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	eZ80
Core Size	8-Bit
Speed	20MHz
Connectivity	EBI/EMI, I²C, IrDA, SPI, UART/USART
Peripherals	DMA, WDT
Number of I/O	24
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80l92az020sg

- Four other block transfer instructions are modified to improve performance related to the eZ80190 device. These modified instructions are:
  - IND2R (input from I/O, decrement the memory address, decrement the I/O address, and repeat).
  - INI2R (input from I/O, increment the memory address, increment the I/O address, and repeat).
  - OTD2R (output to I/O, decrement the memory address, decrement the I/O address, and repeat).
  - OTI2R (output to I/O, increment the memory address, increment the I/O address, and repeat).

For more information on the eZ80 CPU, its instruction set, and eZ80 programming, refer to the *eZ80® CPU User Manual.* For more information on the eZ80190, refer to the *eZ80190 Product Specification*.

Low-Power Modes

The eZ80L92 MCU provides a range of power-saving features. The highest level of power reduction is provided by SLEEP mode. The next level of power reduction is provided by the HALT instruction. The lowest level of power reduction is provided by the clock peripheral power-down registers.

SLEEP Mode

Execution of the eZ80 CPU's SLP instruction places the eZ80L92 MCU into SLEEP mode. In SLEEP mode, the operating characteristics are:

- Primary crystal oscillator is disabled.
- System clock is disabled.
- eZ80 CPU is idle.
- Program counter (PC) stops incrementing.
- 32 kHz crystal oscillator continues to operate and drives the Real-Time Clock and the Watchdog Timer (if WDT is configured to operate from the 32 kHz oscillator.)

The eZ80 CPU can be brought out of SLEEP mode by any of the following operations:

- RESET through the external $\overline{\text{RESET}}$ pin driven Low.
- RESET through a Real-Time Clock alarm.
- RESET through a Watchdog Timer time-out (if running off of the 32 kHz oscillator and configured to generate a RESET upon time-out).
- RESET through execution of a Debug RESET command.

After exiting SLEEP mode, the standard RESET delay occurs to allow the primary crystal oscillator to stabilize. See Reset on page 33.

HALT Mode

Execution of the eZ80 CPU's HALT instruction places the eZ80L92 MCU into HALT mode. In HALT mode, the operating characteristics are:

- Primary crystal oscillator is enabled and continues to operate.
- System clock is enabled and continues to operate.
- eZ80 CPU is idle.
- Program counter stops incrementing.

Chip Selects and Wait States

The eZ80L92 MCU generates four Chip Selects for external devices. Each Chip Select is programmed to access either memory space or I/O space. The Memory Chip Selects can be individually programmed on a 64 KB boundary. Each I/O Chip Select can choose a 256-byte section of I/O space. In addition, each Chip Select can be programmed for up to 7 wait states.

Memory and I/O Chip Selects

Each of the Chip Selects is enabled for either the memory address space or the I/O address space, but not both. To select the memory address space for a particular Chip Select, CSx_IO (CS*x*_CTL[4]) must be reset to 0. To select the I/O address space for a particular Chip Select, CSx_IO must be set to 1. After RESET, the default is for all Chip Selects to be configured for the memory address space. For either the memory address space or the I/O address space, the individual Chip Selects must be enabled by setting CSx_EN (CS*x*_CTL[3]) to 1.

Memory Chip Select Operation

Operation of each Memory Chip Selects is controlled by three control registers. To enable a particular Memory Chip Select, following conditions must be fulfilled:

- The Chip Select is enabled by setting CSx_EN to 1.
- The Chip Select is configured for Memory by clearing CSx_IO to 0.
- The address is in the associated Chip Select range:

  CS*x*_LBR[7:0] ≤ ADDR[23:16] ≤ CS*x*_UBR[7:0]

- No higher priority (lower number) Chip Select meets the above conditions.
- A memory access instruction must be executing.

If all the above conditions are met to generate a Memory Chip Select, then the following actions occur:

- The appropriate Chip Select—$\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, or $\overline{CS3}$—is asserted (driven Low).
- $\overline{MREQ}$ is asserted (driven Low).
- Depending upon the instruction, either $\overline{RD}$ or $\overline{WR}$ is asserted (driven Low).

