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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80l92az050ec00tr

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

Pin No	Symbol	Function	Signal Direction	Description
52	NMI	Nonmaskable Interrupt	Schmitt Trigger Input, Active Low	The $\overline{\text{NMI}}$ input is a higher priority input than the maskable interrupts. It is always recognized at the end of an instruction, regardless of the state of the interrupt enable control bits. This input includes a Schmitt trigger to allow RC rise times.
53	BUSREQ	Bus Request	Input, Active Low	External devices can request the eZ80L92 MCU to release the memory interface bus for their use, by driving this pin Low.
54	BUSACK	Bus Acknowledge	Output, Active Low	The eZ80L92 MCU responds to a Low on $\overline{\text{BUSREQ}}$, by tristating the address, data, and control signals, and by driving the $\overline{\text{BUSACK}}$ line Low. During bus acknowledge cycles ADDR[23:0], $\overline{\text{IORQ}}$, and $\overline{\text{MREQ}}$ are inputs.
55	HALT_SLP	HALT and SLEEP Indicator	Output, Active Low	A Low on this pin indicates that the CPU has entered either HALT or SLEEP mode because of execution of either a HALT or SLP instruction.
56	V_{DD}	Power Supply		Power Supply.
57	V_{SS}	Ground		Ground.
58	RTC_XIN	Real-Time Clock Crystal Input	Input	This pin is the input to the low-power 32 kHz crystal oscillator for the Real-Time Clock.
59	RTC_XOUT	Real-Time Clock Crystal Output	Bidirectional	This pin is the output from the low-power 32 kHz crystal oscillator for the Real-Time Clock. This pin is an input when the RTC is configured to operate from 50/60 Hz input clock signals and the 32 kHz crystal oscillator is disabled.
60	RTC_V_{DD}	Real-Time Clock Power Supply		Power supply for the Real-Time Clock and associated 32 kHz oscillator. Isolated from the power supply to the remainder of the chip. A battery can be connected to this pin to supply constant power to the Real-Time Clock and 32 kHz oscillator.
61	V_{SS}	Ground		Ground.
62	TMS	JTAG Test Mode Select	Input	JTAG Mode Select Input.

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

Pin No	Symbol	Function	Signal Direction	Description
70	PD2	GPIO Port D	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 D pin, when programmed as output, can be selected to be an open-drain or open-source output. Port D is multiplexed with one UART.
	RTS0	Request to Send	Output, Active Low	Modem control signal from UART. This signal is multiplexed with PD2.
71	PD3	GPIO Port D	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 D pin, when programmed as output, can be selected to be an open-drain or open-source output. Port D is multiplexed with one UART.
	CTS0	Clear to Send	Input, Active Low	Modem status signal to the UART. This signal is multiplexed with PD3.
72	PD4	GPIO Port D	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 D pin, when programmed as output, can be selected to be an open-drain or open-source output. Port D is multiplexed with one UART.
	DTR0	Data Terminal Ready	Output, Active Low	Modem control signal to the UART. This signal is multiplexed with PD4.
73	PD5	GPIO Port D	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 D pin, when programmed as output, can be selected to be an open-drain or open-source output. Port D is multiplexed with one UART.
	DSR0	Data Set Ready	Input, Active Low	Modem status signal to the UART. This signal is multiplexed with PD5.

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

Pin No	Symbol	Function	Signal Direction	Description
90	PB2	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.
	SS	Slave Select	Input, Active Low	The slave select input line is used to select a slave device in SPI mode. This signal is multiplexed with PB2.
91	PB3	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.
	SCK	SPI Serial Clock	Bidirectional	SPI serial clock. This signal is multiplexed with PB3.
92	PB4	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.
	T4_OUT	Timer 4 Out	Output	Programmable Reload Timer 4 timer-out signal. This signal is multiplexed with PB4.
93	PB5	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.
	T5_OUT	Timer 5 Out	Output	Programmable Reload Timer 5 timer-out signal. This signal is multiplexed with PB5.

