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	50MHz
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/ez80l92az050sc

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

Pin No	Symbol	Function	Signal Direction	Description
7	V_{DD}	Power Supply		Power Supply.
8	V_{SS}	Ground		Ground.
9	ADDR6	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
10	ADDR7	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
11	ADDR8	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
12	ADDR9	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.
13	ADDR10	Address Bus	Bidirectional	Configured as an output in normal operation. The address bus selects a location in memory or I/O space to be read or written. Configured as an input during bus acknowledge cycles. Drives the Chip Select/Wait State Generator block to generate Chip Selects.

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

Pin No	Symbol	Function	Signal Direction	Description
84	V_{SS}	Ground		Ground.
85	X_{IN}	System Clock Oscillator Input	Input	This pin is the input to the onboard crystal oscillator for the primary system clock. If an external oscillator is used, its clock output should be connected to this pin. When a crystal is used, it should be connected between X_{IN} and X_{OUT}.
86	X_{OUT}	System Clock Oscillator Output	Output	This pin is the output of the onboard crystal oscillator. When used, a crystal should be connected between X_{IN} and X_{OUT}.
87	V_{DD}	Power Supply		Power Supply.
88	PB0	GPIO Port B	Bidirectional	This pin can be used for GPIO. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	T0_IN	Timer 0 In	Input	Alternate clock source for Programmable Reload Timers 0 and 2. This signal is multiplexed with PB0.
89	PB1	GPIO Port B	Bidirectional	This pin can be used for GPIO. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	T1_IN	Timer 1 In	Input	Alternate clock source for Programmable Reload Timers 1 and 3. This signal is multiplexed with PB1.

Pin Characteristics

Table 2 describes the characteristics of each pin in the eZ80L92 MCU's 100-pin LQFP package.

Table 2. Pin Characteristics of eZ80L92 MCU

Pin No.	Symbol	Direction	Reset Direction	Active Low/High	Tristate Output	Pull Up/Down	Schmitt Trigger Input	Open Drain/Source
1	ADDR0	I/O	O	N/A	Yes	No	No	No
2	ADDR1	I/O	O	N/A	Yes	No	No	No
3	ADDR2	I/O	O	N/A	Yes	No	No	No
4	ADDR3	I/O	O	N/A	Yes	No	No	No
5	ADDR4	I/O	O	N/A	Yes	No	No	No
6	ADDR5	I/O	O	N/A	Yes	No	No	No
7	V_{DD}							
8	V_{SS}							
9	ADDR6	I/O	O	N/A	Yes	No	No	No
10	ADDR7	I/O	O	N/A	Yes	No	No	No
11	ADDR8	I/O	O	N/A	Yes	No	No	No
12	ADDR9	I/O	O	N/A	Yes	No	No	No
13	ADDR10	I/O	O	N/A	Yes	No	No	No
14	ADDR11	I/O	O	N/A	Yes	No	No	No
15	ADDR12	I/O	O	N/A	Yes	No	No	No
16	ADDR13	I/O	O	N/A	Yes	No	No	No
17	ADDR14	I/O	O	N/A	Yes	No	No	No
18	V_{DD}							
19	V_{SS}							
20	ADDR15	I/O	O	N/A	Yes	No	No	No
21	ADDR16	I/O	O	N/A	Yes	No	No	No
22	ADDR17	I/O	O	N/A	Yes	No	No	No
23	ADDR18	I/O	O	N/A	Yes	No	No	No
24	ADDR19	I/O	O	N/A	Yes	No	No	No

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.