If the upper and lower bounds are set to the same value (CS*x*_UBR = CS*x*_LBR), then a particular Chip Select is valid for a single 64 KB page.

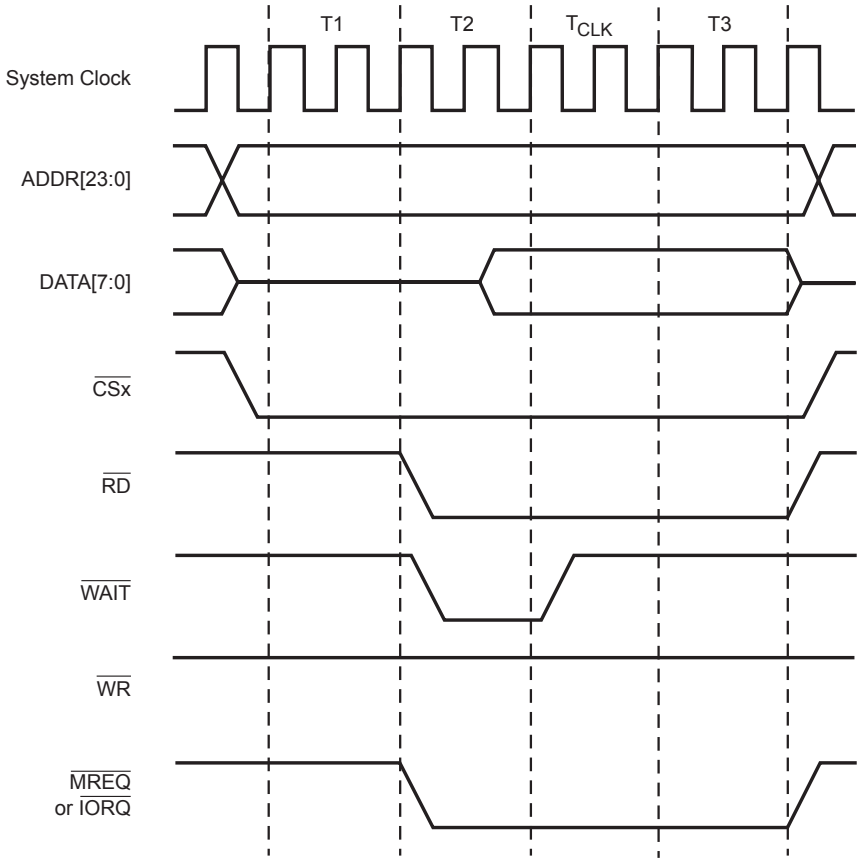


Figure 7. Z80® Bus Mode Read Timing Example

determine the source of the NMI event, provided that the last RESET was not caused by the WDT.

Watchdog Timer Registers

Watchdog Timer Control Register

The Watchdog Timer Control register, described in Table 27, is an 8-bit Read/Write register used to enable the Watchdog Timer, set the time-out period, indicate the source of the most recent RESET, and select the required operation upon WDT time-out.

Table 27. Watchdog Timer Control Register (WDT_CTL = 0093h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0/1	0	0	0	0	0
CPU Access	R/W	R/W	R	R/W	R/W	R	R/W	R/W
Note: R = Read only; R/W = Read/Write.								

Bit Position	Value	Description
7 WDT_EN	0	WDT is disabled.
	1	WDT is enabled. When enabled, the WDT cannot be disabled without a full RESET.
6 NMI_OUT	0	WDT time-out resets the CPU.
	1	WDT time-out generates a nonmaskable interrupt (NMI) to the CPU.
5 RST_FLAG*	0	RESET caused by external full-chip reset or ZDI reset.
	1	RESET caused by WDT time-out. This flag is set by the WDT time-out, even if the NMI_OUT flag is set to 1. The CPU can poll this bit to determine the source of the RESET or NMI.
[4:3] WDT_CLK	00	WDT clock source is system clock.
	01	WDT clock source is Real-Time Clock source (32 kHz on-chip oscillator or 50/60Hz input as set by RTC_CTRL[4]) .
	10	Reserved.
	11	Reserved.
2 RESERVED	0	Reserved.

