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	Discontinued at Digi-Key
Core Processor	eZ80
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, Ethernet MAC, I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
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	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80f91gaz0beg

Table 2. Pin Identification on the eZ80F91 ASSP Device (Continued)

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
17	F1	ADDR12	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.
18	F2	ADDR13	Address Bus	Bidirectional	
19	F3	ADDR14	Address Bus	Bidirectional	
20	F4	ADDR15	Address Bus	Bidirectional	
21	G1	ADDR16	Address Bus	Bidirectional	
22	G2	V_{DD}	Power Supply		Power Supply.
23	G3	V_{SS}	Ground		Ground.
24	F5	ADDR17	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.
25	H1	ADDR18	Address Bus	Bidirectional	
26	H2	ADDR19	Address Bus	Bidirectional	
27	G4	ADDR20	Address Bus	Bidirectional	
28	H3	ADDR21	Address Bus	Bidirectional	
29	J1	ADDR22	Address Bus	Bidirectional	
30	G5	ADDR23	Address Bus	Bidirectional	
31	J2	V_{DD}	Power Supply		Power Supply.
32	H4	V_{SS}	Ground		Ground.
33	J3	CS0	Chip Select 0	Output, Active Low	$\overline{CS0}$ Low indicates that an access is occurring in the defined $\overline{CS0}$ memory or I/O address space.
34	K1	CS1	Chip Select 1	Output, Active Low	$\overline{CS1}$ Low indicates that an access is occurring in the defined $\overline{CS1}$ memory or I/O address space.
35	K2	CS2	Chip Select 2	Output, Active Low	$\overline{CS2}$ Low indicates that an access is occurring in the defined $\overline{CS2}$ memory or I/O address space.
36	L1	CS3	Chip Select 3	Output, Active Low	$\overline{CS3}$ Low indicates that an access is occurring in the defined $\overline{CS3}$ memory or I/O address space.
37	M1	V_{DD}	Power Supply		Power Supply.
38	M2	V_{SS}	Ground		Ground.

Table 2. Pin Identification on the eZ80F91 ASSP Device (Continued)

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
53	M6	INSTRD	Instruction Read Indicator	Output, Active Low	$\overline{\text{INSTRD}}$ (with $\overline{\text{MREQ}}$ and $\overline{\text{RD}}$) indicates the eZ80F91 device is fetching an instruction from memory. This pin is in a high-impedance state during bus acknowledge cycles.
54	L6	WAIT	WAIT Request	Schmitt Trigger input, Active Low	Driving the $\overline{\text{WAIT}}$ pin Low forces the CPU to wait additional clock cycles for an external peripheral or external memory to complete its read or write operation.
55	K6	RESET	Reset	Bidirectional, Active Low Schmitt Trigger input or open drain output	This signal is used to initialize the eZ80F91, and/or allow the eZ80F91 to signal when it resets. See the Reset chapter on page 38 for the timing details. This Schmitt Trigger input allows for RC rise times.
56	J6	NMI	Nonmaskable Interrupt	Schmitt Trigger input, Active Low, edge-triggered interrupt	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 for RC rise times.
57	M7	BUSREQ	Bus Request	Schmitt Trigger input, Active Low	External devices request the eZ80F91 device to release the memory interface bus for their use by driving this pin Low.
58	L7	BUSACK	Bus Acknowledge	Output, Active Low	The eZ80F91 device responds to a Low on $\overline{\text{BUSREQ}}$ making the address, data, and control signals high impedance, 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.
59	K7	V_{DD}	Power Supply		Power Supply.
60	H6	V_{SS}	Ground		Ground.

eZ80 CPU Core

The eZ80 CPU is the first 8-bit CPU to support 16MB linear addressing. Each software module or task under a real-time executive or operating system operates in Z80-compatible (64KB) mode or full 24-bit (16MB) address mode.

The CPU instruction set is a superset of the instruction sets for the Z80 and Z180 CPUs. Z80 and Z180 programs can be executed on an eZ80 CPU with little or no modification.

