z8 Encore! XP 64K Series Flash Microcontrollers.

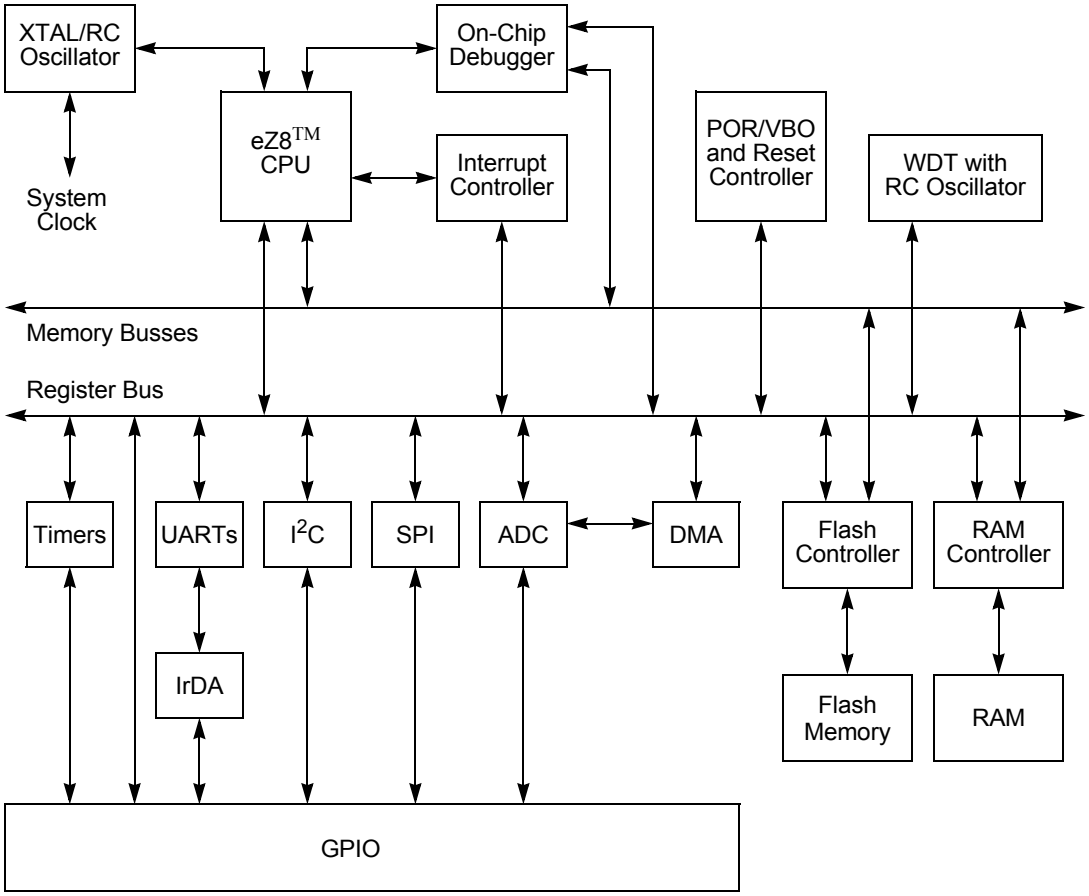


Figure 1. Z8 Encore! XP 64K Series Flash Microcontrollers Block Diagram

CPU and Peripheral Overview

eZ8™ CPU Features

The latest 8-bit eZ8 CPU meets the continuing demand for faster and more code-efficient microcontrollers. The eZ8 CPU executes a superset of the original Z8® instruction set.

Table 4. Pin Characteristics of the Z8 Encore! XP 64K Series Flash Microcontrollers

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tri-State Output	Internal Pull-up or Pull-down	Schmitt-Trigger Input	Open Drain Output
AVSS	N/A	N/A	N/A	N/A	No	No	N/A
AVDD	N/A	N/A	N/A	N/A	No	No	N/A
DBG	I/O	I	N/A	Yes	No	Yes	Yes
VSS	N/A	N/A	N/A	N/A	No	No	N/A
PA[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PB[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PC[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PD[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PE7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PF[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PG[7:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
PH[3:0]	I/O	I	N/A	Yes	No	Yes	Yes, Programmable
RESET	I	I	Low	N/A	Pull-up	Yes	N/A
VDD	N/A	N/A	N/A	N/A	No	No	N/A
XIN	I	I	N/A	N/A	No	No	N/A
XOUT	O	O	N/A	Yes, in STOP mode	No	No	No

Note: *x* represents integer 0, 1,... to indicate multiple pins with symbol mnemonics that differ only by the integer.