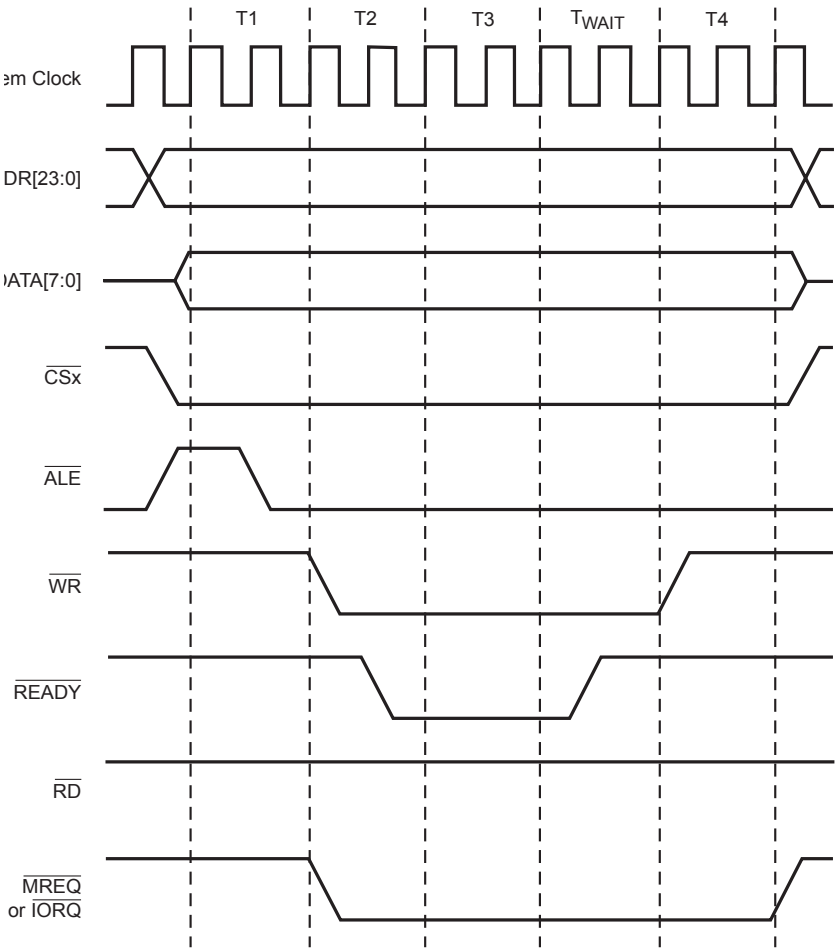


Figure 11. Intel® Bus Mode Write Timing Example (Separate Address and Data Buses)

Signal timing for Intel® Bus Mode with multiplexed address and data is illustrated for a Read operation in Figure 12 and for a Write operation in Figure 13. In the figures, each Intel® Bus Mode state is 2 eZ80 system clock cycles in duration. Figure 12 and Figure 13 also illustrate the assertion of one WAIT state (T_{WAIT}) by the selected peripheral.

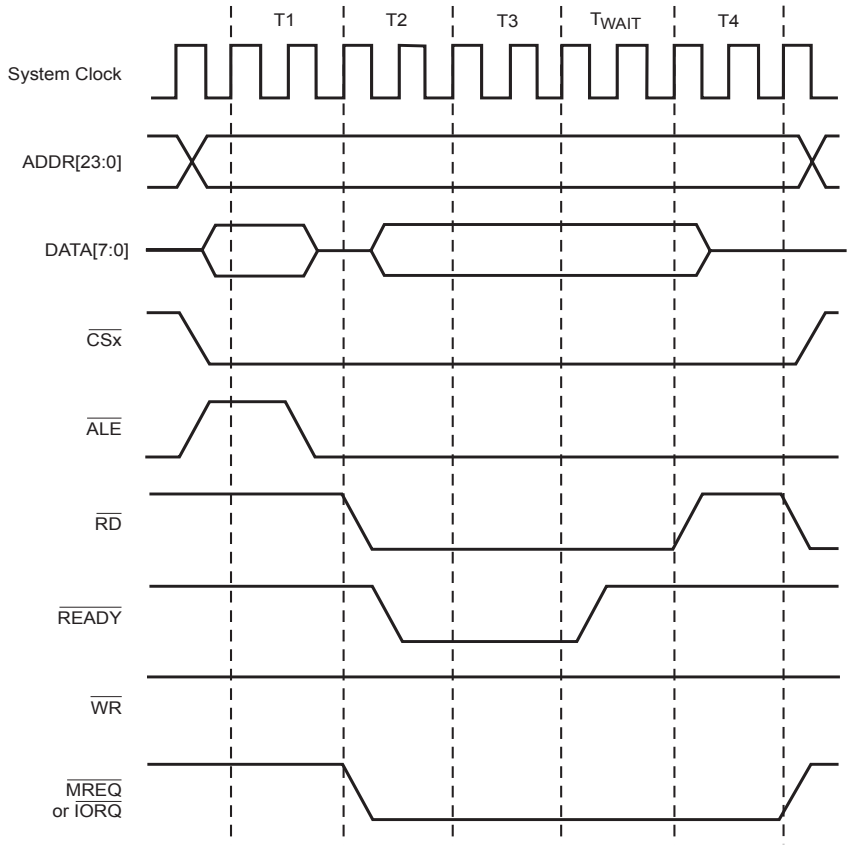


Figure 12. Intel® Bus Mode Read Timing Example (Multiplexed Address and Data Bus)