Features

The features of eZ80 CPU include:

- Code-compatible with Z80 and Z180 products
- 24-bit linear address space
- Single-cycle instruction fetch
- Pipelined fetch, decode, and execute
- Dual stack pointers for ADL (24-bit) and Z80 (16-bit) memory modes
- 24-bit CPU registers and Arithmetic Logic Unit (ALU)
- Debug support
- Nonmaskable Interrupt (NMI), plus support for 128 maskable vectored interrupts

New Instructions

Two new eZ80 CPU instructions load/unload the I Register with a 16-bit value. These new instructions are:

- LD I,HL (ED C7)
- LD HL,I (ED D7)

For more information about the eZ80 CPU, its instruction set, and eZ80 programming, refer to the eZ80 CPU User Manual (UM0077), which is available free for download from the Zilog website.

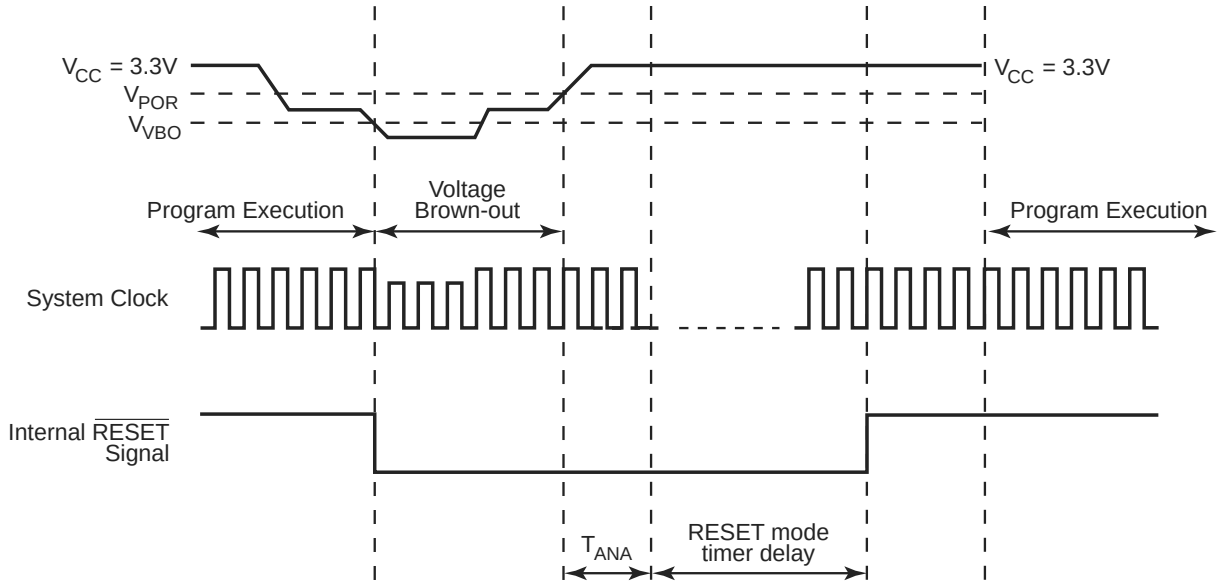

Figure 4. Voltage Brown-Out Reset Operation

General-Purpose Input/Output

The eZ80F91 device features 32 General-Purpose Input/Output (GPIO) pins. The GPIO pins are assembled as four 8-bit ports: Port A, Port B, Port C, and Port D. All port signals are configured as either inputs or outputs. In addition, all of the port pins are used as vectored interrupt sources for the CPU.

The eZ80F91 ASSPs GPIO ports are slightly different from its eZ80 predecessors. Specifically, Port A pins source 8 mA and sink 10 mA. In addition, the Port B and C inputs now feature Schmitt Trigger input buffers.

GPIO Operation

GPIO operation is the same for all four GPIO ports (Ports A, B, C, and D). Each port features eight GPIO port pins. The operating mode for each pin is controlled by four bits that are divided between four 8-bit registers. The GPIO mode control registers are:

- Port *x* Data Register (Px_DR)
- Port *x* Data Direction Register (Px_DDR)
- Port *x* Alternate Register 1 (Px_ALT1)
- Port *x* Alternate Register 2 (Px_ALT2)

In the above list, *x* can be A, B, C or D, representing any of the four GPIO ports. The mode for each pin is controlled by setting each register bit pertinent to the pin to be configured. For example, the operating mode for port B pin 7 (PB7) is set by the values contained in PB_DR[7], PB_DDR[7], PB_ALT1[7], and PB_ALT2[7].

