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	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/ez80l92az020sc00tr

- Operating temperature range:
  - Standard, 0 ºC to +70 ºC
  - Extended, –40 ºC to +105 ºC

Note: All signals with an overline are active Low. For example, B/$\overline{W}$, for which WORD is active Low, and $\overline{B}$/W, for which BYTE is active Low.

Power connections follow these conventional descriptions.

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

Table 1. 100-Pin LQFP Pin Identification of eZ80L92 MCU (Continued)

Pin No	Symbol	Function	Signal Direction	Description
43	V_{DD}	Power Supply		Power Supply.
44	V_{SS}	Ground		Ground.
45	IORQ	Input/Output Request	Bidirectional, Active Low	$\overline{IORQ}$ indicates that the CPU is accessing a location in I/O space. $\overline{RD}$ and $\overline{WR}$ indicate the type of access. The eZ80L92 MCU does not drive this line during RESET. It is an input in bus acknowledge cycles.
46	MREQ	Memory Request	Bidirectional, Active Low	$\overline{MREQ}$ Low indicates that the CPU is accessing a location in memory. The $\overline{RD}$, $\overline{WR}$, and $\overline{INSTRD}$ signals indicate the type of access. The eZ80L92 MCU does not drive this line during RESET. It is an input in bus acknowledge cycles.
47	RD	Read	Output, Active Low	$\overline{RD}$ Low indicates that the eZ80L92 MCU is reading from the current address location. This pin is tristated during bus acknowledge cycles.
48	WR	Write	Output, Active Low	$\overline{WR}$ indicates that the CPU is writing to the current address location. This pin is tristated during bus acknowledge cycles.
49	INSTRD	Instruction Read Indicator	Output, Active Low	$\overline{INSTRD}$ (with $\overline{MREQ}$ and $\overline{RD}$) indicates the eZ80L92 MCU is fetching an instruction from memory. This pin is tristated during bus acknowledge cycles.
50	WAIT	WAIT Request	Input, Active Low	Driving the $\overline{WAIT}$ pin Low forces the CPU to wait additional clock cycles for an external peripheral or external memory to complete its Read or Write operation.
51	RESET	Reset	Schmitt Trigger Input, Active Low	This signal is used to initialize the eZ80L92 MCU. This input must be Low for a minimum of 3 system clock cycles, and must be held Low until the clock is stable. This input includes a Schmitt trigger to allow RC rise times.

Table 3. Register Map (Continued)

Address (hex)	Mnemonic	Name	Reset (hex)	CPU Access	Page No
Universal Asynchronous Receiver/Transmitter 1 (UART1) Block					
00D0	UART1_RBR	UART 1 Receive Buffer Register	XX	R	112
	UART1_THR	UART 1 Transmit Holding Register	XX	W	111
	UART1_BRG_L	UART 1 Baud Rate Generator Register—Low Byte	02	R/W	110
00D1	UART1_IER	UART 1 Interrupt Enable Register	00	R/W	112
	UART1_BRG_H	UART 1 Baud Rate Generator Register—High Byte	00	R/W	110
00D2	UART1_IIR	UART 1 Interrupt Identification Register	01	R	113
	UART1_FCTL	UART 1 FIFO Control Register	00	W	114
00D3	UART1_LCTL	UART 1 Line Control Register	00	R/W	115
00D4	UART1_MCTL	UART 1 Modem Control Register	00	R/W	117
Universal Asynchronous Receiver/Transmitter 1 (UART1) Block					
00D5	UART1_LSR	UART 1 Line Status Register	60	R/W	118
00D6	UART1_MSR	UART 1 Modem Status Register	XX	R/W	120
00D7	UART1_SPR	UART 1 Scratch Pad Register	00	R/W	122
Low-Power Control					
00DB	CLK_PPD1	Clock Peripheral Power-Down Register 1	00	R/W	36
00DC	CLK_PPD2	Clock Peripheral Power-Down Register 2	00	R/W	37
Real-Time Clock					
00E0	RTC_SEC	RTC Seconds Register[3]	XX	R/W	90
00E1	RTC_MIN	RTC Minutes Register	XX	R/W[3]	91
00E2	RTC_HRS	RTC Hours Register	XX	R/W[3]	92
00E3	RTC_DOW	RTC Day-of-the-Week Register	XX	R/W[3]	93
00E4	RTC_DOM	RTC Day-of-the-Month Register	XX	R/W[3]	94
00E5	RTC_MON	RTC Month Register	XX	R/W[3]	95
00E6	RTC_YR	RTC Year Register	XX	R/W[3]	96
00E7	RTC_CEN	RTC Century Register	XX	R/W[3]	97
00E8	RTC_ASEC	RTC Alarm Seconds Register	XX	R/W	98

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

- Non-maskable interrupt (NMI).
- Maskable interrupt.
- RESET through the external $\overline{\text{RESET}}$ pin driven Low.
- Watchdog Timer time-out (if configured to generate either an NMI or RESET upon time-out).
- RESET through execution of a Debug RESET command.