Table 2. Pin Characteristics of eZ80L92 MCU (Continued)

Pin No.	Symbol	Direction	Reset Direction	Active Low/High	Tristate Output	Pull Up/Down	Schmitt Trigger Input	Open Drain/Source
25	ADDR20	I/O	O	N/A	Yes	No	No	No
26	ADDR21	I/O	O	N/A	Yes	No	No	No
27	ADDR22	I/O	O	N/A	Yes	No	No	No
28	ADDR23	I/O	O	N/A	Yes	No	No	No
29	CS0	O	O	Low	No	No	No	No
30	CS1	O	O	Low	No	No	No	No
31	CS2	O	O	Low	No	No	No	No
32	CS3	O	O	Low	No	No	No	No
33	V_{DD}							
34	V_{SS}							
35	DATA0	I/O	I	N/A	Yes	No	No	No
36	DATA1	I/O	I	N/A	Yes	No	No	No
37	DATA2	I/O	I	N/A	Yes	No	No	No
38	DATA3	I/O	I	N/A	Yes	No	No	No
39	DATA4	I/O	I	N/A	Yes	No	No	No
40	DATA5	I/O	I	N/A	Yes	No	No	No
41	DATA6	I/O	I	N/A	Yes	No	No	No
42	DATA7	I/O	I	N/A	Yes	No	No	No
43	V_{DD}							
44	V_{SS}							
45	IORQ	I/O	O	Low	Yes	No	No	No
46	MREQ	I/O	O	Low	Yes	No	No	No
47	RD	O	O	Low	No	No	No	No
48	WR	O	O	Low	No	No	No	No
49	INSTRD	O	O	Low	No	No	No	No
50	WAIT	I	I	Low	N/A	No	No	N/A
51	RESET	I	I	Low	N/A	Up	Yes	N/A
52	NMI	I	I	Low	N/A	No	Yes	N/A

{MBASE[7:0], I[7:0], 1Eh} and {MBASE, I[7:0], 1Fh}. The least significant byte is stored at the lower address.

When any one or more of the interrupt requests (IRQs) become active, an interrupt request is generated by the interrupt controller and sent to the CPU. The corresponding 8-bit interrupt vector for the highest priority interrupt is placed on the 8-bit interrupt vector bus, IVECT[7:0]. The interrupt vector bus is internal to the eZ80L92 and is therefore not visible externally. The response time of the eZ80 CPU to an interrupt request is a function of the current instruction being executed as well as the number of WAIT states being asserted.

The interrupt vector, {I[7:0], IVECT[7:0]}, is visible on the address bus, ADDR[15:0], when the interrupt service routine begins. The response of the eZ80 CPU to a vectored interrupt on the eZ80L92 is explained in Table 12. Interrupt sources are required to be active until the Interrupt Service Routine (ISR) starts. We recommend you to change the Interrupt Page Address Register (I) value from its default value of 00h as this address can create conflicts between the non-maskable interrupt vector, the RST instruction addresses, and the maskable interrupt vectors.

Table 12. Vectored Interrupt Operation

Memory Mode	ADL Bit	MADL Bit	Operation
Z80 Mode	0	0	Read the LSB of the interrupt vector placed on the internal vectored interrupt bus, IVECT [7:0], by the interrupting peripheral. • IEF1 ← 0 • IEF2 ← 0 • The starting Program Counter is effectively {MBASE, PC[15:0]}. • Push the 2-byte return address PC[15:0] on the ({MBASE,SPS}) stack. • The ADL mode bit remains cleared to 0. • The interrupt vector address is located at { MBASE, I[7:0], IVECT[7:0] }. • PC[15:0] ← ({ MBASE, I[7:0], IVECT[7:0] }). • The ending Program Counter is effectively {MBASE, PC[15:0]} • The interrupt service routine must end with RETI.