The combination of the GPIO control register bits allows individual configuration of each port pin for nine modes. In all modes, reading of the Port *x* Data Register returns the sampled state or level of the signal on the corresponding pin. Table 6 indicates the function of each port signal based on these four register bits. After a RESET event, all GPIO port pins are configured as standard digital inputs with the interrupts disabled.

In addition to the four mode control registers, each port has an 8-bit register, which is used for clearing edge-triggered interrupts. This register is the Port *x* Alternate Register 0 (Px_ALT0), in which *x* can be A, B, C or D representing the four GPIO ports. When a GPIO pin is configured as an edge-triggered interrupt, writing 1 to the corresponding bit of the Px_ALT0 Register clears the interrupt.

$\overline{\text{WAIT}}$ Input Signal

Similar to the programmable wait states, an external peripheral drives the $\overline{\text{WAIT}}$ input pin to force the CPU to provide additional clock cycles to complete its read or write operation. Driving the $\overline{\text{WAIT}}$ pin Low stalls the CPU. The CPU resumes operation on the first rising edge of the internal system clock following deassertion of the $\overline{\text{WAIT}}$ pin.

Caution: If the $\overline{\text{WAIT}}$ pin is to be driven by an external device, the corresponding chip select for the device must be programmed to provide at least one wait state. Due to input sampling of the $\overline{\text{WAIT}}$ input pin (see Figure 8), one programmable wait state is required to allow the external peripheral sufficient time to assert the $\overline{\text{WAIT}}$ pin. It is recommended that the corresponding chip select for the external device be programmed to provide the maximum number of wait states (seven).

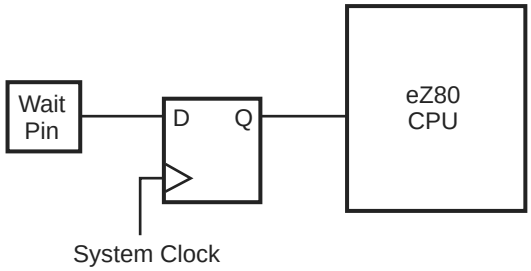

Figure 8. Wait Input Sampling Block Diagram

An example of wait state operation is shown in Figure 9. In this example, the chip select is configured to provide a single wait state. The external peripheral accessed drives the $\overline{\text{WAIT}}$ pin Low to request assertion of an additional wait state. If the $\overline{\text{WAIT}}$ pin is asserted for additional system clock cycles, wait states are added until the $\overline{\text{WAIT}}$ pin is deasserted (active High).

Bus Mode Controller

The bus mode controller allows the address and data bus timing and signal formats of the eZ80F91 to be configured to connect with external devices compatible with eZ80, Z80, Intel and Motorola microcontrollers. Bus modes for each of the chip selects are configured independently using the Chip Select Bus Mode Control Registers. The number of CPU system clock cycles per bus mode state is also independently programmable. For Intel bus mode, multiplexed address and data are selected in which both the lower byte of the address and the data byte use the data bus, DATA[7:0]. Each of the bus modes are explained in the following sections.

eZ80 BUS Mode

Chip selects configured for eZ80 BUS Mode do not modify the bus signals from the CPU. The timing diagrams for external Memory and I/O read and write operations are shown in the AC Characteristics section on page 343. The default mode for each chip select is eZ80 Mode.

Z80 BUS Mode

Chip selects configured for Z80 Mode modify the eZ80 bus signals to match the Z80 microprocessor address and data bus interface signal format and timing. During read operations, the Z80 bus mode employs three states: T1, T2, and T3, as described in Table 19.

Table 19. Z80 BUS Mode Read States

STATE T1	The read cycle begins in State T1. The CPU drives the address onto the address bus and the associated chip select signal is asserted.
STATE T2	During State T2, the $\overline{RD}$ signal is asserted. Depending on the instruction, either the $\overline{MREQ}$ or $\overline{IORQ}$ signal is asserted. If the external $\overline{WAIT}$ pin is driven Low at least one CPU system clock cycle prior to the end of State T2, additional wait states (T_{WAIT}) are asserted until the $\overline{WAIT}$ pin is driven High.
STATE T3	During State T3, no bus signals are altered. The data is latched by the eZ80F91 at the rising edge of the CPU system clock at the end of State T3.