Note: *RST_FLAG is only cleared by a non-WDT RESET.

of the timer is achieved by loading 0001h. Maximum duration is achieved by loading 0000h, because the timer first rolls over to FFFFh and then continues counting down to 0000h.

The time-out period of the PRT is returned by the following equation:

$$\text{PRT Time-Out Period} = \frac{\text{Clock Divider Ratio x Reload Value}}{\text{System Clock Frequency}}$$

To calculate the time-out period with the above equation when using an initial value of 0000h, enter a reload value of 65536 (FFFFh + 1).

Minimum time-out duration is 4 times longer than the input clock period and is generated by setting the clock divider ratio to 1:4 and the reload value to 0001h. Maximum time-out duration is 2^{24} (16,777,216) times longer than the input clock period and is generated by setting the clock divider ratio to 1:256 and the reload value to 0000h.

SINGLE PASS Mode

In SINGLE PASS mode, when the end-of-count value, 0000h, is reached, counting halts, the timer is disabled, and the PRT_EN bit resets to 0. To restart the timer, the CPU must reenable the timer by setting the PRT_EN bit to 1. An example of a PRT operating in SINGLE PASS mode is illustrated in Figure 19. Timer register information is indicated in Table 29.

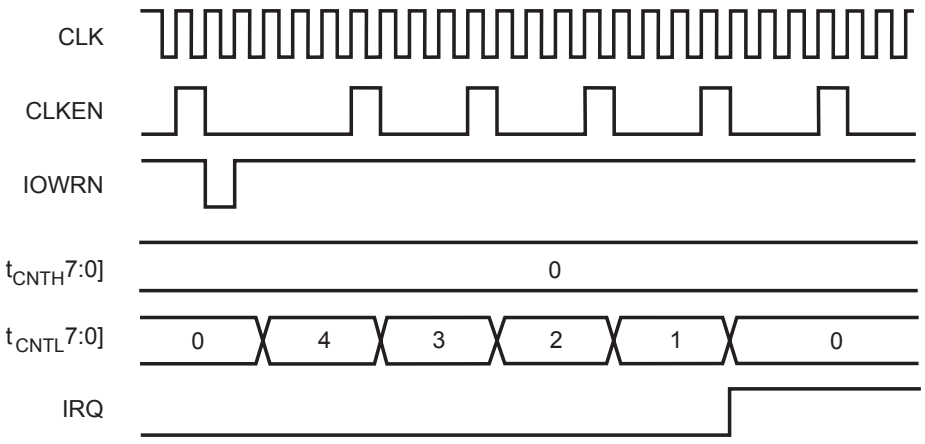


Figure 19. PRT SINGLE PASS Mode Operation Example

Table 29. PRT SINGLE PASS Mode Operation Example

Parameter	Control Register(s)	Value
PRT Enabled	TMRx_CTL[0]	1
Reload and Restart Enabled	TMRx_CTL[1]	1
PRT Clock Divider = 4	TMRx_CTL[3:2]	00b
SINGLE PASS Mode	TMRx_CTL[4]	0
PRT Interrupt Enabled	TMRx_CTL[6]	1
PRT Reload Value	{TMRx_RR_H, TMRx_RR_L}	0004h

CONTINUOUS Mode

In CONTINUOUS mode, when the end-of-count value, 0000h, is reached, the timer automatically reloads the 16-bit start value from the Timer Reload registers, TMRx_RR_H and TMRx_RR_L. Downcounting continues on the next clock edge. In CONTINUOUS mode, the PRT continues to count until disabled. An example of a PRT operating in CONTINUOUS mode is illustrated in Figure 20. Timer register information is indicated in Table 30.