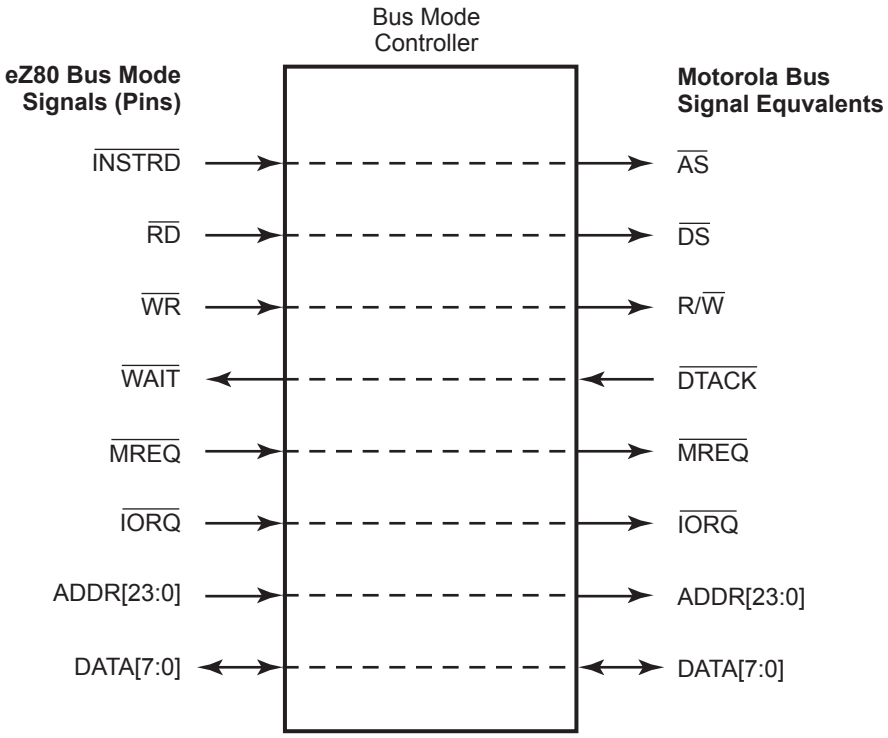


Figure 14. Motorola Bus Mode Signal and Pin Mapping

During Write operations, the Motorola Bus Mode employs 8 states (S0, S1, S2, S3, S4, S5, S6, and S7) as described in Table 20.

Table 20. Motorola Bus Mode Read States

STATE S0	The Read cycle starts in state S0. The CPU drives R/$\overline{W}$ High to identify a Read cycle.
STATE S1	Entering state S1, the CPU drives a valid address on the address bus, ADDR[23:0].
STATE S2	On the rising edge of state S2, the CPU asserts $\overline{AS}$ and $\overline{DS}$.
STATE S3	During state S3, no bus signals are altered.
STATE S4	During state S4, the CPU waits for a cycle termination signal $\overline{DTACK}$ ($\overline{WAIT}$), a peripheral signal. If the termination signal is not asserted at least one full CPU clock period prior to the rising clock edge at the end of S4, the CPU inserts WAIT (T_{WAIT}) states until $\overline{DTACK}$ is asserted. Each WAIT state is a full bus mode cycle.
STATE S5	During state S5, no bus signals are altered.