Bit	Description (Continued)
[1:0] WDT_PERIOD	**Watchdog Timer Period**
	00: WDT_CLK = 00: WDT time-out period is 2^{27} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{17} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{15} clock cycles. WDT_CLK = 11: reserved.
	01: WDT_CLK = 00: WDT time-out period is 2^{25} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{14} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{13} clock cycles. WDT_CLK = 11: reserved.
	10: WDT_CLK = 00: WDT time-out period is 2^{22} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{11} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{9} clock cycles. WDT_CLK = 11: reserved.
	11: WDT_CLK = 00: WDT time-out period is 2^{18} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{7} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{5} clock cycles. WDT_CLK = 11: reserved.

Note: When the WDT is enabled, no writes are allowed to the WDT_CTL Register.

Timer Interrupt Enable Register

The Timer *x* Interrupt Enable Register, shown in Table 55, is used to control timer interrupt operations. Only bits related to functions present in a given timer are active.

Table 55. Timer Interrupt Enable (TMR*x*_IER)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	IRQ_OC*x*_EN				IRQ_ ICB_EN	IRQ_ ICA_EN	IRQ_ EOC_EN
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	TMR0_IER = 0061h, TMR1_IER = 0066h, TMR2_IER = 0070h, TMR3_IER = 0075h							

Note: R = read only; R/W = read/write.

Bit	Description
[7]	**Reserved** This bit is unused and must be programmed to 0.
[6] IRQ_OC3_EN	**Interrupt Request Output Compare 3 Enable** 0: Interrupt requests for OC3 are disabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3. 1: Interrupt requests for OC3 are enabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3.
[5] IRQ_OC2_EN	**Interrupt Request Output Compare 2 Enable** 0: Interrupt requests for OC2 are disabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3. 1: Interrupt requests for OC2 are enabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3.
[4] IRQ_OC1_EN	**Interrupt Request Output Compare 1 Enable** 0: Interrupt requests for OC1 are disabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3. 1: Interrupt requests for OC1 are enabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3.
[3] IRQ_OC0_EN	**Interrupt Request Output Compare 0 Enable** 0: Interrupt requests for OC0 are disabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3. 1: Interrupt requests for OC0 are enabled (valid only in OUTPUT COMPARE Mode). OC operations occur in Timer 3.

Timer Data High Byte Register

The Timer *x* Data High Byte Register, shown in Table 58, returns the high byte of the count value of the selected timer as it existed at the time that the low byte was read. The Timer Data High Byte Register is read when the timer is in operation. Reading the current count value does not affect timer operation. To read the 16-bit data of the current count value, {TMR*x*_DR_H[7:0], TMR*x*_DR_L[7:0]}, first read the Timer Data Low Byte Register followed by the Timer Data High Byte Register. The Timer Data High Byte Register value is latched into temporary storage when a read of the Timer Data Low Byte Register occurs.

This register shares its address with the corresponding timer reload register.

Table 58. Timer Data High Byte Register (TMR*x*_DR_H)

Bit	7	6	5	4	3	2	1	0
Field	TMR*x*_DR_H							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	TMR0_DR_H = 0064h, TMR1_DR_H = 0069h, TMR2_DR_H = 0073h, TMR3_DR_H = 0078h							

Note: R = read only.

Bit	Description
[7:0] TMR_DR_H	**Timer Data Low Byte** 00h–FFh: These bits represent the high byte of the 2-byte timer data value, {TMR*x*_DR_H[7:0], TMR*x*_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.

Table 60. Timer Reload High Byte Register (TMR*x*_RR_H)

Bit	7	6	5	4	3	2	1	0
Field	TMR_RR_H							
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	TMR0_RR_H = 0064h, TMR1_RR_H = 0069h, TMR2_RR_H = 0073h, TMR3_RR_H = 0078h							

Note: W = write only.

Bit	Description
[7:0] TMR_RR_H	**Timer Reload High Byte** 00h–FFh: These bits represent the high byte of the 2-byte timer reload value, {TMR*x*_RR_H[7:0], TMR*x*_RR_L[7:0]}. Bit 7 is bit 15 (msb) of the 16-bit timer reload value. Bit 0 is bit 8 of the 16-bit timer reload value.

Timer Input Capture Control Register

The Timer *x* Input Capture Control Register, shown in Table 61, is used to select the edge or edges to be captured. For Timer 1, CAP_EDGE_B is used for IC1 and CAP_EDGE_A is for IC0. For Timer 3, CAP_EDGE_B is for IC3, and CAP_EDGE_A is for IC2.