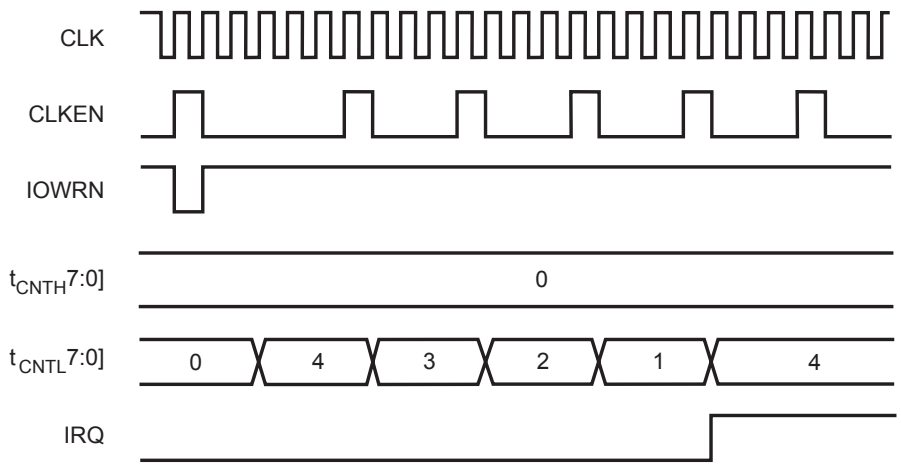


Figure 20. PRT CONTINUOUS Mode Operation Example

Bit Position	Value	Description
[7:0] TMRx_DR_H	00h–FFh	These bits represent the High byte of the 2-byte timer data value, {TMRx_DR_H[7:0], TMRx_DR_L[7:0]}. Bit 7 is bit 15 (msb) of the 16-bit timer data value. Bit 0 is bit 8 of the 16-bit timer data value.

Timer Reload Registers—Low Byte

The Timer Reload Register—Low Byte, described in Table 35, stores the least significant byte (LSB) of the 2-byte timer reload value. In CONTINUOUS mode, the timer reload value is reloaded into the timer upon end-of-count. When RST_EN (TMRx_CTL[1]) is set to 1 to enable the automatic reload and restart function, the timer reload value is written to the timer on the next rising edge of the clock.

Note: The Timer Data registers and Timer Reload registers share the same address space.

Table 35. Timer Reload Registers—Low Byte (TMR0_RR_L = 0081h, TMR1_RR_L = 0084h, TMR2_RR_L = 0087h, TMR3_RR_L = 008Ah, TMR4_RR_L = 008Dh, or TMR5_RR_L = 0090h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	W	W	W	W	W	W	W	W
Note: W = Write only.								

Bit Position	Value	Description
[7:0] TMRx_RR_L	00h–FFh	These bits represent the Low byte of the 2-byte timer reload value, {TMRx_RR_H[7:0], TMRx_RR_L[7:0]}. Bit 7 is bit 7 of the 16-bit timer reload value. Bit 0 is bit 0 (lsb) of the 16-bit timer reload value.

Real Time Clock Month Register

This register contains the current month count. See Table 43.

Table 43. Real Time Clock Month Register (RTC_MON = 00E5h)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Note: X = Unchanged by RESET; R/W* = Read-only if RTC locked, Read/Write if RTC unlocked.								

Binary-Coded-Decimal Operation (BCD_EN = 1)		
Bit Position	**Value**	**Description**
[7:4] TENS_MON	0–1	The tens digit of the current month count.
[3:0] MON	0–9	The ones digit of the current month count.

Binary Operation (BCD_EN = 0)		
Bit Position	**Value**	**Description**
[7:0] MON	01h–0Ch	The current month count.

Real Time Clock Century Register

This register contains the current century count. See Table 45.

Table 45. Real Time Clock Century Register (RTC_CEN = 00E7h)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Note: X = Unchanged by RESET; R/W* = Read-only if RTC locked, Read/Write if RTC unlocked.								

Binary-Coded-Decimal Operation (BCD_EN = 1)		
Bit Position	**Value**	**Description**
[7:4] TENS_CEN	0–9	The tens digit of the current century count.
[3:0] CEN	0–9	The ones digit of the current century count.

Binary Operation (BCD_EN = 0)		
Bit Position	**Value**	**Description**
[7:0] CEN	00h–63h	The current century count.

Real Time Clock Alarm Hours Register

This register contains the alarm hours value. See Table 48.

Table 48. Real Time Clock Alarm Hours Register (RTC_AHRS = 00EAh)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Note: X = Unchanged by RESET; R/W = Read/Write.								