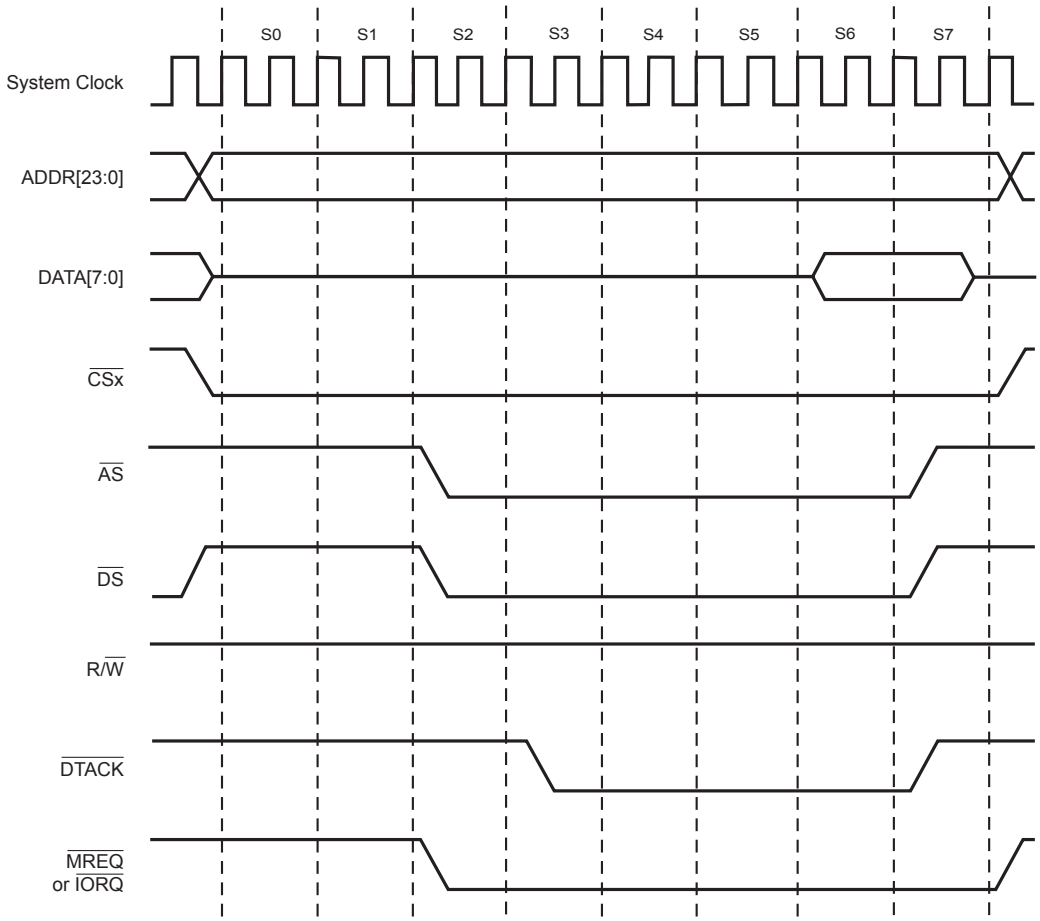


Figure 15. Motorola Bus Mode Read Timing Example

Programmable Reload Timers

The eZ80L92 MCU features six Programmable Reload Timers (PRT). Each PRT contains a 16-bit downcounter and a 16-bit reload register. In addition, each PRT features a clock prescaler with four selectable taps for CLK ÷ 4, CLK ÷ 16, CLK ÷ 64, and CLK ÷ 256. Each timer can be individually enabled to operate in either SINGLE PASS or CONTINUOUS mode. The timer can be programmed to start, stop, restart from the current value, or restart from the initial value, and generate interrupts to the CPU.

Four of the Programmable Reload Timers (timers 0–3) feature a selectable clock source input. The input for these timers can be either the system clock or the Real-Time Clock (RTC) source. Timers 0–3 can also be used for event counting, with their inputs received from a GPIO port pin. Output from timers 4 and 5 can be directed to a GPIO port pin.

Each of the six PRTs available on the eZ80L92 can be controlled individually. They do not share the same counters, reload registers, control registers, or interrupt signals.
A simplified block diagram of a programmable reload timer is illustrated in Figure 18.

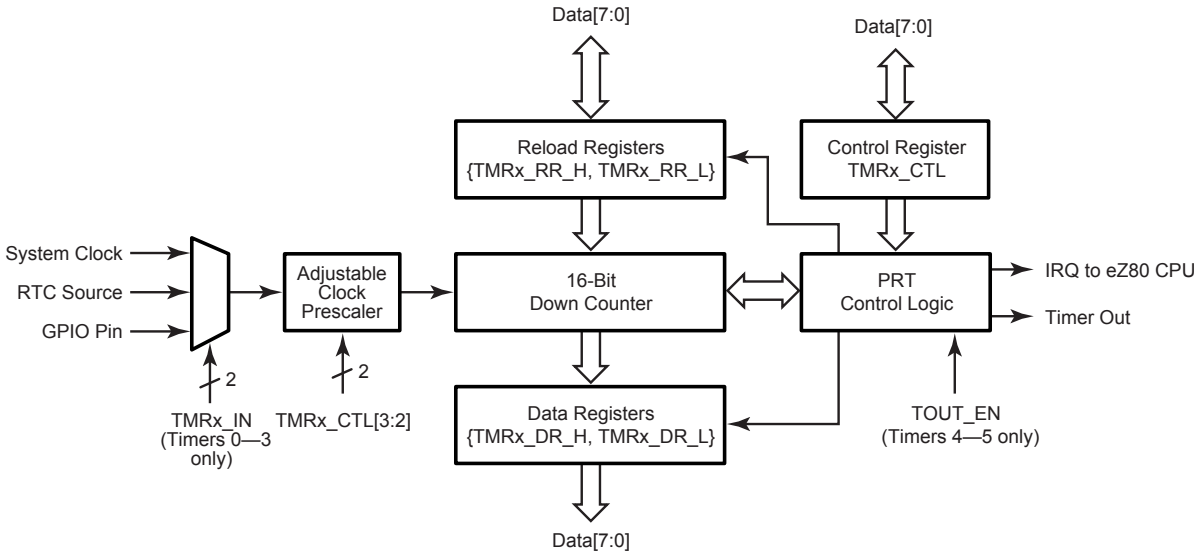


Figure 18. Programmable Reload Timer Block Diagram

Programmable Reload Timer Operation

Setting Timer Duration

There are three factors to consider when determining Programmable Reload Timer duration—clock frequency, clock divider ratio, and initial count value. Minimum duration

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.

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.