Table 12. Vectored Interrupt Operation (Continued)

Memory Mode	ADL Bit	MADL Bit	Operation
ADL Mode	1	0	Read the LSB of the interrupt vector placed on the internal vectored interrupt bus, IVECT [7:0], by the interrupting peripheral. • IEF1 ← 0 • IEF2 ← 0 • The starting Program Counter is PC[23:0]. • Push the 3-byte return address, PC[23:0], onto the SPL stack. • The ADL mode bit remains set to 1. • The interrupt vector address is located at { 00h, I[7:0], IVECT[7:0] }. • PC[15:0] ← ({ 00h, I[7:0], IVECT[7:0] }). • The ending Program Counter is { 00h, PC[15:0] }. • The interrupt service routine must end with RETI.
Z80 Mode	0	1	Read the LSB of the interrupt vector placed on the internal vectored interrupt bus, IVECT[7:0], bus by the interrupting peripheral. • IEF1 ← 0 • IEF2 ← 0 • The starting Program Counter is effectively {MBASE, PC[15:0]}. • Push the 2-byte return address, PC[15:0], onto the SPL stack. • Push a 00h byte onto the SPL stack to indicate an interrupt from Z80® mode (because ADL = 0). • Set the ADL mode bit to 1. • The interrupt vector address is located at { 00h, I[7:0], IVECT[7:0] }. • PC[15:0] ← ({ 00h, I[7:0], IVECT[7:0] }). • The ending Program Counter is { 00h, PC[15:0] }. • The interrupt service routine must end with RETI.L

Watchdog Timer Operation

Enabling and Disabling the WDT

The Watchdog Timer is disabled upon a system reset (RESET). To enable the WDT, the application program must set the WDT_EN bit (bit 7) of the WDT_CTL register. When enabled, the WDT cannot be disabled without a RESET.

Time-Out Period Selection

There are four choices of time-out periods for the WDT—2^{18}, 2^{22}, 2^{25}, and 2^{27} system clock cycles. The WDT time-out period is defined by the WDT_PERIOD field of the WDT_CTL register (WDT_CTL[1:0]). The approximate time-out periods for two different WDT clock sources is listed in Table 26.

Table 26. Watchdog Timer Approximate Time-Out Delays

Clock Source	Divider Value	Time Out Delay
32.768 kHz Crystal Oscillator	2^{18}	8.00 s
32.768 kHz Crystal Oscillator	2^{22}	128 s
32.768 kHz Crystal Oscillator	2^{25}	1024 s
32.768 kHz Crystal Oscillator	2^{27}	4096 s
20 MHz System Clock	2^{18}	13.1 ms
20 MHz System Clock	2^{22}	209.7 ms
20 MHz System Clock	2^{25}	1.68s
20 MHz System Clock	2^{27}	6.71s
50 MHz System Clock	2^{18}	5.2 ms*
50 MHz System Clock	2^{22}	83.9 ms*
50 MHz System Clock	2^{25}	0.67 s
50 MHz System Clock	2^{27}	2.68 s

RESET Or NMI Generation

Upon a WDT time-out, the RST_FLAG in the WDT_CTL register is set to 1. In addition, the WDT can cause a RESET or send a nonmaskable interrupt (NMI) signal to the CPU. The default operation is for the WDT to cause a RESET. It asserts/deasserts on the rising edge of the clock. The RST_FLAG bit can be polled by the CPU to determine the source of the RESET event.

If the NMI_OUT bit in the WDT_CTL register is set to 1, then upon time-out, the WDT asserts an NMI for CPU processing. The RST_FLAG bit can be polled by the CPU to

Real Time Clock Year Register

This register contains the current year count. See Table 44.

Table 44. Real Time Clock Year Register (RTC_YR = 00E6h)

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_YR	0–9	The tens digit of the current year count.
[3:0] YR	0–9	The ones digit of the current year count.

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

Real Time Clock Alarm Minutes Register

This register contains the alarm minutes value. See Table 47.