Address Space

Overview

The eZ8™ CPU can access three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral, and general-purpose I/O port control registers.
- The Program Memory contains addresses for all memory locations having executable code and/or data.
- The Data Memory consists of the addresses for all memory locations that hold only data.

These three address spaces are covered briefly in the following subsections. For more information on eZ8 CPU and its address space, refer to *eZ8™ CPU Core User Manual (UM0128)* available for download at www.zilog.com.

Register File

The Register File address space in the 64K Series is 4 KB (4096 bytes). The Register File is composed of two sections—control registers and general-purpose registers. When instructions are executed, registers are read from when defined as sources and written to when defined as destinations. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4 KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256-byte control register section are reserved (unavailable). Reading from an reserved Register File addresses returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The 64K Series provide 2 KB to 4 KB of on-chip RAM depending upon the device. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect. To determine the amount of RAM available for the specific 64K Series device, see Part Selection Guide on page 2.

Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FCD	Interrupt Edge Select	IRQES	00	78
FCE	Interrupt Port Select	IRQPS	00	78
FCF	Interrupt Control	IRQCTL	00	79
GPIO Port A				
FD0	Port A Address	PAADDR	00	61
FD1	Port A Control	PACTL	00	62
FD2	Port A Input Data	PAIN	XX	66
FD3	Port A Output Data	PAOUT	00	66
GPIO Port B				
FD4	Port B Address	PBADDR	00	61
FD5	Port B Control	PBCTL	00	62
FD6	Port B Input Data	PBIN	XX	66
FD7	Port B Output Data	PBOUT	00	66
GPIO Port C				
FD8	Port C Address	PCADDR	00	61
FD9	Port C Control	PCCTL	00	62
FDA	Port C Input Data	PCIN	XX	66
FDB	Port C Output Data	PCOUT	00	66
GPIO Port D				
FDC	Port D Address	PDADDR	00	61
FDD	Port D Control	PDCTL	00	62
FDE	Port D Input Data	PDIN	XX	66
FDF	Port D Output Data	PDOUT	00	66
GPIO Port E				
FE0	Port E Address	PEADDR	00	61
FE1	Port E Control	PECTL	00	62
FE2	Port E Input Data	PEIN	XX	66
FE3	Port E Output Data	PEOUT	00	66
GPIO Port F				
FE4	Port F Address	PFADDR	00	61
FE5	Port F Control	PFCTL	00	62
FE6	Port F Input Data	PFIN	XX	66
FE7	Port F Output Data	PFOUT	00	66
GPIO Port G				
FE8	Port G Address	PGADDR	00	61
FE9	Port G Control	PGCTL	00	62
FEA	Port G Input Data	PGIN	XX	66
FEB	Port G Output Data	PGOUT	00	66
GPIO Port H				
FEC	Port H Address	PHADDR	00	61
FED	Port H Control	PHCTL	00	62
FEE	Port H Input Data	PHIN	XX	66

Table 23. Interrupt Vectors in Order of Priority

Priority	Program Memory Vector Address	Interrupt Source
Highest	0002H	Reset (not an interrupt)
	0004H	Watchdog Timer (see Watchdog Timer on page 97)
	0006H	Illegal Instruction Trap (not an interrupt)
	0008H	Timer 2
	000AH	Timer 1
	000CH	Timer 0
	000EH	UART 0 receiver
	0010H	UART 0 transmitter
	0012H	I^2C
	0014H	SPI
	0016H	ADC
	0018H	Port A7 or Port D7, rising or falling input edge
	001AH	Port A6 or Port D6, rising or falling input edge
	001CH	Port A5 or Port D5, rising or falling input edge
	001EH	Port A4 or Port D4, rising or falling input edge
	0020H	Port A3 or Port D3, rising or falling input edge
	0022H	Port A2 or Port D2, rising or falling input edge
	0024H	Port A1 or Port D1, rising or falling input edge
	0026H	Port A0 or Port D0, rising or falling input edge
	0028H	Timer 3 *(not available in 44-pin packages)*
	002AH	UART 1 receiver
	002CH	UART 1 transmitter
	002EH	DMA
	0030H	Port C3, both input edges
	0032H	Port C2, both input edges
	0034H	Port C1, both input edges
Lowest	0036H	Port C0, both input edges

6. Write to the Timer Control 1 register to enable the timer.
7. Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge.

In m/COMPARE mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

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

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.