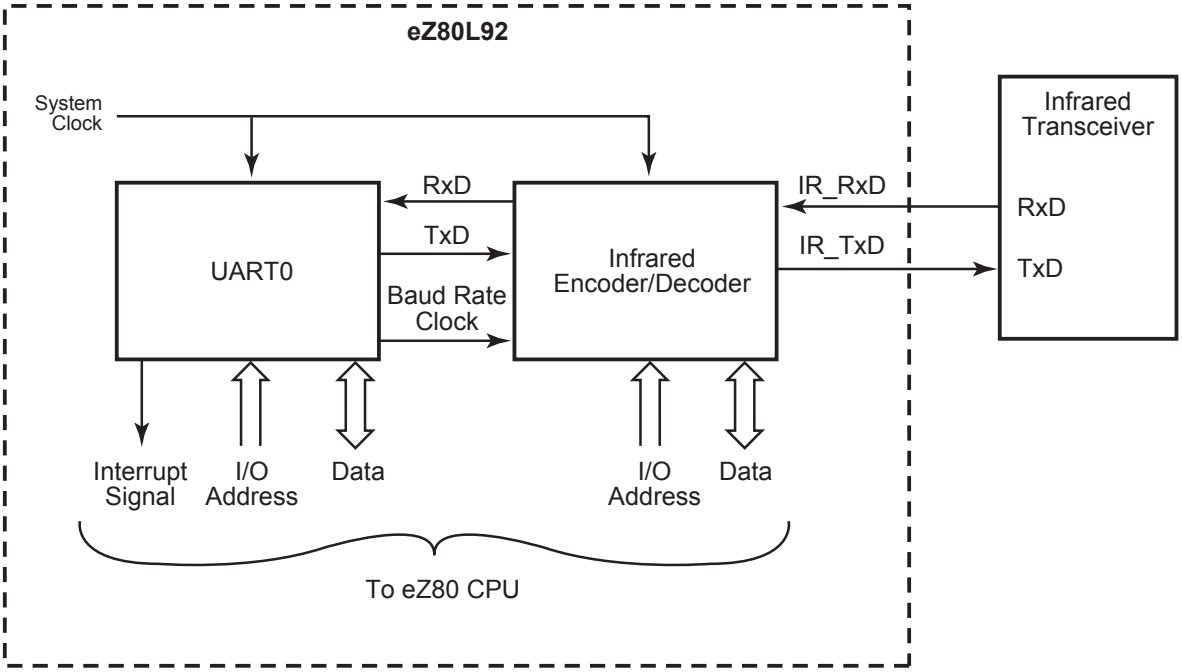


Figure 24. Infrared System Block Diagram

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.

Bit Position	Value	Description
7 IEN	0	I^2C interrupt is disabled.
	1	I^2C interrupt is enabled.
6 ENAB	0	The I^2C bus (SCL/SDA) is disabled and all inputs are ignored.
	1	The I^2C bus (SCL/SDA) is enabled.
5 STA	0	Master mode START condition is sent.
	1	Master mode start-transmit START condition on the bus.
4 STP	0	Master mode STOP condition is sent.
	1	Master mode stop-transmit STOP condition on the bus.
3 IFLG	0	I^2C interrupt flag is not set.
	1	I^2C interrupt flag is set.
2 AAK	0	Not Acknowledge.
	1	Acknowledge.
[1:0]	00	Reserved.

I^2C Status Register

The I2C_SR register is a Read-Only register that contains a 5-bit status code in the five most significant bits: the three least significant bits are always 0. The Read-Only I2C_SR registers share the same I/O addresses as the Write-Only I2C_CCR registers. See Table 85.

Table 85. I^2C Status Registers (I2C_SR = 00CCh)

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

Bit Position	Value	Description
[7:3] STAT	00000–11111	5-bit I^2C status code.
[2:0]	000	Reserved.

Table 86. I²C Status Codes (Continued)

Code	Status
B0h	Arbitration lost in address as master, slave address and Read bit received, ACK transmitted
B8h	Data byte transmitted in SLAVE mode, ACK received
C0h	Data byte transmitted in SLAVE mode, ACK not received
C8h	Last byte transmitted in SLAVE mode, ACK received
D0h	Second Address byte and Write bit transmitted, ACK received
D8h	Second Address byte and Write bit transmitted, ACK not received
F8h	No relevant status information, IFLG = 0

If an illegal condition occurs on the I²C bus, the bus error state is entered (status code 00h). To recover from this state, the STP bit in the I2C_CTL register must be set and the IFLG bit cleared. The I²C then returns to the idle state. No STOP condition is transmitted on the I²C bus.

Note: The STP and STA bits may be set to 1 at the same time to recover from the bus error. The I²C then sends a START.

I²C Clock Control Register

The I2C_CCR register is a Write-Only register. The seven LSBs control the frequency at which the I²C bus is sampled and the frequency of the I²C clock line (SCL) when the I²C is in MASTER mode. The Write-Only I2C_CCR registers share the same I/O addresses as the Read-Only I2C_SR registers. See Table 87.

Table 87. I²C Clock Control Registers (I2C_CCR = 00CCh)

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 = Read only.								

Bit Position	Value	Description
7	0	Reserved.

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.