To minimize current in HALT mode, the system clock must be disabled for all unused on-chip peripherals through the Clock Peripheral Power-Down Registers.

Clock Peripheral Power-Down Registers

To reduce power, the Clock Peripheral Power-Down Registers allow the system clock to be disabled to unused on-chip peripherals. On RESET, all peripherals are enabled. The clock to unused peripherals can be disabled by setting the appropriate bit in the Clock Peripheral Power-Down Registers to 1. When powered down, the peripherals are completely disabled. To re-enable, the bit in the Clock Peripheral Power-Down Registers must be cleared to 0.

Many peripherals feature separate enable/disable control bits that must be appropriately set for operation. These peripheral specific enable/disable bits do not provide the same level of power reduction as the Clock Peripheral Power-Down Registers. When powered down, the standard peripheral control registers are not accessible for Read or Write access. See Table 4 and Table 5.

edge-triggered interrupt, writing a 1 to that pin's Port *x* Data register causes a reset of the edge-detected interrupt. You must set the bit in the Port *x* Data register to 1 before entering either single or dual edge-triggered interrupt mode for that port pin.

When configured for dual edge-triggered interrupt mode (GPIO Mode 6), both a rising and a falling edge on the pin cause an interrupt request to be sent to the eZ80 CPU.

When configured for single edge-triggered interrupt mode (GPIO Mode 9), the value in the Port *x* Data register determines if a positive or negative edge causes an interrupt request. A 0 in the Port *x* Data register bit sets the selected pin to generate an interrupt request for falling edges. A 1 in the Port *x* Data register bit sets the selected pin to generate an interrupt request for rising edges.

GPIO Control Registers

The 12 GPIO Control Registers operate in groups of four with a set for each Port (Ports B, C, and D). Each GPIO port features a Port Data register, Port Data Direction register, Port Alternate register 1, and Port Alternate register 2.

Port *x* Data Registers

When the port pins are configured for one of the output modes, the data written to the Port *x* Data Registers (see Table 7) are driven on the corresponding pins. In all modes, reading from the Port *x* Data registers always returns the current sampled value of the corresponding pins. When the port pins are configured as edge-triggered interrupt sources, writing 1 to the corresponding bit in the Port *x* Data register clears the interrupt signal that is sent to the eZ80 CPU. When the port pins are configured for edge-selectable interrupts or level-sensitive interrupts, the value written to the Port *x* Data register bit selects the interrupt edge or interrupt level. See Table 6.

Table 7. Port x Data Registers (PB_DR = 009Ah, PC_DR = 009Eh, PD_DR = 00A2h)

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 = Undefined; R/W = Read/Write.								