Table 61. Timer Input Capture Control Register (TMR1_CAP_CTL, TMR3_CAP_CTL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				CAP_EDGE_B		CAP_EDGE_A	
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	TMR1_CAP_CTL = 006Ah, TMR3_CAP_CTL = 007Bh							

Note: R = read only; R/W = read/write.

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3:2] CAP_EDGE_B	**Capture Edge Enable B** 00: Disable capture on ICB. 01: Enable capture only on the falling edge of ICB. 10: Enable capture only on the rising edge of ICB. 11: Enable capture on both edges of ICB.

UART Functional Description

The UART Baud Rate Generator (BRG) creates the clock for the serial transmit and receive functions. The UART module supports all of the various options in the asynchronous transmission and reception protocol including:

- 5- to 9-bit transmit/receive
- Start bit generation and detection
- Parity generation and detection
- Stop bit generation and detection
- Break generation and detection

The UART contains 16-byte-deep FIFOs in each direction. The FIFOs are enabled or disabled by the application. The receive FIFO features trigger-level detection logic, which enables the CPU to block-transfer data bytes from the receive FIFO.

UART Functions

The UART function implements:

- The transmitter and associated control logic
- The receiver and associated control logic
- The modem interface and associated logic

UART Transmitter

The transmitter block controls the data transmitted on the TxD output. It implements the FIFO, access via the UARTx_THR Register, the transmit shift register, the parity generator, and control logic for the transmitter to control parameters for the asynchronous communications protocol.

The UARTx_THR is a write-only register. The CPU writes the data byte to be transmitted into this register. In FIFO Mode, up to 16 data bytes are written via the UARTx_THR Register. The data byte from the FIFO is transferred to the transmit shift register at the appropriate time and transmitted via TxD output. After SYNC_RESET, the UARTx_THR Register is empty. Therefore, the Transmit Holding Register Empty (THRE) bit (bit 5 of the UARTx_LSR Register) is 1. An interrupt is sent to the CPU if interrupts are enabled. The CPU resets this interrupt by loading data into the UARTx_THR Register, which clears the transmitter interrupt.

The transmit shift register places the byte to be transmitted on the TxD signal serially. The least-significant bit of the byte to be transmitted is shifted out first and the most-significant

meet IrDA specifications. The UART must be enabled to use the endec. For more information about the UART and its BRG, see the Universal Asynchronous Receiver/Transmitter chapter on page 172.

Transmit

The data to be transmitted via the IR transceiver is the data sent to UART0. The UART transmit signal, TxD, and Baud Rate Clock are used by the endec to generate the modulation signal, IR_TxD, that drives the infrared transceiver. Each UART bit is 16 clocks wide. If the data to be transmitted is a logic 1 (High), the IR_TxD signal remains Low (0) for the full 16-clock period. If the data to be transmitted is a logic 0, a 3-clock High (1) pulse is output following a 7-clock Low (0) period. Following the 3-clock High pulse, a 6-clock Low pulse completes the full 16-clock data period. Data transmission is shown in Figure 38. During data transmission, the IR receive function must be disabled by clearing the IR_RxEN bit in the IR_CTL reg to 0 to prevent transmitter-to-receiver crosstalk.