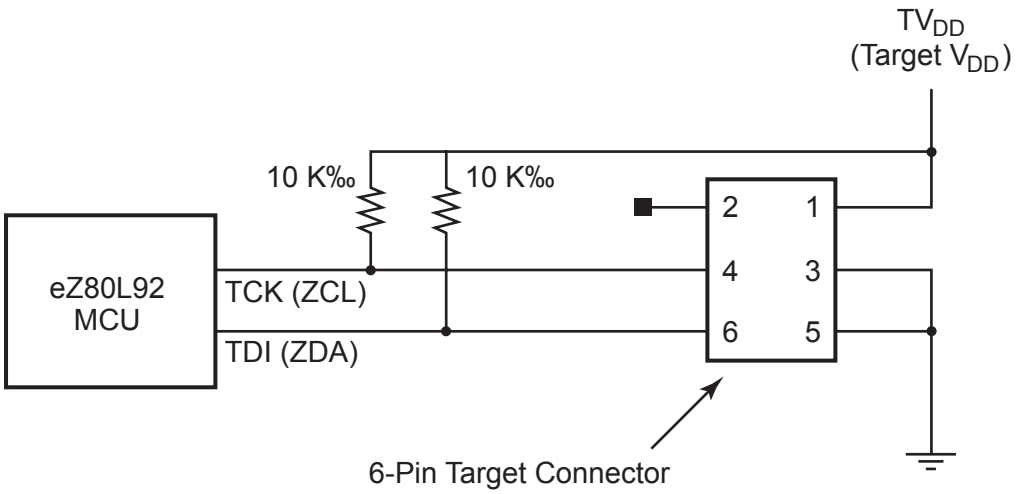


Figure 36. Schematic For Building a Target Board ZPAK Connector

ZDI Block Write

The Block Write operation is initiated in the same manner as the single-byte Write operation, but instead of terminating the Write operation after the first data byte is transferred, the ZDI master can continue to transmit additional bytes of data to the ZDI slave on the eZ80L92. After the receipt of each byte of data the ZDI register address increments by 1. If the ZDI register address reaches the end of the Write-Only ZDI register address space (30h), the address stops incrementing. Figure 41 illustrates the timing for ZDI Block Write operations.

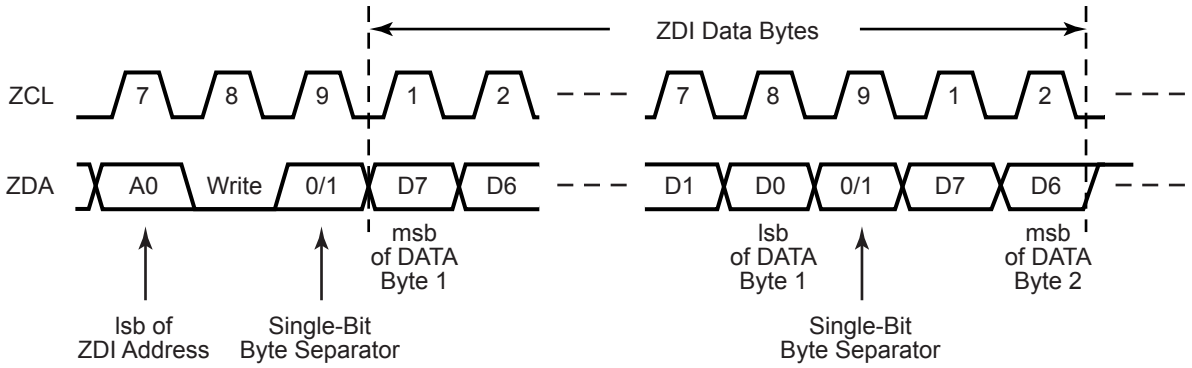


Figure 41. ZDI Block Data Write Timing

ZDI Read Operations

ZDI Single-Byte Read

Single-byte Read operations are initiated in the same manner as single-byte Write operations, with the exception that the R/$\overline{W}$ bit of the ZDI register address is set to 1. Upon receipt of a slave address with the R/$\overline{W}$ bit set to 1, the eZ80L92's ZDI block loads the selected data into the shifter at the beginning of the first cycle following the single-bit data separator. The most significant bit (msb) is shifted out first. Figure 42 illustrates the timing for ZDI single-byte Read operations.

When using the ZDI interface, any Write or Read operations of peripheral registers in the I/O address space produces the same effect as Read or Write operations using the CPU. Because many register Read/Write operations exhibit secondary effects, such as clearing flags or causing operations to commence, the effects of the Read/Write operations during a ZDI Break must be taken into consideration.

Bus Requests During ZDI Debug Mode

The ZDI block on the eZ80L92 allows an external device to take control of the address and data bus while the eZ80L92 is in DEBUG mode. ZDI_BUSACK_EN causes ZDI to allow or prevent acknowledgement of bus requests by external peripherals. The bus acknowledge only occurs at the end of the current ZDI operation (indicated by a High during the single-bit byte separator). The default reset condition is for bus acknowledgement to be disabled. To allow bus acknowledgement, the ZDI_BUSACK_EN must be written.