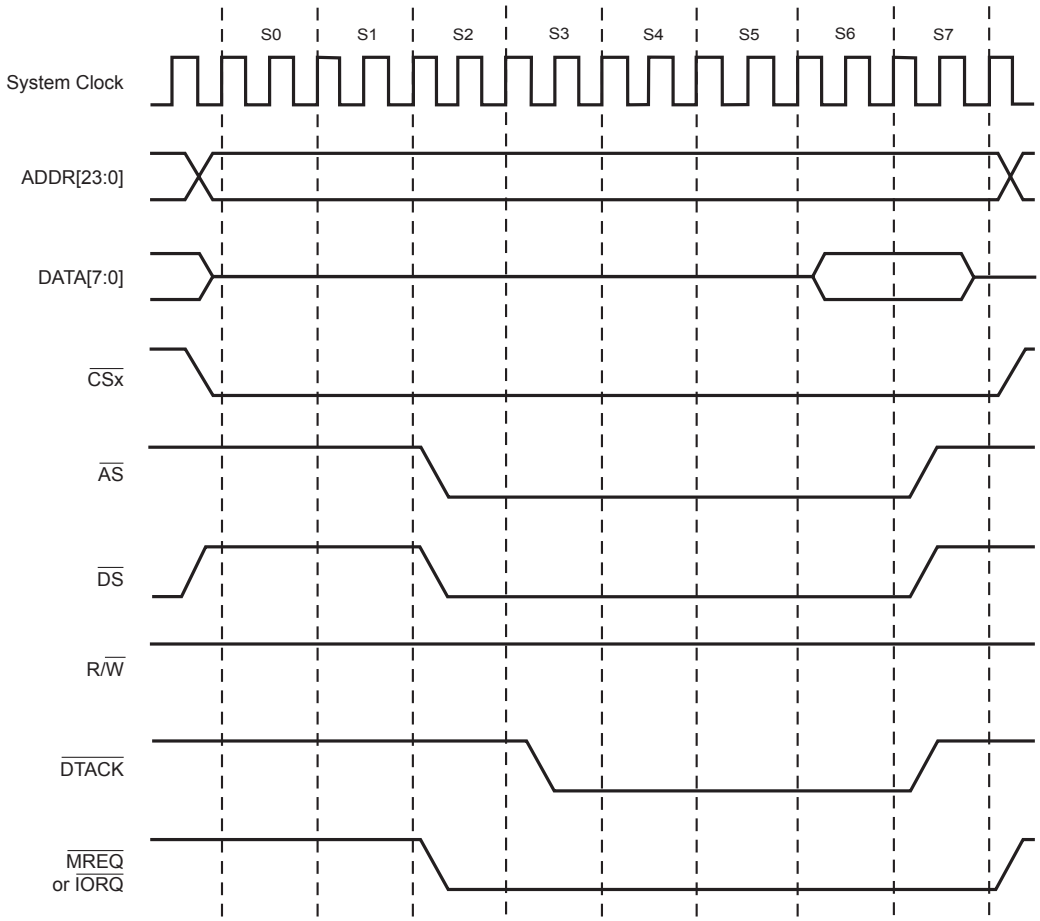


Figure 15. Motorola Bus Mode Read Timing Example

Table 37. Timer Input Source Select Register (TMR_ISS = 0092h)

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
[7:6] TMR3_IN	00	The timer counts at the system clock divided by the prescaler.
	01	The timer event input is the Real-Time Clock source (32 kHz or 50/60 Hz—refer to the Real Time Clock section on page 88 for details).
	10	The timer event input is the GPIO Port B pin 1.
	11	The timer event input is the GPIO Port B pin 1.
[5:4] TMR2_IN	00	The timer counts at the system clock divided by the prescaler.
	01	The timer event input is the Real-Time Clock source (32 kHz or 50/60 Hz—refer to the Real Time Clock section on page 88 for details).
	10	The timer event input is the GPIO Port B pin 0.
	11	The timer event input is the GPIO Port B pin 0.
[3:2] TMR1_IN	00	The timer counts at the system clock divided by the prescaler.
	01	The timer event input is the Real-Time Clock source (32 kHz or 50/60 Hz—refer to the Real Time Clock section on page 88 for details).
	10	The timer event input is the GPIO Port B pin 1.
	11	The timer event input is the GPIO Port B pin 1.
[1:0] TMR0_IN	00	Timer counts at system clock divided by prescaler.
	01	Timer event input is Real-Time Clock source (32 kHz or 50/60 Hz—refer to the Real Time Clock section on page 88 for details).
	10	The timer event input is the GPIO Port B pin 0.
	11	The timer event input is the GPIO Port B pin 0.

Real Time Clock Day-of-the-Month Register

This register contains the current day-of-the-month count. The RTC_DOM register begins counting at 01h. See Table 42.

Table 42. Real Time Clock Day-of-the-Month Register (RTC_DOM = 00E4h)

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_DOM	0–3	The tens digit of the current day-of-the-month count.
[3:0] DOM	0–9	The ones digit of the current day-of-the-month count.

Binary Operation (BCD_EN = 0)		
Bit Position	**Value**	**Description**
[7:0] DOM	01h–1Fh	The current day-of-the-month count.

The third source of a receiver interrupt is a line status error, indicating an error in byte reception. This error may result from:

- Incorrect received parity.
- Incorrect framing; that is, the stop bit is not detected by receiver at the end of the byte.
- Receiver over run condition.
- A BREAK condition being detected on the receive data input.

An interrupt due to one of the above conditions is cleared when the UARTx_LSR register is read. In FIFO mode, a line status interrupt is generated only after the received byte with an error reaches the top of the FIFO and is ready to be read.

A line status interrupt is activated (provided this interrupt is enabled) as long as the Read pointer of the receiver FIFO points to the location of the FIFO that contains a byte with the error. The interrupt is immediately cleared when the UARTx_LSR register is read. The ERR bit of the UARTx_LSR register is active as long as an erroneous byte is present in the receiver FIFO.

UART Modem Status Interrupt

The modem status interrupt is generated if there is any change in state of the modem status inputs to the UART. This interrupt is cleared when the processor reads the UARTx_MSR register.

UART Recommended Usage

The following is the standard sequence of events that occur in the eZ80L92 MCU using the UART. A description of each follows.

1. Module reset.
2. Control transfers to configure UART operation.
3. Data transfers.

Module Reset. Upon reset, all internal registers are set to their default values. All command status registers are programmed with their default values, and the FIFOs are flushed.

Control Transfers. Based on the requirements of the application, the data transfer baud rate is determined and the BRG is configured to generate a 16X clock frequency. Interrupts are disabled and the communication control parameters are programmed in the UARTx_LCTL register. The FIFO configuration is determined and the receive trigger levels are set in the UARTx_FCTL register. The status registers, UARTx_LSR and UARTx_MSR, are read, and ensure that none of the interrupt sources are active. The inter-

Infrared Encoder/Decoder

The eZ80L92 MCU contains a UART to infrared encoder/decoder (endec). The infrared encoder/decoder is integrated with the on-chip UART0 to allow easy communication between the eZ80 CPU and IrDA Physical Layer Specification Version 1.3 compliant infrared transceivers as illustrated in Figure 24. Infrared communication provides secure, reliable, high-speed, low-cost, and point-to-point communication between PCs, PDAs, mobile telephones, printers, and other infrared enabled devices.