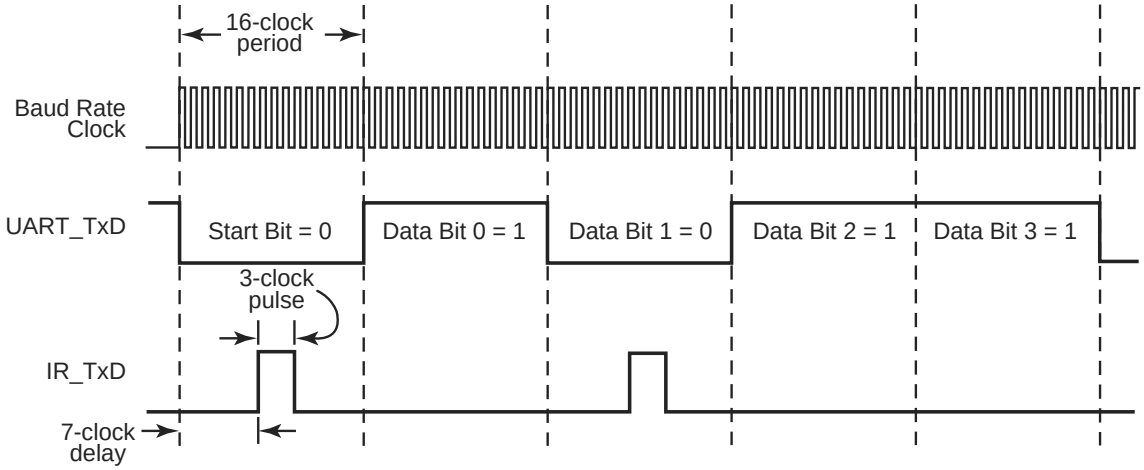

Figure 38. Infrared Data Transmission

Receive

Data received from the IR transceiver via the IR_RxD signal is decoded by the endec and passed to the UART. The IR_RxEN bit in the IR_CTL Register must be set to enable the receiver decoder. The IrDA serial infrared (SIR) data format uses half duplex communication. Therefore, the UART must not be allowed to transmit while the receiver decoder is enabled. The UART Baud Rate Clock is used by the endec to generate the demodulated signal, RxD, that drives the UART. Each UART bit is 16 clocks wide. If the data to be received is a logic 1 (High), the IR_RxD signal remains High (1) for the full 16-clock