Table 47. Real Time Clock Alarm Minutes Register (RTC_AMIN = 00E9h)

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_MIN	0–5	The tens digit of the alarm minutes value.
[3:0] AMIN	0–9	The ones digit of the alarm minutes value.

Binary Operation (BCD_EN = 0)

Bit Position	Value	Description
[7:0] AMIN	00h–3Bh	The alarm minutes 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.

Universal Asynchronous Receiver/ Transmitter

The UART module implements the logic required to support various asynchronous communications protocols. The module also implements two separate 16-byte-deep FIFOs for both transmission and reception. A block diagram of the UART is illustrated in Figure 23.

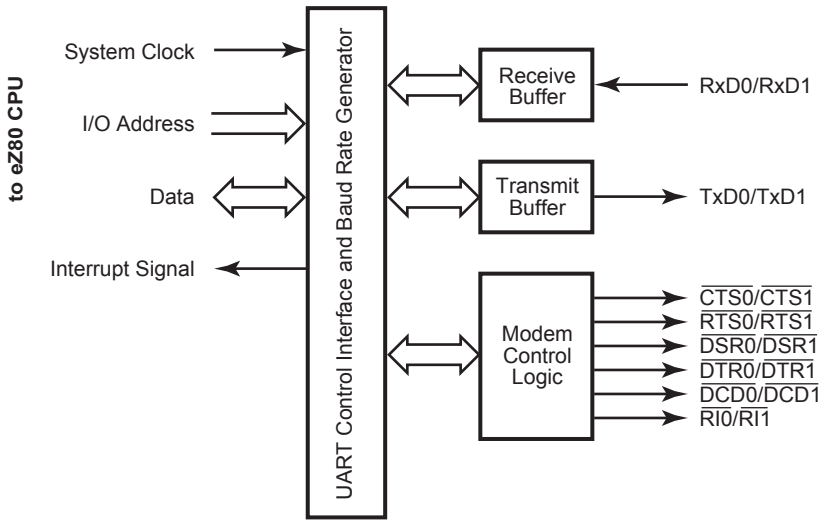


Figure 23. UART Block Diagram

The UART module provides the following asynchronous communication protocol-related features and functions:

- 5-bit, 6-bit, 7-bit, or 8-bit data transmission
- Even parity/odd parity or no parity bit generation and detection
- Start and stop bit generation and detection (supports up to two stop bits)
- Line break detection and generation
- Receiver overrun and framing errors detection
- Logic and associated I/O to provide modem handshake capability

is reset only when the processor reads all of the received data bytes. If the number of bits received is less than eight, the unused most significant bits of the data byte Read are 0.

The receiver uses the clock from the BRG for receiving the data. This clock must be 16 times the appropriate baud rate. The receiver synchronizes the shift clock on the falling edge of the RxD input start bit. It then receives a complete byte according to the set parameters. The receiver also implements logic to detect framing errors, parity errors, overrun errors, and break signals.

UART Modem Control

The modem control logic provides two outputs and four inputs for handshaking with the modem. Any change in the modem status inputs, except $\overline{RI}$, is detected and an interrupt can be generated. For $\overline{RI}$, an interrupt is generated only when the trailing edge of the $\overline{RI}$ is detected. The module also provides LOOP mode for self-diagnostics.

UART Interrupts

There are five different sources of interrupts from the UART are:

- Transmitter.
- Receiver (three different interrupts).
- Modem status.

UART Transmitter Interrupt

The transmitter interrupt is generated if there is no data available for transmission. This interrupt can be disabled using the individual interrupt enable bit or cleared by writing data into the UARTx_THR register.

UART Receiver Interrupts

A receiver interrupt can be generated by three possible sources. The first source, a receiver data ready, indicates that one or more data bytes are received and are ready to be read. This interrupt is generated if the number of bytes in the receiver FIFO is greater than or equal to the trigger level. If the FIFO is not enabled, the interrupt is generated if the receive buffer contains a data byte. This interrupt is cleared by reading the UARTx_RBR.