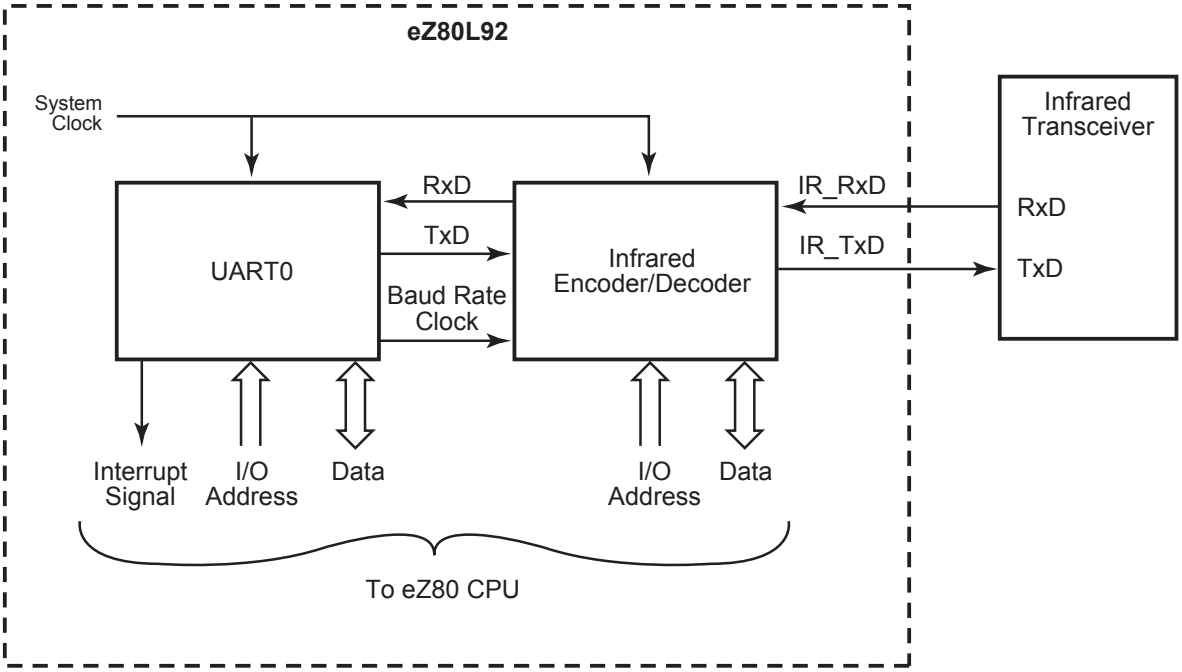


Figure 24. Infrared System Block Diagram

Functional Description

When the infrared encoder/decoder is enabled, the transmit data from the on-chip UART is encoded as digital signals in accordance with the IrDA standard and output to the infrared transceiver. Likewise, data received from the infrared transceiver is decoded by the infrared encoder/decoder and passed to the UART. Communication is half-duplex meaning that simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART Baud Rate Generator and supports IrDA standard baud rates from 9600 bps to 115.2 Kbps. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the infrared encoder/decoder.

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.

Serial Clock

The Serial Clock (SCK) is used to synchronize data movement both in and out of the device through its MOSI and MISO pins. The master and slave are capable of exchanging a data byte during a sequence of eight clock cycles. As SCK is generated by the master, the SCK pin becomes an input on a slave device. The SPI contains an internal divide-by-two clock divider. In MASTER mode, the SPI serial clock is one-half the frequency of the clock signal created by the SPI's Baud Rate Generator.

As demonstrated in Figure 29 and Table 68, four possible timing relations may be chosen by using control bits CPOL and CPHA in the SPI Control register. See the SPI Control Register (SPI_CTL) on page 135. Both the master and slave must operate with the identical timing, Clock Polarity (CPOL), and Clock Polarity (CPHA). The master device always places data on the MOSI line a half-cycle before the clock edge (SCK signal), in order for the slave device to latch the data.

Table 89 lists the recommended frequencies of the ZDI clock in relation to the system clock.

Table 89. Recommended ZDI Clock versus System Clock Frequency

System Clock Frequency	ZDI Clock Frequency
3–10 MHz	1 MHz
8–16 MHz	2 MHz
12–24 MHz	4 MHz
20–50 MHz	8 MHz

ZDI-Supported Protocol

ZDI supports a bidirectional serial protocol. The protocol defines any device that sends data as the *transmitter* and any receiving device as the *receiver*. The device controlling the transfer is the *master* and the device being controlled is the *slave*. The master always initiates the data transfers and provides the clock for both receive and transmit operations. The ZDI block on the eZ80L92 MCU is considered as slave in all data transfers.

Figure 36 illustrates the schematic for building a connector on a target board. This connector allows you to connect directly to the ZPAK emulator using a six-pin header.