Binary-Coded-Decimal Operation (BCD_EN = 1)

Bit Position	Value	Description
[7:4] ATEN_HRS	0–2	The tens digit of the alarm hours value.
[3:0] AHRS	0–9	The ones digit of the alarm hours value.

Binary Operation (BCD_EN = 0)

Bit Position	Value	Description
[7:0] AHRS	00h–17h	The alarm hours value.

Real Time Clock Alarm Day-of-the-Week Register

This register contains the alarm day-of-the-week value. See Table 49.

Table 49. Real Time Clock Alarm Day-of-the-Week Register (RTC_ADOW = 00EBh)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	X	X	X	X
CPU Access	R	R	R	R	R/W*	R/W*	R/W*	R/W*

Note: X = Unchanged by RESET; R = Read Only; R/W* = Read-only if RTC locked, Read/Write if RTC unlocked.

Binary-Coded-Decimal Operation (BCD_EN = 1)

Bit Position	Value	Description
[7:4]	0000	Reserved.
[3:0] ADOW	1-7	The alarm day-of-the-week.value.

Binary Operation (BCD_EN = 0)

Bit Position	Value	Description
[7:4]	0000	Reserved.
[3:0] ADOW	01h–07h	The alarm day-of-the-week value.

CLK_SEL and FREQ_SEL select the RTC clock source. If the 32 kHz crystal option is selected the oscillator is enabled and the internal prescaler is set to divide by 32768. If the power-line frequency option is selected, the prescale value is set by FREQ_SEL, and the 32 kHz oscillator is disabled. See Table 51.

Table 51. Real Time Clock Control Register (RTC_CTRL = 00EDh)

Bit	7	6	5	4	3	2	1	0
Reset	X	0	X	X	X	X	0/1	0
CPU Access	R	R/W	R/W	R/W	R/W	R	R	R/W
Note: X = Unchanged by RESET; R = Read-only; R/W = Read/Write.								

Bit Position	Value	Description
7 ALARM	0	Alarm interrupt is inactive.
	1	Alarm interrupt is active.
6 INT_EN	0	Interrupt on alarm condition is disabled.
	1	Interrupt on alarm condition is enabled.
5 BCD_EN	0	RTC count and alarm value registers are binary.
	1	RTC count and alarm value registers are binary-coded decimal (BCD).
4 CLK_SEL	0	RTC clock source is crystal oscillator output (32768 Hz). On-chip 32768 Hz oscillator is enabled.
	1	RTC clock source is power-line frequency input. On-chip 32768 Hz oscillator is disabled.
3 FREQ_SEL	0	Power-line frequency is 60 Hz.
	1	Power-line frequency is 50 Hz.
2	0	Reserved.
1 SLP_WAKE	0	RTC does not generate a Sleep-Mode Recovery reset.
	1	RTC Alarm generates a Sleep-Mode Recovery reset.
0 RTC_UNLOCK	0	RTC count registers are locked to prevent write access. RTC counter is enabled.
	1	RTC count registers are unlocked to allow write access. RTC counter is disabled.

SPI Signals

The four basic SPI signals are:

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

The following section describes the SPI signals. 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.

Slave Select

The active Low Slave Select ($\overline{SS}$) input signal is used to select the SPI as a slave device. It must be Low prior to all data communication and must stay Low for the duration of the data transfer.

The $\overline{SS}$ input signal must be High for the SPI to operate as a master device. If the $\overline{SS}$ signal goes Low, a Mode Fault error flag (MODF) is set in the SPI_SR register. See the SPI Status Register (SPI_SR) on page 136 for more information.

When the clock phase (CPHA) is set to 0, the shift clock is the logical OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go High between successive characters in an SPI message. When CPHA is set to 1, $\overline{SS}$ can remain Low for several SPI characters. In cases where there is only one SPI slave, its $\overline{SS}$ line could be tied Low as long as CPHA is set to 1. See the SPI Control Register (SPI_CTL) on page 135 for more information about CPHA.

Table 81. I2C Slave Address Register (I2C_SAR = 00C8h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Note: R/W = Read/Write.								