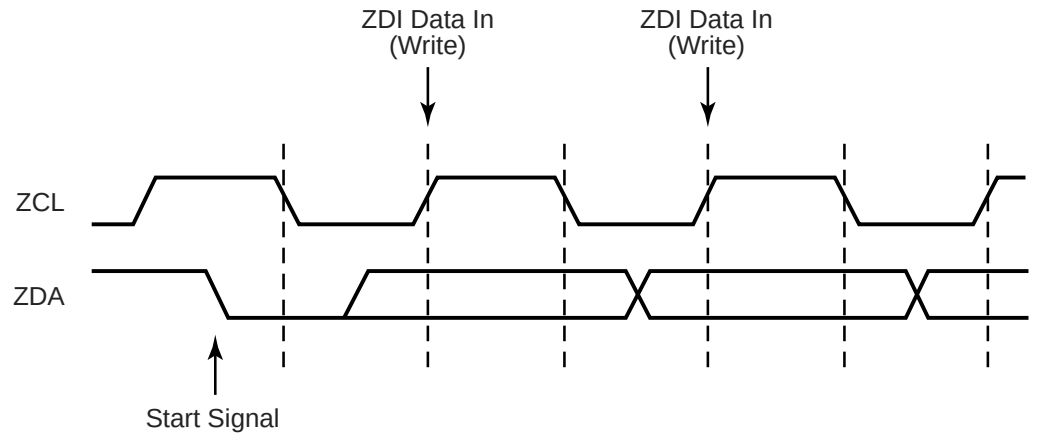

Figure 50. ZDI Write Timing

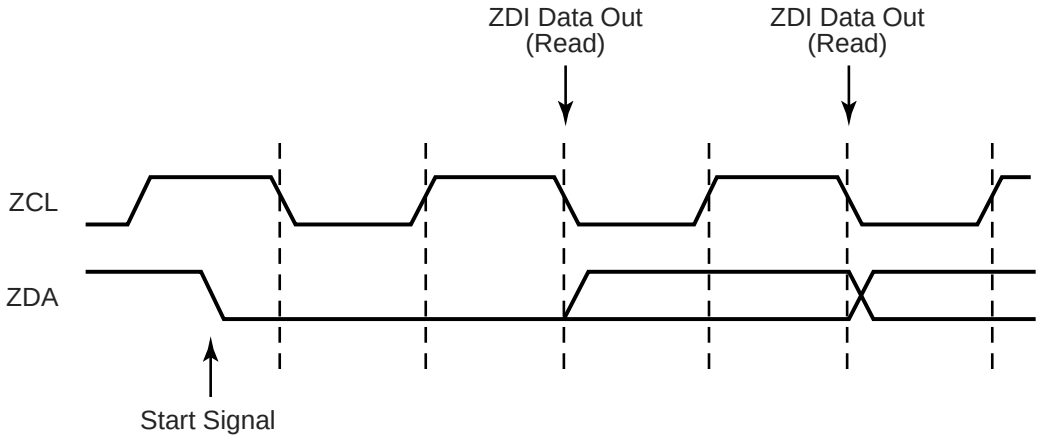

Figure 51. ZDI Read Timing

ZDI Single-Bit Byte Separator

Following each 8-bit ZDI data transfer, a single-bit byte separator is used. To initiate a new ZDI command, the single-bit byte separator must be High (logic 1) to allow for a new ZDI start command to be sent. For all other cases, the single-bit byte separator is either Low (logic 0) or High (logic 1). When ZDI is configured to allow the CPU to accept external bus requests, the single-bit byte separator must be Low (logic 0) during all ZDI commands. This Low value indicates that ZDI is still operating and is not ready to relinquish the bus. The CPU does not accept the external bus requests until the single-bit byte separa-

Bit	Description
[7] ZDI_RESET	**ZDI System Reset** 0: No action. 1: Initiate a RESET of the eZ80F91 MCU. This bit is automatically cleared at the end of the RESET event.
[6:0]	**Reserved** These bits are reserved and must be programmed to 0000000.

ZDI Write Data Registers

These three registers are used in the ZDI write-only register address space to store the data that is written when a write instruction is sent to the ZDI Read/Write Control Register (ZDI_RW_CTL). The ZDI Read/Write Control Register is located at ZDI address 16h immediately following the ZDI Write Data registers. As a result, the ZDI Master is allowed to write the data to {ZDI_WR_U, ZDI_WR_H, ZDI_WR_L} and the write command in one data transfer operation. See Table 139.

Table 139. ZDI Write Data Registers (ZDI_WR_U, ZDI_WR_H, ZDI_WR_L)

Bit	7	6	5	4	3	2	1	0
Field	ZDI_WR_L, ZDI_WR_H or ZDI_WR_L							
Reset	U	U	U	U	U	U	U	U
R/W	W	W	W	W	W	W	W	W
Address	ZDI_WR_U = 13h, ZDI_WR_H = 14h and ZDI_WR_L = 15h in the ZDI Register write-only address space							

Note: U = undefined; W = write.

Bit	Description
[7:0] ZDI_WR_L, ZDI_WR_H, or ZDI_WR_L	**ZDI Write Data** 00h–FFh: These registers contain the data that is written during execution of a write operation defined by the ZDI_RW_CTL Register. The 24-bit data value is stored as {ZDI_WR_U, ZDI_WR_H, ZDI_WR_L}. If less than 24 bits of data are required to complete the required operation, the data is taken from the least-significant byte(s).

ZDI Read/Write Control Register

The ZDI Read/Write Control Register is used in the ZDI write-only register address to read data from, write data to, and manipulate the CPU's registers or memory locations. When this register is written, the eZ80F91 device immediately performs the operation corresponding to the data value written as described in Table 140. When a read operation is executed via this register, the requested data values are placed in the ZDI Read Data registers {ZDI_RD_U, ZDI_RD_H, ZDI_RD_L}. When a write operation is executed via this

Table 180. Receive Descriptor Status (Continued)

Bit	Name	Description
10	RxCF	1 = The packet is a control frame.
9	RxMcPkt	1 = The packet contains a multicast address.
8	RxBcPkt	1 = The packet contains a broadcast address.
7	RxVLAN	1 = The packet is a VLAN packet.
6	RxUOpCode	1 = An unsupported op code is indicated in the op code field of the Ethernet packet.
5	RxLOOR	1 = The Type/Length field is out of range (larger than 1518 bytes).
4	RxLCError	1 = Type/Length field is not a Type field and it does not match the actual data byte length of the Ethernet packet. The data byte length is the number of bytes of data in the Ethernet packet between the Type/ Length field and the FCS.
3	RxCodeV	1 = A code violation is detected. The PHY asserts Rx error (RxER).
2	RxCEvent	1 = A carrier event is previously seen. This event is defined as Rx error RxER = 1, receive data valid (RxDV) = 0 and receive data (RxD) = Eh.
1	RxDvEvent	1 = A receive data (RxDV) event is previously seen. Indicates that the last Receive event is not long enough to be a valid packet.
0	RxOVR	1 = A Receive overrun occurs in this packet. An overrun occurs when all of the EMAC Receive buffers are in use and the Receive FIFO is full. The hardware ignores all incoming packets until the EmacIStat Register [Rx_Ovr] bit is cleared by the software. There is no indication as to how many packets are ignored.