Timer Output Signal Operation

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

Timer Control Register Definitions

Timers 0-2 are available in all packages. Timer 3 is only available in the 64-, 68-, and 80-pin packages.

Timer 0-3 High and Low Byte Registers

The Timer 0-3 High and Low Byte (TxH and TxL) registers (see Table 39 and Table 40 on page 91) contain the current 16-bit timer count value. When the timer is enabled, a read from TxH causes the value in TxL to be stored in a temporary holding register. A read from TMRL always returns this temporary register when the timers are enabled. When the timer is disabled, reads from the TMRL reads the register directly.

Writing to the Timer High and Low Byte registers while the timer is enabled is not recommended. There are no temporary holding registers available for write operations, so simultaneous 16-bit writes are not possible. If either the Timer High or Low Byte registers are written during counting, the 8-bit written value is placed in the counter (High or Low Byte) at the next clock edge. The counter continues counting from the new value.

Timer 3 is unavailable in the 40- and 44-pin packages.

110 = Divide by 64
111 = Divide by 128

TMODE—TIMER mode
000 = ONE-SHOT mode
001 = CONTINUOUS mode
010 = COUNTER mode
011 = PWM mode
100 = CAPTURE mode
101 = COMPARE mode
110 = GATED mode
111 = CAPTURE/COMPARE mode

PE—Parity Error
This bit indicates that a parity error has occurred. Reading the UART Receive Data register clears this bit.
0 = No parity error occurred.
1 = A parity error occurred.

OE—Overrun Error
This bit indicates that an overrun error has occurred. An overrun occurs when new data is received and the UART Receive Data register has not been read. If the RDA bit is reset to 0, then reading the UART Receive Data register clears this bit.
0 = No overrun error occurred.
1 = An overrun error occurred.

FE—Framing Error
This bit indicates that a framing error (no Stop bit following data reception) was detected. Reading the UART Receive Data register clears this bit.
0 = No framing error occurred.
1 = A framing error occurred.

BRKD—Break Detect
This bit indicates that a break occurred. If the data bits, parity/multiprocessor bit, and Stop bit(s) are all zeros then this bit is set to 1. Reading the UART Receive Data register clears this bit.
0 = No break occurred.
1 = A break occurred.

TDRE—Transmitter Data Register Empty
This bit indicates that the UART Transmit Data register is empty and ready for additional data. Writing to the UART Transmit Data register resets this bit.
0 = Do not write to the UART Transmit Data register.
1 = The UART Transmit Data register is ready to receive an additional byte to be transmitted.

TXE—Transmitter Empty
This bit indicates that the transmit shift register is empty and character transmission is finished.
0 = Data is currently transmitting.
1 = Transmission is complete.

CTS—$\overline{\text{CTS}}$ signal
When this bit is read it returns the level of the $\overline{\text{CTS}}$ signal.

UART Status 1 Register

This register contains multiprocessor control and status bits.

Table 55. UART Status 1 Register (UxSTAT1)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved						NEWFRM	MPRX
RESET	0							
R/W	R				R/W		R	
ADDR	F44H and F4CH							

Reserved—Must be 0.

NEWFRM—Status bit denoting the start of a new frame. Reading the UART Receive Data register resets this bit to 0.
0 = The current byte is not the first data byte of a new frame.
1 = The current byte is the first data byte of a new frame.

MPRX—Multiprocessor Receive
Returns the value of the last multiprocessor bit received. Reading from the UART Receive Data register resets this bit to 0.

UART Control 0 and Control 1 Registers

The UART Control 0 and Control 1 registers (see Table 56 and Table 57 on page 118) configure the properties of the UART's transmit and receive operations. The UART Control registers must not been written while the UART is enabled.