When an external bus request ($\overline{\text{BUSREQ}}$ pin asserted) is detected, ZDI waits until completion of the current operation before responding. ZDI acknowledges the bus request by asserting the bus acknowledge ($\overline{\text{BUSACK}}$) signal. If the ZDI block is not currently shifting data, it acknowledges the bus request immediately. ZDI uses the single-bit byte separator of each data word to determine if it is at the end of a ZDI operation. If the bit is a logical 0, ZDI does not assert $\overline{\text{BUSACK}}$ to allow additional data Read or Write operations. If the bit is a logical 1, indicating completion of the ZDI commands, $\overline{\text{BUSACK}}$ is asserted.

Potential Hazards of Enabling Bus Requests During Debug Mode

There are some potential hazards that the user must be aware of when enabling external bus requests during ZDI Debug mode. First, when the address and data bus are being used by an external source, ZDI must only access ZDI registers and internal CPU registers to prevent possible Bus contention. The bus acknowledge status is reported in the ZDI_BUS_STAT register. The $\overline{\text{BUSACK}}$ output pin also indicates the bus acknowledge state.

A second hazard is that when a bus acknowledge is granted, the ZDI is subject to any WAIT states that are assigned to the device currently being accessed by the external peripheral. To prevent data errors, ZDI should avoid data transmission while another device is controlling the bus.

Finally, exiting ZDI Debug mode while an external peripheral controls the address and data buses, as indicated by $\overline{\text{BUSACK}}$ assertion, may produce unpredictable results.

Table 98. Instruction Store 4:0 Registers (ZDI_IS4 = 21h, ZDI_IS3 = 22h, ZDI_IS2 = 23h, ZDI_IS1 = 24h, and ZDI_IS0 = 25h in the ZDI Register Write-Only Address Space)

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

Bit Position	Value	Description
[7:0] ZDI_IS4, ZDI_IS3, ZDI_IS2, ZDI_IS1, or ZDI_IS0	00h–FFh	These registers contain the Op Codes and operands for immediate execution by the CPU following a Write to ZDI_IS0. The ZDI_IS0 register contains the first Op Code of the instruction. The remaining ZDI_ISx registers contain any additional Op Codes or operand dates required for execution of the required instruction.

ZDI Write Memory Register

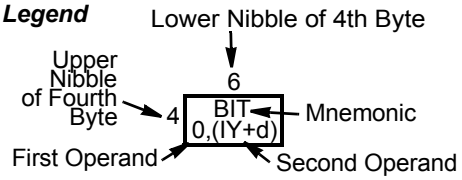
A Write to the ZDI Write Memory register causes the eZ80L92 to write the 8-bit data to the memory location specified by the current address in the program counter. In Z80 MEMORY mode, this address is {MBASE, PC[15:0]}. In ADL MEMORY mode, this address is PC[23:0]. The program counter, PC, increments after each data Write. However, the ZDI register address does not increment automatically when this register is accessed. As a result, the ZDI master is allowed to write any number of data bytes by writing to this address one time followed by any number of data bytes. See Table 99.

Table 99. ZDI Write Memory Register (ZDI_WR_MEM = 30h in the ZDI Register Write-Only Address Space)

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

Table 115. Opcode Map—Fourth Byte After 0FDh, 0CBh, and dd*

Legend



Lower Nibble (Hex)

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

Notes: d = 8-bit two's-complement displacement.

On-Chip Oscillators

The eZ80L92 MCU features two on-chip oscillators for use with an external crystal. The primary oscillator generates the system clock for the internal CPU and for the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin can accept a CMOS-level clock input signal. If an external clock generator is used, the X_{OUT} pin must be left disconnected. The secondary oscillator can drive a 32 kHz crystal to generate the time-base for the real time clock.

20 MHz Primary Crystal Oscillator Operation

Figure 44 illustrates a recommended configuration for connection with an external 20 MHz, fundamental-mode, and parallel-resonant crystal. Table 116 lists recommended crystal specifications. Resistor R_1 limits total power dissipation by the crystal. Printed circuit board layout must add not more than 4 pF of stray capacitance to either the X_{IN} or X_{OUT} pins. If oscillation does not occur, reduce the values of capacitors C_1 and C_2 to decrease loading.