EMAC and the System Clock

Effective Ethernet throughput in any given system is dependent upon factors such as system clock speed, network protocol overhead, application complexity, and network traffic conditions at any given moment. The following information provides a general guideline about the effects of system clock speed on Ethernet operation.

The eZ80F91 ASSP's EMAC block performs a synchronous function that is designed to operate over a wide range of system clock frequencies. To understand its maximum data transfer capabilities at certain system operating frequencies, you must first understand the internal data bus bandwidth that is required under ideal conditions.

For 10BaseT Ethernet connectivity, the data rate is 10 Mbps, which equates to 1.25 Mbps. If the eZ80F91 ASSP is operating in FULL-DUPLEX Mode over 10BaseT, the data rate for RX data and TX data is 1.25Mbps. Because raw data transfers at this rate consume a certain amount of CPU bandwidth, the CPU must support traffic from both directions as well as operate at a minimum clock frequency of (1.25 + 1.25) * 2 = 5MHz while transferring Ethernet packets to and from the physical layer.

EMAC Transmit Polling Timer Register

This register sets the Transmit Polling Period in increments of TPTMR = SYSCLK ÷ 256. Whenever this register is written, the status of the Transmit Buffer Descriptor is checked to determine if the EMAC owns the Transmit buffer. It then rechecks this status every TPTMR (calculated by TPTMR x EMAC_PTMR[7:0]). The Transmit Polling Timer is disabled if this register is set to 00h (which also disables the transmitting of packets). If a transmission is in progress when EMAC_PTMR is set to 00h, the transmission will complete. See Table 204.

Table 204. EMAC Transmit Polling Timer Register (EMAC_PTMR)

Bit	7	6	5	4	3	2	1	0
Field	EMAC_PTMR							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	0040h							

Note: R/W = read/write.

Bit	Description
[7:0] EMAC_PTMR	**Transmit Polling Timer** 00h–FFh: The transmit polling period.

EMAC Reset Control Register

The bit values in the EMAC Reset Control Register, shown in Table 205, are not self-clearing bits. You are responsible for controlling their state.

Table 205. EMAC Reset Control Register (EMAC_RST)

Bit	7	6	5	4	3	2	1	0
Field	Reserved		SRST	HRTFN	HRRFN	HRTMC	HRRMC	HRMGT
Reset	0	0	1	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Address	0041h							

Note: R = read only; R/W = read/write.

Table 212. EMAC Receive High Boundary Pointer High Byte Register (EMAC_RHBP_H)

Bit	7	6	5	4	3	2	1	0
Field	EMAC_RHBP_H							
Reset	1	1	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Address	0048h							

Note: R = read only, R/W = read/write.

Bit	Description
[7:0] EMAC_RHBP_H	**Receive High Boundary Pointer High Byte** 00h–FFh: These bits represent the high byte of the two-byte EMAC Receive High Boundary Pointer value, {EMAC_RHBP_H, EMAC_RHBP_L}. Bit 7 is bit 15 (msb) of the 16-bit value. Bit 0 is bit 8 of the 16-bit value.

Note: *Bits 7:5 are not used by the EMAC; these bits return 000 upon reset.

EMAC Receive Read Pointer High and Low Byte Registers

The Receive Read Pointer Registers, shown in Tables 213 and 214, must be initialized to the EMAC_BP value (i.e., the start of the Receive buffer). This register points to the address location in which the next Receive packet is read from. The EMAC_BP[12:5] is loaded into this register whenever the EMAC_RST [(HRRFN) is set to 1. The RxDMA block uses Emac_Rrp[12:5] to compare to EmacRwp[12:5] for determining how many buffers remain. The result equates to the EmacBlksLeft Register.

Table 213. EMAC Receive Read Pointer Low Byte Register (EMAC_RRP_L)

Bit	7	6	5	4	3	2	1	0
Field	EMAC_RRP_L							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R
Address	0049h							

Note: R = read only, R/W = read/write.

Bit	Description
[7:0] EMAC_RRP_L	**Receive Read Pointer Low Byte** 00h–FFh: These bits represent the low byte of the two-byte EMAC Receive Read Pointer value, {EMAC_RRP_H, EMAC_RRP_L}. Bit 7 is bit 7 of the 16-bit value. Bit 0 is bit 0 (lsb) of the 16-bit value.

E