Table 56. UART Control 0 Register (UxCTL0)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
RESET	0							
R/W	R/W							
ADDR	F42H and F4AH							

TEN—Transmit Enable
This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is low and the CTSE bit is 1, the transmitter is enabled.
0 = Transmitter disabled.
1 = Transmitter enabled.

Serial Peripheral Interface

Overview

The Serial Peripheral Interface is a synchronous interface allowing several SPI-type devices to be interconnected. SPI-compatible devices include EEPROMs, Analog-to-Digital Converters, and ISDN devices. Features of the SPI include:

- Full-duplex, synchronous, character-oriented communication
- Four-wire interface
- Data transfers rates up to a maximum of one-half the system clock frequency
- Error detection
- Dedicated Baud Rate Generator

Architecture

The SPI may be configured as either a Master (in single or multi-master systems) or a Slave as displayed in Figure 22 through Figure 24.

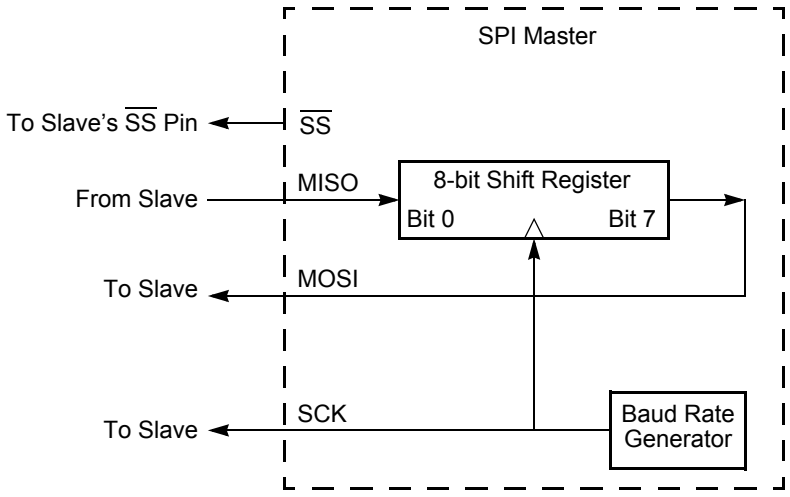


Figure 22. SPI Configured as a Master in a Single Master, Single Slave System

16. If the I²C slave sends an acknowledge by pulling the SDA signal low during the next high period of SCL, the I²C Controller sets the ACK bit in the I²C Status register. Continue with step 17.

    If the slave does not acknowledge the second address byte or one of the data bytes, the I²C Controller sets the NCKI bit and clears the ACK bit in the I²C Status register. Software responds to the Not Acknowledge interrupt by setting the STOP and FLUSH bits and clearing the TXI bit. The I²C Controller sends the STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore the following steps).
17. The I²C Controller shifts the data out by the SDA signal. After the first bit is sent, the Transmit interrupt is asserted.
18. If more bytes remain to be sent, return to step 14.
19. If the last byte is currently being sent, software sets the STOP bit of the I²C Control register (or START bit to initiate a new transaction). In the STOP case, software also clears the TXI bit of the I²C Control register at the same time.
20. The I²C Controller completes transmission of the last data byte on the SDA signal.
21. The slave may either Acknowledge or Not Acknowledge the last byte. Because either the STOP or START bit is already set, the NCKI interrupt does not occur.
22. The I²C Controller sends the STOP (or RESTART) condition to the I²C bus and clears the STOP (or START) bit.

Read Transaction with a 7-Bit Address

Figure 32 displays the data transfer format for a read operation to a 7-bit addressed slave. The shaded regions indicate data transferred from the I²C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I²C Controller.

S	Slave Address	R = 1	A	Data	A	Data	$\overline{A}$	P/S

Figure 32. Receive Data Transfer Format for a 7-Bit Addressed Slave

Follow the steps below for a read operation to a 7-bit addressed slave:

1. Software writes the I²C Data register with a 7-bit slave address plus the read bit (=1).
2. Software asserts the START bit of the I²C Control register.
3. If this is a single byte transfer, 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 is sent to the I²C slave.

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.

finish the interrupt service routine it may be in and return the BRK instruction. When the CPU returns to the BRK instruction it was previously looping on, it automatically sets the DBGMODE bit and enter DEBUG mode.