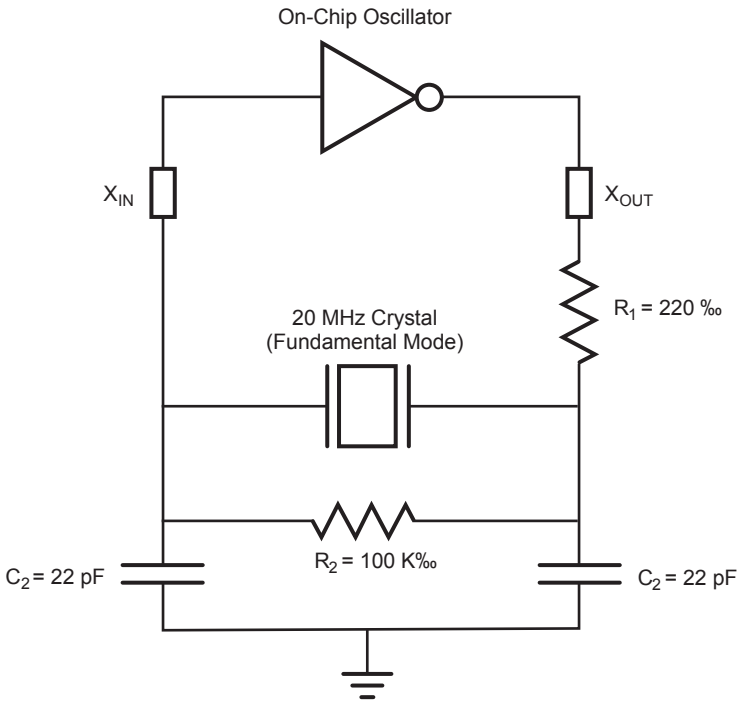


Figure 44. Recommended Crystal Oscillator Configuration (20 MHz operation)

General Purpose I/O Port Input Sample Timing

Figure 54 illustrates timing of the GPIO input sampling. The input value on a GPIO port pin is sampled on the rising edge of the system clock. The port value is then available to the CPU on the second rising clock edge following the change of the port value.

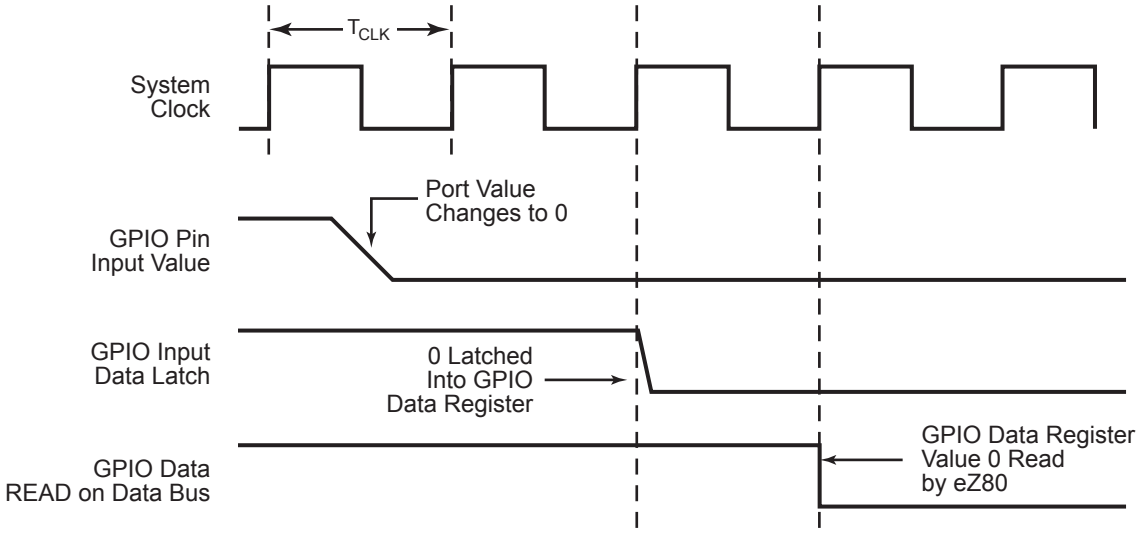


Figure 54. Port Input Sample Timing

General Purpose I/O Port Output Timing

Figure 55 and Table 125 provide timing information for GPIO port pins.

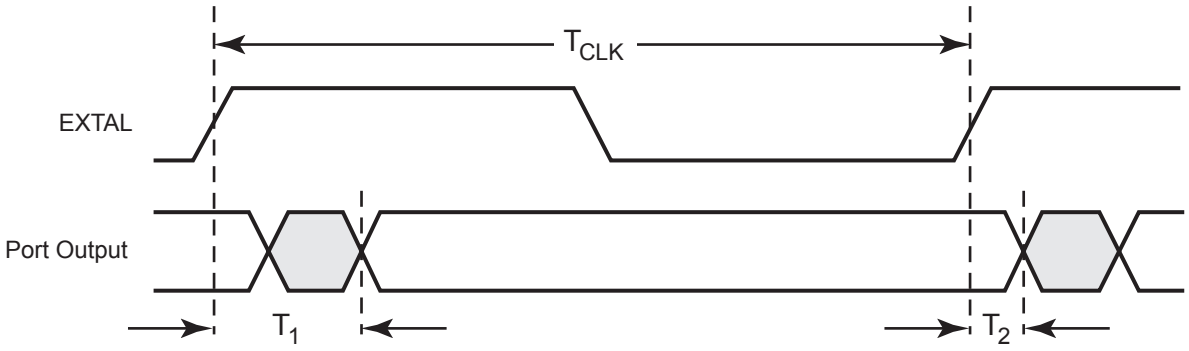


Figure 55. GPIO Port Output Timing