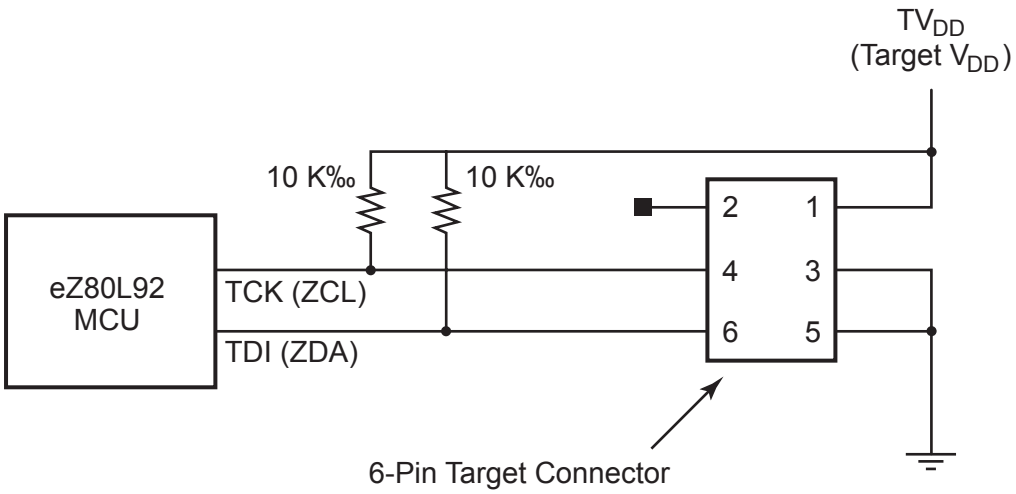


Figure 36. Schematic For Building a Target Board ZPAK Connector

can be repeated. This allows repeated Read or Write operations without having to resend the ZDI command. A START signal must follow to initiate a new ZDI command. Figure 39 illustrates the timing for address Writes to ZDI registers.

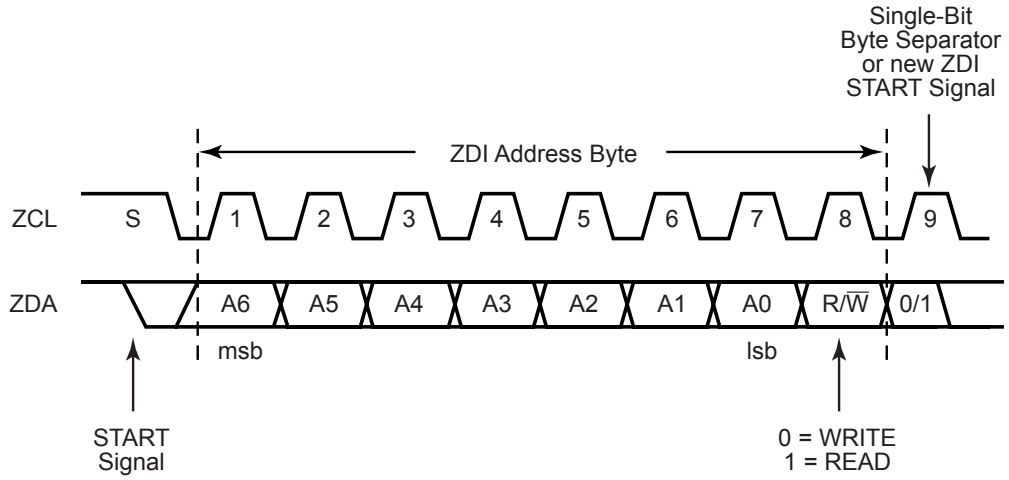


Figure 39. ZDI Address Write Timing

ZDI Write Operations

ZDI Single-Byte Write

For single-byte Write operations, the address and write control bit are first written to the ZDI block. Following the single-bit byte separator, the data is shifted into the ZDI block on the next 8 rising edges of ZCL. The master terminates activity after 8 clock cycles. Figure 40 illustrates the timing for ZDI single-byte Write operations.