Software detects that the majority of the OCD commands are still disabled when the eZ8™ CPU is looping on a BRK instruction. The eZ8 CPU must be stopped and the part must be in DEBUG mode before these commands can be issued.

Breakpoints in Flash Memory

The BRK instruction is opcode 00H, which corresponds to the fully programmed state of a byte in Flash memory. To implement a Breakpoint, write 00H to the desired address, overwriting the current instruction. To remove a Breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

On-Chip Debugger Commands

The host communicates to the On-Chip Debugger by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG mode, all OCD commands become available unless the user code and control registers are protected by programming the Read Protect Option Bit (RP). The Read Protect Option Bit prevents the code in memory from being read out of the 64K Series products. When this option is enabled, several of the OCD commands are disabled. Table 101 contains a summary of the On-Chip Debugger commands. Each OCD command is described in detail in the bulleted list following Table 101.
Table 101 indicates those commands that operate when the device is not in DEBUG mode (normal operation) and those commands that are disabled by programming the Read Protect Option Bit.

Table 101. On-Chip Debugger Commands

Debug Command	Command Byte	Enabled when NOT in DEBUG mode?	Disabled by Read Protect Option Bit
Read OCD Revision	00H	Yes	-
Read OCD Status Register	02H	Yes	-
Read Runtime Counter	03H	-	-
Write OCD Control Register	04H	Yes	Cannot clear DBGMODE bit
Read OCD Control Register	05H	Yes	-

```
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

- **Write Program Memory (0AH)**—The Write Program Memory command writes data to Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Program Memory (0BH)**—The Read Program Memory command reads data from Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1-65536 bytes at a time (65536 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, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

- **Write Data Memory (0CH)**—The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). If the device is not in DEBUG mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

- **Read Data Memory (0DH)**—The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the device is not in DEBUG mode, this command returns FFH for the data.