Bit Position	Value	Description
[7:1] SLA	00h–7Fh	7-bit slave address or upper 2 bits,I2C_SAR[2:1], of address when operating in 10-bit mode.
0 GCE	0	I^2C not enabled to recognize the General Call Address.
	1	I^2C enabled to recognize the General Call Address.

I^2C Extended Slave Address Register

The I2C_XSAR register is used in conjunction with the I2C_SAR register to provide 10-bit addressing of the I^2C when in SLAVE mode. The I2C_SAR value forms the lower 8 bits of the 10-bit slave address. The full 10-bit address is supplied by {I2C_SAR[2:1], I2C_XSAR[7:0]}.

When the register receives an address starting with F7h to F0h (I2C_SAR[7:3] = 11110b), the I^2C recognizes that a 10-bit slave addressing mode is being selected. The I^2C sends an ACK after receiving the I2C_XSAR byte (the device does not generate an interrupt at this point). After the next byte of the address (I2C_XSAR) is received, the I^2C generates an interrupt and goes into SLAVE mode. Then I2C_SAR[2:1] are used as the upper 2 bits for the 10-bit extended address. The full 10-bit address is supplied by {I2C_SAR[2:1], I2C_XSAR[7:0]}. See Table 82.

Table 82. I^2C Extended Slave Address Register (I2C_XSAR = 00C9h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Note: R/W = Read/Write.								

Bit Position	Value	Description
[7:0] SLAX	00h–FFh	Least significant 8 bits of the 10-bit extended slave address.

I²C Data Register

This register contains the data byte/slave address to be transmitted or the data byte just received. In transmit mode, the most significant bit of the byte is transmitted first. In receive mode, the first bit received is placed in the most significant bit of the register. After each byte is transmitted, the I2C_DR register contains the byte that is present on the bus in case a lost arbitration event occurs. See Table 83.

Table 83. I²C Data Register (I2C_DR = 00CAh)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Note: R/W = Read/Write.								

Bit Position	Value	Description
[7:0] DATA	00h–FFh	I²C data byte.

I²C Control Register

The I2C_CTL register is a control register that is used to control the interrupts and the master slave relationships on the I²C bus.

When the Interrupt Enable bit (IEN) is set to 1, the interrupt line goes High when the IFLG is set to 1. When IEN is cleared to 0, the interrupt line always remains Low.

When the Bus Enable bit (ENAB) is set to 0, the I²C bus inputs SCLx and SDAx are ignored and the I²C module does not respond to any address on the bus. When ENAB is set to 1, the I²C responds to calls to its slave address and to the general call address if the GCE bit (I2C_SAR[0]) is set to 1.

When the Master Mode Start bit (STA) is set to 1, the I²C enters MASTER mode and sends a START condition on the bus when the bus is free. If the STA bit is set to 1 when the I²C module is already in MASTER mode and one or more bytes are transmitted, then a repeated START condition is sent. If the STA bit is set to 1 when the I²C block is being

Bit Position	Value	Description
7 BRK_NEXT	0	The ZDI BREAK on the next CPU instruction is disabled. Clearing this bit releases the CPU from its current BREAK condition.
	1	The ZDI BREAK on the next CPU instruction is enabled. The CPU can use multibyte Op Codes and multibyte operands. break points only occur on the first Op Code in a multibyte Op Code instruction. If the ZCL pin is High and the ZDA pin is Low at the end of RESET, this bit is set to 1 and a BREAK occurs on the first instruction following the RESET. This bit is set automatically during ZDI BREAK on address match. A BREAK can also be forced by writing a 1 to this bit.
6 brk_addr3	0	The ZDI BREAK, upon matching break address 3, is disabled.
	1	The ZDI BREAK, upon matching break address 3, is enabled.
5 brk_addr2	0	The ZDI BREAK, upon matching break address 2, is disabled.
	1	The ZDI BREAK, upon matching break address 2, is enabled.
4 brk_addr1	0	The ZDI BREAK, upon matching break address 1, is disabled.
	1	The ZDI BREAK, upon matching break address 1, is enabled.
3 brk_addr0	0	The ZDI BREAK, upon matching break address 0, is disabled.
	1	The ZDI BREAK, upon matching break address 0, is enabled.
2 ign_low_1	0	The *Ignore the Low Byte* function of the ZDI Address Match 1 registers is disabled. If BRK_ADDR1 is set to 1, ZDI initiates a BREAK when the entire 24-bit address, ADDR[23:0], matches the 3-byte value {ZDI_ADDR1_U, ZDI_ADDR1_H, ZDI_ADDR1_L}.
	1	The *Ignore the Low Byte* function of the ZDI Address Match 1 registers is enabled. If BRK_ADDR1 is set to 1, ZDI initiates a BREAK when only the upper 2 bytes of the 24-bit address, ADDR[23:8], match the 2-byte value {ZDI_ADDR1_U, ZDI_ADDR1_H}. As a result, a BREAK can occur anywhere within a 256-byte page.