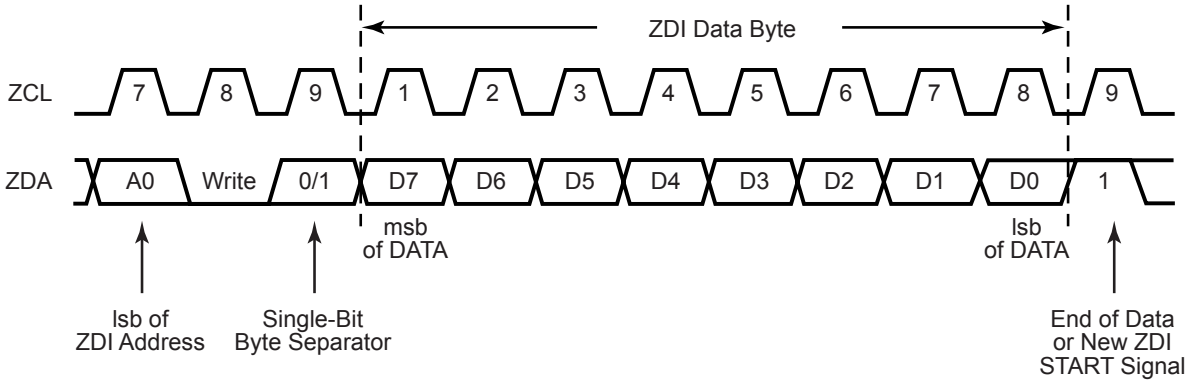


Figure 40. ZDI Single-Byte Data Write Timing

Bit Position	Value	Description
[7:0] zdi_wr_mem	00h–FFh	The 8-bit data that is transferred to the ZDI slave following a Write to this address is written to the address indicated by the current program counter. The program counter is incremented following each 8 bits of data. In Z80 MEMORY mode, ({MBASE, PC[15:0]}) ← 8 bits of transferred data. In ADL MEMORY mode, (PC[23:0]) ← 8 bits of transferred data.

eZ80® Product ID Low and High Byte Registers

The eZ80 Product ID Low and High Byte registers combine to provide a means for an external device to determine the particular eZ80 product being addressed. For the eZ80L92, these two bytes, {ZDI_ID_H, ZDI_ID_L} return the value {00h, 06h}. See Tables 100 and 101.

Table 100. eZ80® Product ID Low Byte Register (ZDI_ID_L = 00h in the ZDI Register Read-Only Address Space)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	1	1	0
CPU Access	R	R	R	R	R	R	R	R
Note: R = Read-only.								

Bit Position	Value	Description
[7:0] zdi_id_l	06h	{ZDI_ID_H, ZDI_ID_L} = {00h, 06h} indicates the eZ80L92 product.

Table 101. eZ80® Product ID High Byte Register (ZDI_ID_H = 01h in the ZDI Register Read-Only Address Space)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	R	R	R	R	R	R	R	R
Note: R = Read-only.								

On-Chip Instrumentation

On-Chip Instrumentation[1] (OCI™) for the ZiLOG eZ80 CPU core enables powerful debugging features. The OCI provides run control, memory and register visibility, complex breakpoints, and trace history features.

The OCI employs all the functions of the ZiLOG Debug Interface (ZDI) as described in the ZDI section. It also adds the following debug features:

- Control through a 4-pin JTAG port that conforms to IEEE Standard 1149.1 (Test Access Port and Boundary-Scan Architecture)[2].
- Complex break-point trigger functions.
- Break-point enhancements, such as the ability to:
  - Define two break-point addresses that form a range.
  - Break on masked data values.
  - Start or stop trace.
  - Assert a trigger output signal.
- Trace history buffer.
- Software break-point instruction.

OCI has the following sections:

1. JTAG interface.
2. ZDI debug control.
3. Trace buffer memory.
4. Complex triggers.

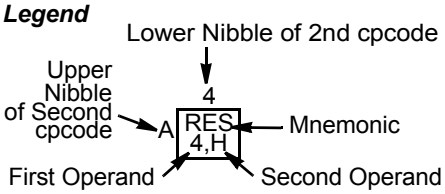
OCI Activation

The OCI features clock initialization circuitry so that external debug hardware can be detected during power up. The external debugger must drive the OCI clock pin (TCK) Low at least two system clock cycles prior to the end of the RESET to activate the OCI block. If TCK is High at the end of the RESET, the OCI block shuts down so that it does not draw power in normal product operation. When the OCI is shut down, ZDI is enabled directly and can be accessed through the clock (TCK) and data (TDI) pins. For more information on ZDI, see ZiLOG Debug Interface on page 160.

1. On-Chip Instrumentation and OCI are trademarks of First Silicon Solutions, Inc.
2. The eZ80L92 MCU does not contain the boundary scan register required for 1149.1 compliance.