```
DBG ← 0DH
DBG ← Data Memory Address[15:8]
```

DC Characteristics

Table 106 lists the DC characteristics of the 64K Series products. All voltages are referenced to V_{SS}, the primary system ground.

Table 106. DC Characteristics

Symbol	Parameter	T_A = –40 °C to 125 °C			Units	Conditions
		Minimum	Typical	Maximum		
V_{DD}	Supply Voltage	3.0	–	3.6	V	
V_{IL1}	Low Level Input Voltage	-0.3	–	0.3*V_{DD}	V	For all input pins except $\overline{\text{RESET}}$, DBG, XIN
V_{IL2}	Low Level Input Voltage	-0.3	–	0.2*V_{DD}	V	For $\overline{\text{RESET}}$, DBG, and XIN.
V_{IH1}	High Level Input Voltage	0.7*V_{DD}	–	5.5	V	Port A, C, D, E, F, and G pins.
V_{IH2}	High Level Input Voltage	0.7*V_{DD}	–	V_{DD}+0.3	V	Port B and H pins.
V_{IH3}	High Level Input Voltage	0.8*V_{DD}	–	V_{DD}+0.3	V	$\overline{\text{RESET}}$, DBG, and XIN pins
V_{OL1}	Low Level Output Voltage Standard Drive	–	–	0.4	V	I_{OL} = 2 mA; VDD = 3.0 V High Output Drive disabled.
V_{OH1}	High Level Output Voltage Standard Drive	2.4	–	–	V	I_{OH} = -2 mA; VDD = 3.0 V High Output Drive disabled.
V_{OL2}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 20 mA; VDD = 3.3 V High Output Drive enabled T_A = -40 °C to +70 °C
V_{OH2}	High Level Output Voltage High Drive	2.4	–	–	V	I_{OH} = -20 mA; VDD = 3.3 V High Output Drive enabled; T_A = -40 °C to +70 °C
V_{OL3}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 15 mA; VDD = 3.3 V High Output Drive enabled; T_A = +70 °C to +105 °C
V_{OH3}	High Level Output Voltage High Drive	2.4	–	–	V	I_{OH} = 15 mA; VDD = 3.3 V High Output Drive enabled; T_A = +70 °C to +105 °C
V_{RAM}	RAM Data Retention	0.7	–	–	V	
I_{IL}	Input Leakage Current	-5	–	+5	µA	V_{DD} = 3.6 V; V_{IN} = VDD or VSS[1]
I_{TL}	Tri-State Leakage Current	-5	–	+5	µA	V_{DD} = 3.6 V