The second interrupt source is the receiver time-out. A receiver time-out interrupt is generated when there are fewer data bytes in the receiver FIFO than the trigger level and there are no Reads and Writes to or from the receiver FIFO for four consecutive byte times. When the receiver time-out interrupt is generated, it is cleared only after emptying the entire receive FIFO.

The first two interrupt sources from the receiver (data ready and time-out) share an interrupt enable bit.

slave-transmitter must release the data line to allow the master to generate a STOP or a repeated START condition.

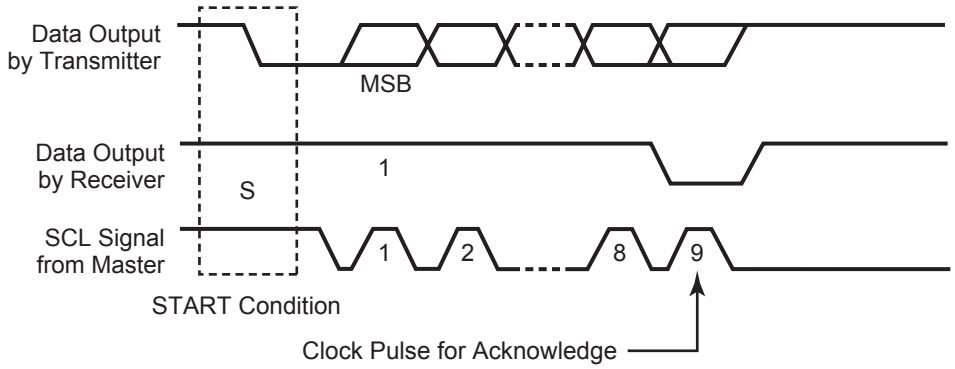


Figure 33. I²C Acknowledge

Clock Synchronization

All masters generate their own clocks on the SCL line to transfer messages on the I²C bus. Data is only valid during the High period of each clock.

Clock synchronization is performed using the wired AND connection of the I²C interfaces to the SCL line, meaning that a High-to-Low transition on the SCL line causes the relevant devices to start counting from their Low period. When a device clock goes Low, it holds the SCL line in that state until the clock High state is reached. See Figure 34. The Low-to-High transition of this clock, however, may not change the state of the SCL line if another clock is still within its Low period. The SCL line is held Low by the device with the longest Low period. Devices with shorter Low periods enter a High wait-state during this time.

When all devices concerned count off their Low period, the clock line is released and goes High. There is no difference between the device clocks and the state of the SCL line, and all of the devices start counting their High periods. The first device to complete its High period again pulls the SCL line Low. In this way, a synchronized SCL clock is generated with its Low period determined by the device with the longest clock Low period, and its High period determined by the one with the shortest clock High period.

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.								

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.

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.