Table 110. Opcode Map—Second Opcode after 0CBh

Legend



Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC L	RRC (HL)	RRC A
1	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR L	RR (HL)	RR A
2	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA L	SRA (HL)	SRA A
3									SRL B	SRL C	SRL D	SRL E	SRL H	SRL L	SRL (HL)	SRL A
4	BIT 0,B	BIT 0,C	BIT 0,D	BIT 0,E	BIT 0,H	BIT 0,L	BIT 0,(HL)	BIT 0,A	BIT 1,B	BIT 1,C	BIT 1,D	BIT 1,E	BIT 1,H	BIT 1,L	BIT 1,(HL)	BIT 1,A
5	BIT 2,B	BIT 2,C	BIT 2,D	BIT 2,E	BIT 2,H	BIT 2,L	BIT 2,(HL)	BIT 2,A	BIT 3,B	BIT 3,C	BIT 3,D	BIT 3,E	BIT 3,H	BIT 3,L	BIT 3,(HL)	BIT 3,A
6	BIT 4,B	BIT 4,C	BIT 4,D	BIT 4,E	BIT 4,H	BIT 4,L	BIT 4,(HL)	BIT 4,A	BIT 5,B	BIT 5,C	BIT 5,D	BIT 5,E	BIT 5,H	BIT 5,L	BIT 5,(HL)	BIT 5,A
7	BIT 6,B	BIT 6,C	BIT 6,D	BIT 6,E	BIT 6,H	BIT 6,L	BIT 6,(HL)	BIT 6,A	BIT 7,B	BIT 7,C	BIT 7,D	BIT 7,E	BIT 7,H	BIT 7,L	BIT 7,(HL)	BIT 7,A
8	RES 0,B	RES 0,C	RES 0,D	RES 0,E	RES 0,H	RES 0,L	RES 0,(HL)	RES 0,A	RES 1,B	RES 1,C	RES 1,D	RES 1,E	RES 1,H	RES 1,L	RES 1,(HL)	RES 1,A
9	RES 2,B	RES 2,C	RES 2,D	RES 2,E	RES 2,H	RES 2,L	RES 2,(HL)	RES 2,A	RES 3,B	RES 3,C	RES 3,D	RES 3,E	RES 3,H	RES 3,L	RES 3,(HL)	RES 3,A
A	RES 4,B	RES 4,C	RES 4,D	RES 4,E	RES 4,H	RES 4,L	RES 4,(HL)	RES 4,A	RES 5,B	RES 5,C	RES 5,D	RES 5,E	RES 5,H	RES 5,L	RES 5,(HL)	RES 5,A
B	RES 6,B	RES 6,C	RES 6,D	RES 6,E	RES 6,H	RES 6,L	RES 6,(HL)	RES 6,A	RES 7,B	RES 7,C	RES 7,D	RES 7,E	RES 7,H	RES 7,L	RES 7,(HL)	RES 7,A
C	SET 0,B	SET 0,C	SET 0,D	SET 0,E	SET 0,H	SET 0,L	SET 0,(HL)	SET 0,A	SET 1,B	SET 1,C	SET 1,D	SET 1,E	SET 1,H	SET 1,L	SET 1,(HL)	SET 1,A
D	SET 2,B	SET 2,C	SET 2,D	SET 2,E	SET 2,H	SET 2,L	SET 2,(HL)	SET 2,A	SET 3,B	SET 3,C	SET 3,D	SET 3,E	SET 3,H	SET 3,L	SET 3,(HL)	SET 3,A
E	SET 4,B	SET 4,C	SET 4,D	SET 4,E	SET 4,H	SET 4,L	SET 4,(HL)	SET 4,A	SET 5,B	SET 5,C	SET 5,D	SET 5,E	SET 5,H	SET 5,L	SET 5,(HL)	SET 5,A
F	SET 6,B	SET 6,C	SET 6,D	SET 6,E	SET 6,H	SET 6,L	SET 6,(HL)	SET 6,A	SET 7,B	SET 7,C	SET 7,D	SET 7,E	SET 7,H	SET 7,L	SET 7,(HL)	SET 7,A

Notes: n = 8-bit data; Mmn = 16- or 24-bit addr or data; d = 8-bit two's-complement displacement.

Table 123. External I/O Read Timing (Continued)

		20 MHz (ns)		50 MHz (ns)	
Parameter	**Description**	**Min**	**Max**	**Min**	**Max**
T_9	Clock Rise to $\overline{RD}$ Assertion Delay	2.7	7.0	2.7	7.0
T_{10}	Clock Rise to $\overline{RD}$ Deassertion Delay	2.4	6.3	2.4	6.3

External I/O Write Timing

Figure 51 and Table 124 diagram the timing for external I/O Writes.