SPI Master Mode Timing

Figure 53 and Table 117 provide timing information for SPI Master mode pins. Timing is shown with SCK rising edge used to source MOSI output data, SCK falling edge used to sample MISO input data. Timing on the SS output pin(s) is controlled by software.

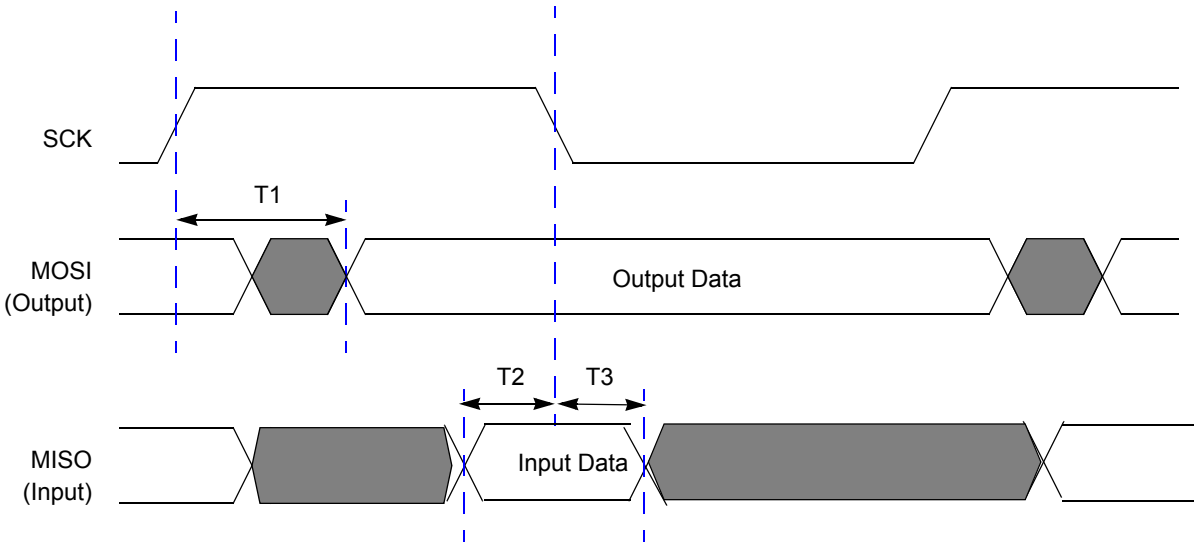


Figure 53. SPI Master Mode Timing

Table 117. SPI Master Mode Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
SPI Master			
T_1	SCK Rise to MOSI output Valid Delay	-5	+5
T_2	MISO input to SCK (receive edge) Setup Time	20	
T_3	MISO input to SCK (receive edge) Hold Time	0	

I²C Timing

Figure 55 and Table 119 provide timing information for I²C pins.

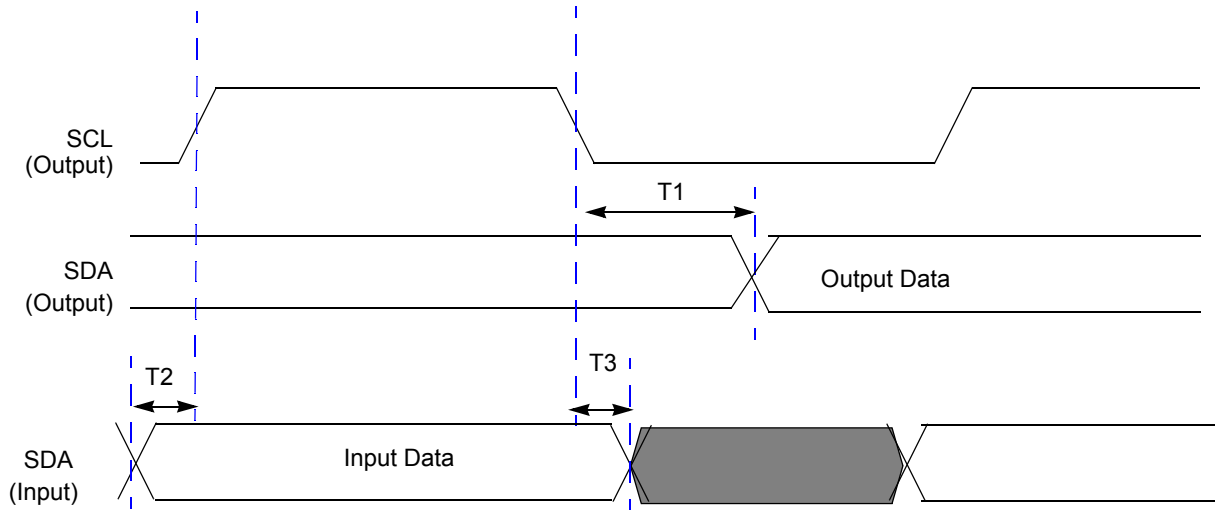
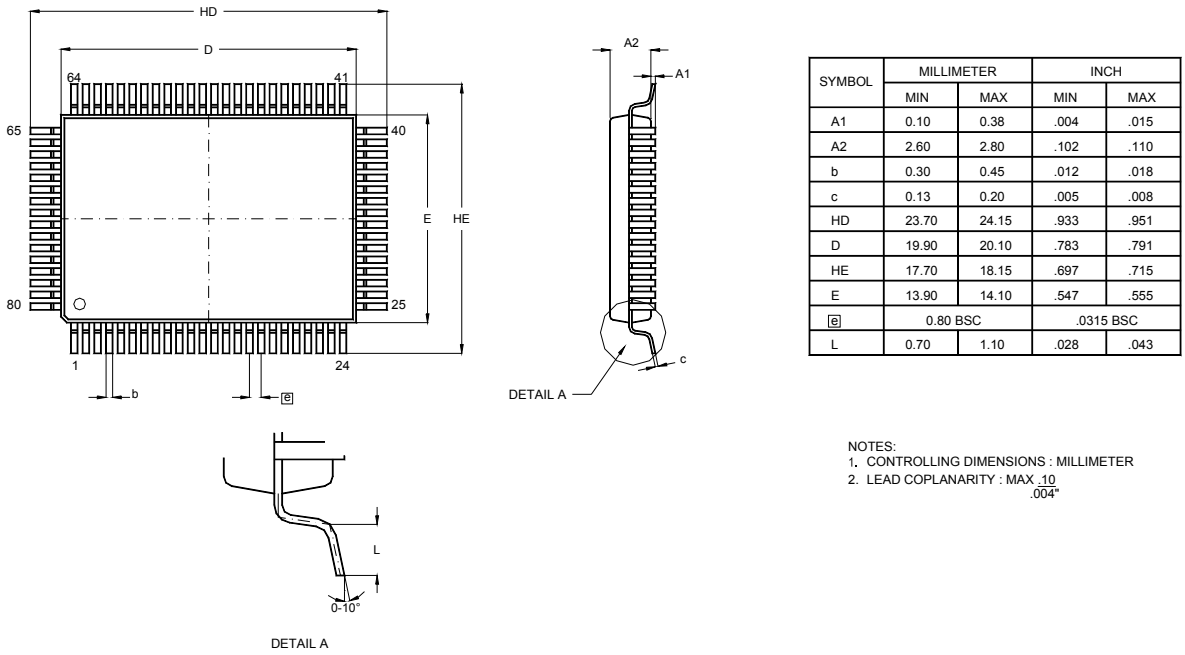


Figure 55. I²C Timing

Table 119. I²C Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
I²C			
T_1	SCL Fall to SDA output delay		SCL period/4
T_2	SDA Input to SCL rising edge Setup Time	0	
T_3	SDA Input to SCL falling edge Hold Time	0	

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



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.10	0.38	.004	.015
A2	2.60	2.80	.102	.110
b	0.30	0.45	.012	.018
c	0.13	0.20	.005	.008
HD	23.70	24.15	.933	.951
D	19.90	20.10	.783	.791
HE	17.70	18.15	.697	.715
E	13.90	14.10	.547	.555
e	0.80 BSC		.0315 BSC	
L	0.70	1.10	.028	.043

NOTES:
1. CONTROLLING DIMENSIONS : MILLIMETER
2. LEAD COPLANARITY : MAX .10 / .004"

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

For technical and customer support, hardware and software development tools, refer to the Zilog® website at www.zilog.com. The latest released version of ZDS can be downloaded from this website.

Part Number Suffix Designations