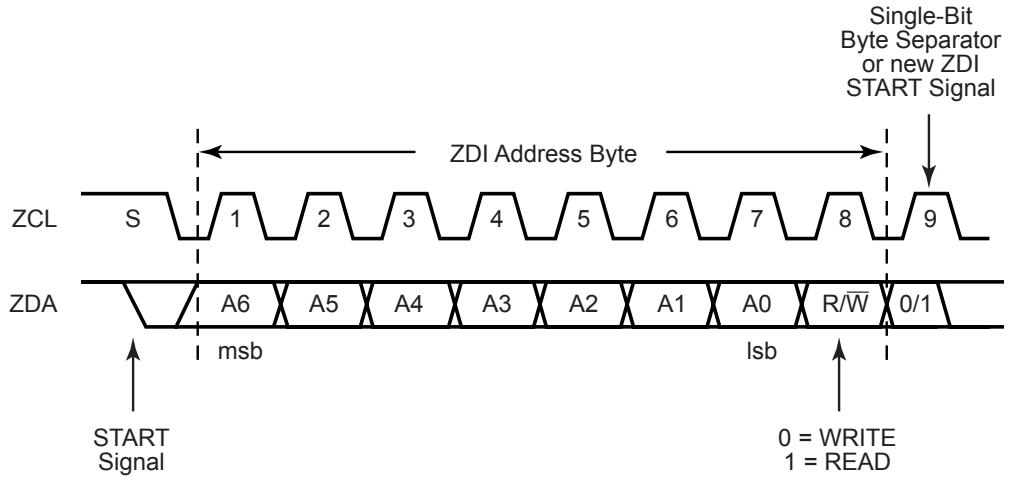


Figure 39. ZDI Address Write Timing

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.

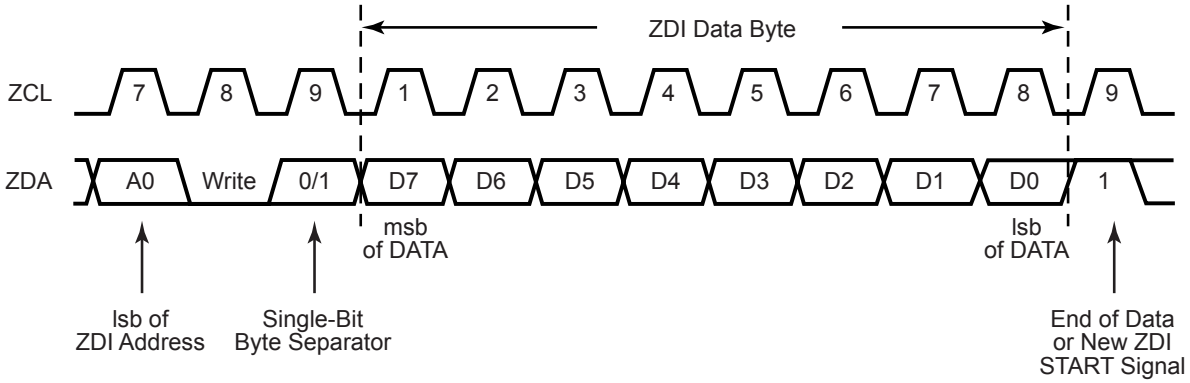


Figure 40. ZDI Single-Byte Data Write Timing

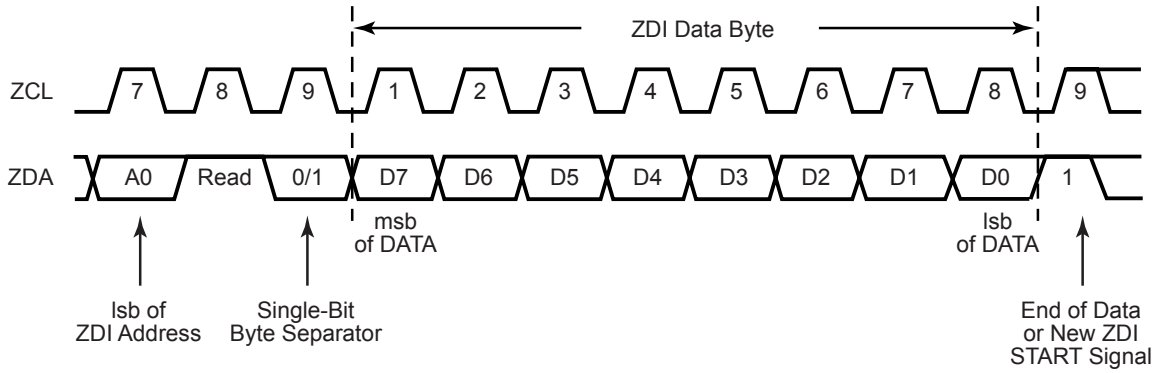


Figure 42. ZDI Single-Byte Data Read Timing

ZDI Block Read

A Block Read operation is initiated the same as a single-byte Read; however, the ZDI master continues to clock in the next byte from the ZDI slave as the ZDI slave continues to output data. The ZDI register address counter increments with each Read. If the ZDI register address reaches the end of the Read-Only ZDI register address space (20h), the address stops incrementing. Figure 43 illustrates the ZDI's Block Read timing.