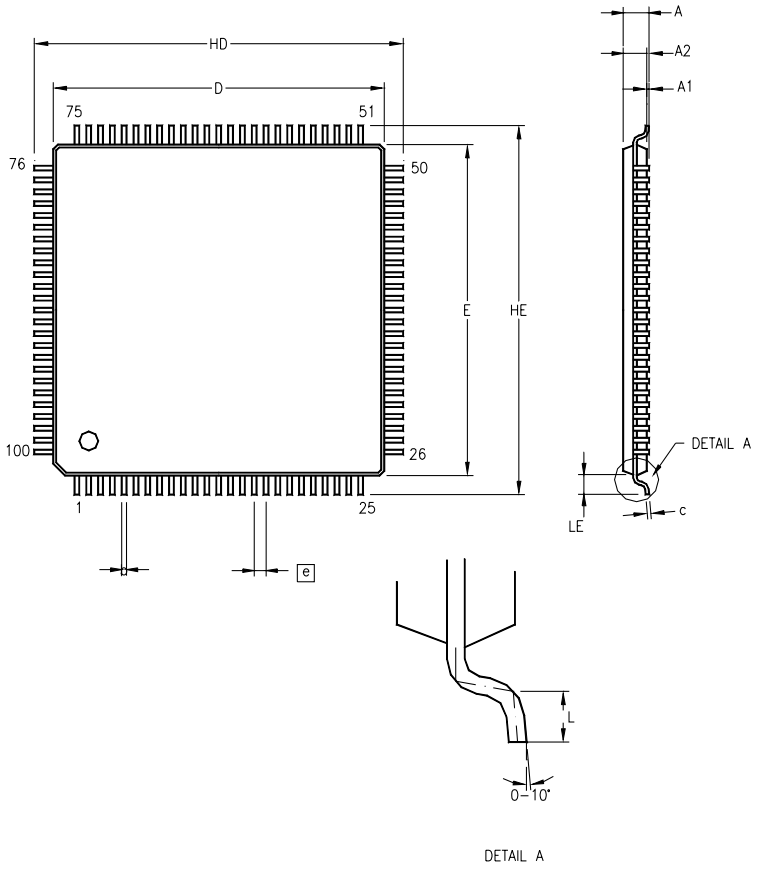
OCI Information Requests

For additional information regarding On-Chip Instrumentation, or to order OCI debug tools, please contact:

First Silicon Solutions, Inc.
5440 SW Westgate Drive, Suite 240
Portland, OR 97221
Phone: (503) 292-6730
Fax: (503) 292-5840
www.fs2.com

Packaging

Figure 56 illustrates the 100-pin low-profile quad flat pack (LQFP) package for the eZ80L92 devices.



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	1.35	1.60	.053	.063
A1	0.05	0.20	.002	.008
A2	1.30	1.50	.051	.059
b	0.15	0.26	.006	.010
c	0.10	0.20	.004	.008
HD	15.85	16.15	.624	.636
D	13.90	14.10	.547	.555
HE	15.85	16.15	.624	.636
E	13.90	14.10	.547	.555
e	0.50 BSC		.0197 BSC	
L	0.35	0.65	.014	.026
LE	0.90	1.10	.035	.043

1. CONTROLLING DIMENSIONS : MM
2. MAX COPLANARITY : .10mm / .004"

Figure 56. 100-Lead Plastic Low-Profile Quad Flat Package (LQFP)

Index

Numerics

A

B

C