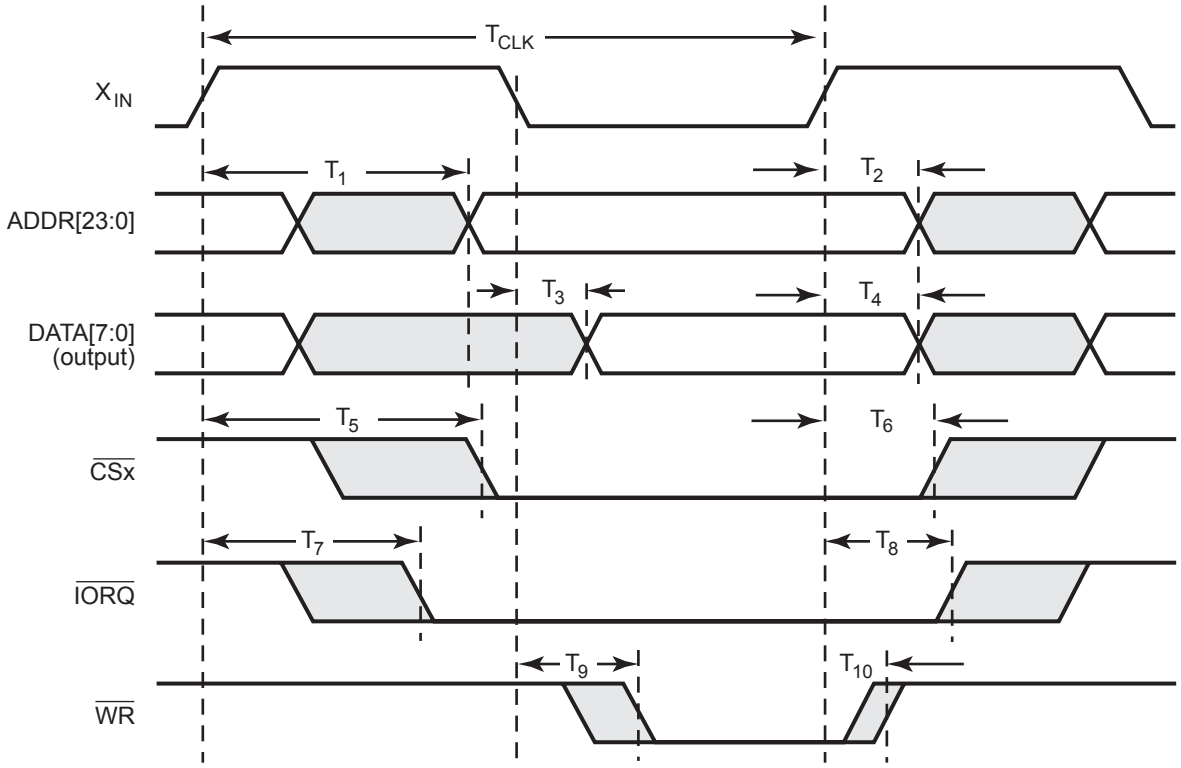


Figure 51. External I/O Write Timing

Table 124. External I/O Write Timing

		20 MHz (ns)		50 MHz (ns)	
Parameter	**Description**	**Min**	**Max**	**Min**	**Max.**
T_1	Clock Rise to ADDR Valid Delay	—	7.7	—	7.7
T_2	Clock Rise to ADDR Hold Time	2.2	—	2.2	—
T_3	Clock Fall to Output DATA Valid Delay	—	6	—	6
T_4	Clock Rise to DATA Hold Time	2.3	—	2.3	—
T_5	Clock Rise to $\overline{CSx}$ Assertion Delay	2.6	10.8	2.6	10.8
T_6	Clock Rise to $\overline{CSx}$ Deassertion Delay	2.4	8.8	2.4	8.8
T_7	Clock Rise to $\overline{IORQ}$ Assertion Delay	2.6	7.0	2.6	7.0
T_8	Clock Rise to $\overline{IORQ}$ Deassertion Delay	2.3	6.3	2.3	6.3
T_9	Clock Fall to $\overline{WR}$ Assertion Delay	1.8	4.5	1.8	4.5
T_{10}	Clock Rise to $\overline{WR}$ Deassertion Delay*	1.6	4.4	1.6	4.4
	$\overline{WR}$ Deassertion to ADDR Hold Time	0.4	—	0.4	—
	$\overline{WR}$ Deassertion to DATA Hold Time	0.5	—	0.5	—
	$\overline{WR}$ Deassertion to $\overline{CSx}$ Hold Time	1.2	—	1.2	—
	$\overline{WR}$ Deassertion to $\overline{IORQ}$ Hold Time	0.5	—	0.5	—

Note: *At the conclusion of a Write cycle, deassertion of $\overline{WR}$ always occurs before any change to ADDR, DATA, $\overline{CSx}$, or $\overline{IORQ}$. In certain applications, the deassertion of $\overline{WR}$ can be concurrent with ADDR, DATA, $\overline{CSx}$, or $\overline{MREQ}$ when buffering is used off-chip.

Table 125. GPIO Port Output Timing

		20 MHz (ns)		50 MHz (ns)	
Parameter	**Description**	**Min**	**Max**	**Min**	**Max**
T_1	Clock Rise to Port Output Valid Delay	—	9.3	—	9.3
T_2	Clock Rise to Port Output Hold Time	2.0	—	2.0	—

External Bus Acknowledge Timing

Table 126 provides information about the external bus acknowledge timing.

Table 126. Bus Acknowledge Timing

		20 MHz (ns)		50 MHz (ns)	
Parameter	**Description**	**Min**	**Max**	**Min**	**Max**
T_1	Clock Rise to $\overline{\text{BUSACK}}$ Assertion Delay	2.8	9.3	2.8	9.3
T_2	Clock Rise to $\overline{\text{BUSACK}}$ Deassertion Delay	2.5	6.5	2.5	6.5

External System Clock Driver (PHI) Timing

Table 127 lists timing information for the PHI pin. The PHI pin allows external peripherals to synchronize with the internal system clock driver on the eZ80L92 MCU.

Table 127. PHI System Clock Timing

		20 MHz (ns)		50 MHz (ns)	
Parameter	**Description**	**Min**	**Max**	**Min**	**Max**
T_1	Clock Rise to PHI Rise	1.6	4.6	1.6	4.6
T_2	Clock Fall to PHI Fall	1.8	4.3	1.8	4.3