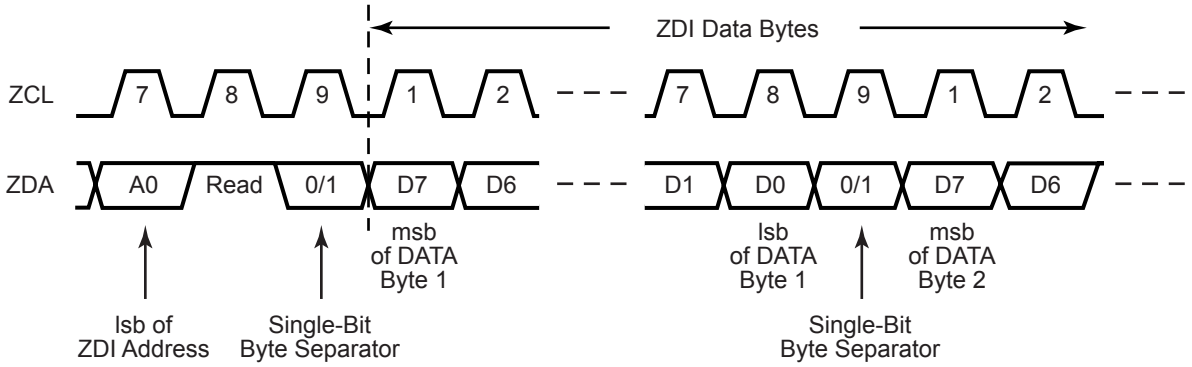


Figure 43. ZDI Block Data Read Timing

Operation of the eZ80L92 During ZDI Break Points

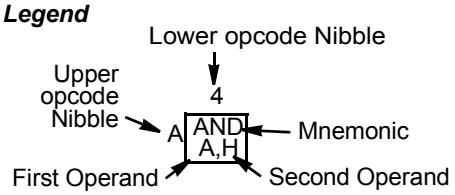
If the ZDI forces the CPU to break, only the CPU suspends operation. The system clock continues to operate and drive other peripherals. Those peripherals that can operate autonomously from the CPU may continue to operate, if so enabled. For example, the Watchdog Timer and Programmable Reload Timers continue to count during a ZDI break point.

Opcode Map

Table 109 through Table 115 indicate the hex values for the eZ80 instructions.

Table 109. Opcode Map—First Opcode

Legend



Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LD BC, Mmn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA
1	DJNZ d	LD DE, Mmn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR d	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA
2	JR NZ,d	LD HL, Mmn	LD (Mmn), HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,d	ADD HL,HL	LD HL, (Mmn)	DEC HL	INC L	DEC L	LD L,n	CPL
3	JR NC,d	LD SP, Mmn	LD (Mmn), A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF	JR CF,d	ADD HL,SP	LD A, (Mmn)	DEC SP	INC A	DEC A	LD A,n	CCF
4	.SIS suffix	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	.LIS suffix	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
5	LD D,B	LD D,C	.SIL suffix	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	.LIL suffix	LD E,H	LD E,L	LD E,(HL)	LD E,A
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A
9	SUB A,B	SUB A,C	SUB A,D	SUB A,E	SUB A,H	SUB A,L	SUB A,(HL)	SUB A,A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A
A	AND A,B	AND A,C	AND A,D	AND A,E	AND A,H	AND A,L	AND A,(HL)	AND A,A	XOR A,B	XOR A,C	XOR A,D	XOR A,E	XOR A,H	XOR A,L	XOR A,(HL)	XOR A,A
B	OR A,B	OR A,C	OR A,D	OR A,E	OR A,H	OR A,L	OR A,(HL)	OR A,A	CP A,B	CP A,C	CP A,D	CP A,E	CP A,H	CP A,L	CP A,(HL)	CP A,A
C	RET NZ	POP BC	JP NZ, Mmn	JP Mmn	CALL NZ, Mmn	PUSH BC	ADD A,n	RST 00h	RET Z	RET	JP Z, Mmn	Table 110	CALL Z, Mmn	CALL Mmn	ADC A,n	RST 08h
D	RET NC	POP DE	JP NC, Mmn	OUT (n),A	CALL NC, Mmn	PUSH DE	SUB A,n	RST 10h	RET CF	EXX	JP CF, Mmn	IN A,(n)	CALL CF, Mmn	Table 111	SBC A,n	RST 18h
E	RET PO	POP HL	JP PO, Mmn	EX (SP),HL	CALL PO, Mmn	PUSH HL	AND A,n	RST 20h	RET PE	JP (HL)	JP PE, Mmn	EX DE,HL	CALL PE, Mmn	Table 112	XOR A,n	RST 28h
F	RET P	POP AF	JP P, Mmn	DI	CALL P, Mmn	PUSH AF	OR A,n	RST 30h	RET M	LD SP,HL	JP M, Mmn	EI	CALL M, Mmn	Table 113	CP A,n	RST 38h

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

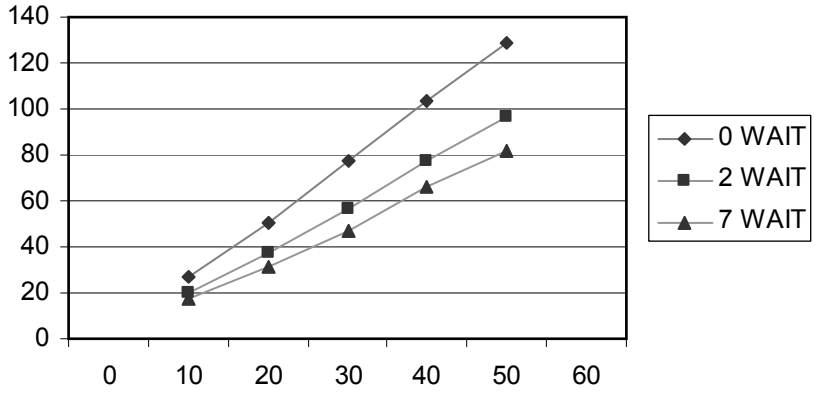


Figure 46. I_{CC} vs. Frequency (Typical at 3.3 V, 25 ºC)

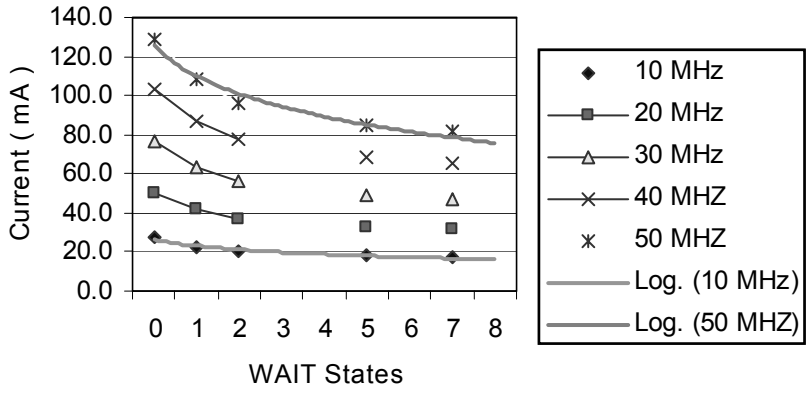


Figure 47. I_{CC} vs. WAIT (Typical at 3.3 V, 25 ºC)

AC Characteristics

The section provides information on the AC characteristics and timing. All AC timing information assumes a standard load of 50 pF on all outputs. See Table 120.

T

U