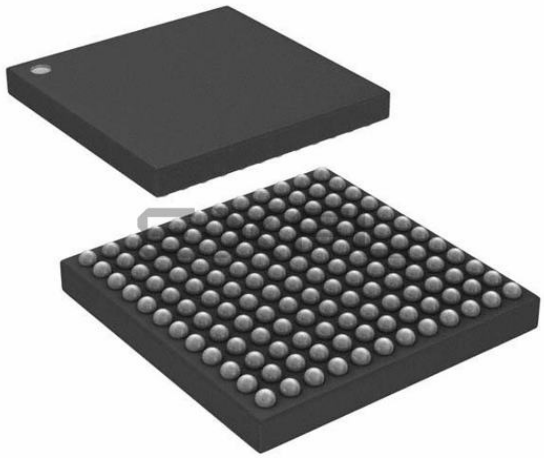

Welcome to E-XFL.COM

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	Ethernet, 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	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LBGA
Supplier Device Package	144-BGA (13x13)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80f91naa50sg

Revision History

Each instance in the following revision history table reflects a change to this document from its previous version. For more details, refer to the corresponding pages provided in the table.

Date	Revision Level	Description	Page Number
May 2012	03	Updated to reference the eZ80AcclaimPlus! Development Kit (eZ80F910300KITG).	354
Oct 2008	02	Updated Table 1, Addressing section in I^2C Serial I/O Interface chapter, Part Number Description, Figure 6, Flash Program Control Register, UART Transmitter, and Figure 40.	4, 47, 109, 173, 198, 220, 355
Jul 2007	01	Original Issue.	All

Table 238. Typical 144-LQFP Package Electrical Characteristics 343
Table 239. External Memory Read Timing 344
Table 240. External Memory Write Timing 345
Table 241. External I/O Read Timing 347
Table 242. External I/O Write Timing 348
Table 243. GPIO Port Output Timing 352
Table 244. Bus Acknowledge Timing 352
Table 245. Ordering Information 354

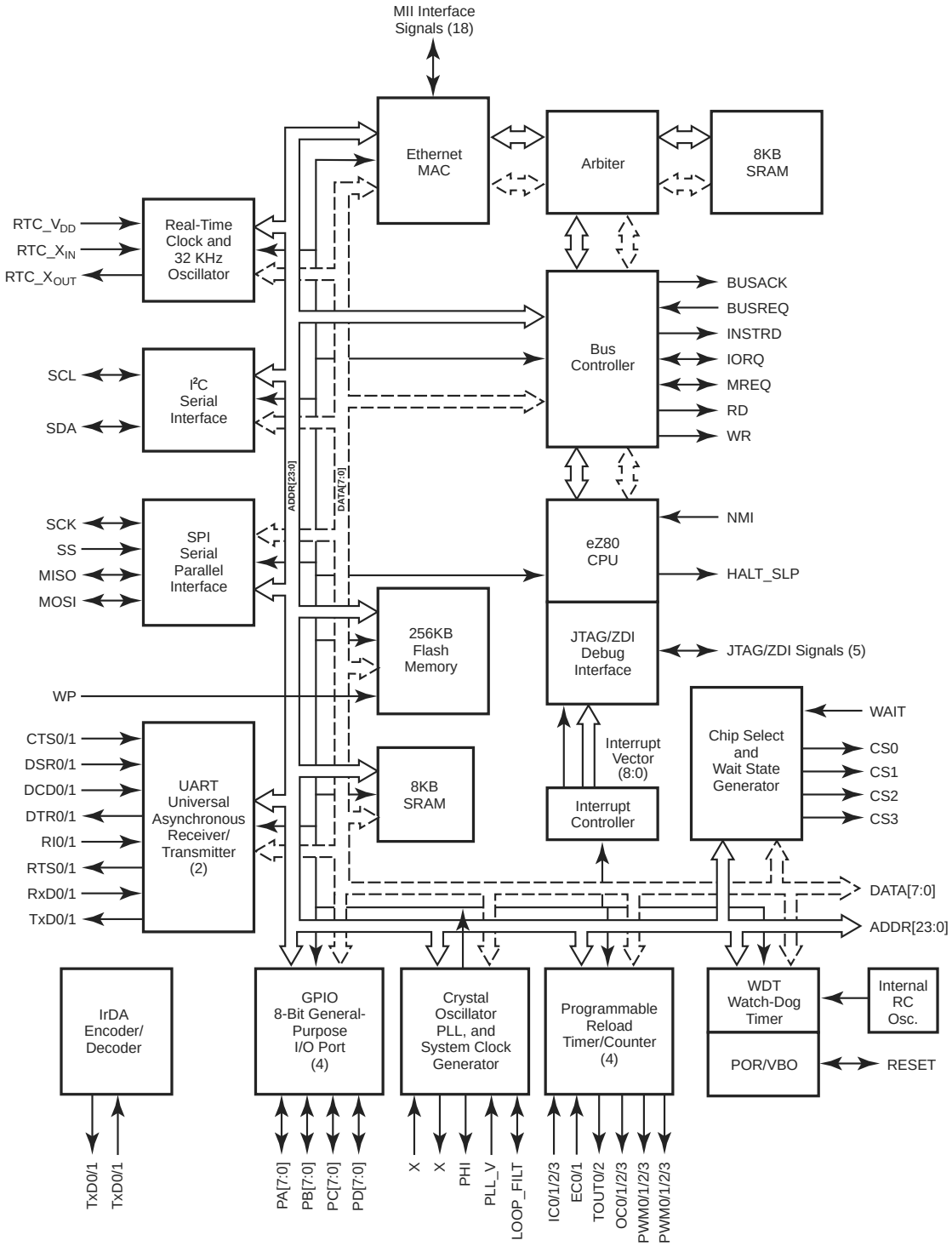

Figure 1. eZ80F91 ASSP Block Diagram

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

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
86	H11	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 must be connected to this pin. When a crystal is used, it must be connected between X_{IN} and X_{OUT}.
87	H10	PLL_V_{DD}	Power Supply		Power Supply for Analog PLL.
88	H9	V_{DD}	Power Supply		Power Supply.
89	G12	V_{SS}	Ground		Ground.
90	G11	PC0	GPIO Port C	Bidirectional with Schmitt Trigger input	This pin is used for GPIO. It is individually programmed as input or output and is also used individually as an interrupt input. Each Port C pin, when programmed as output is selected to be an open-drain or open-source output. Port C is multiplexed with one UART.
		TxD1	Transmit Data	Output	This pin is used by the UART to transmit asynchronous serial data. This signal is multiplexed with PC0.
91	G10	PC1	GPIO Port C	Bidirectional with Schmitt Trigger input	This pin is used for GPIO. It is individually programmed as input or output and is also used individually as an interrupt input. Each Port C pin, when programmed as output is selected to be an open-drain or open-source output. Port C is multiplexed with one UART.
		RxD1	Receive Data	Schmitt Trigger input	This pin is used by the UART to receive asynchronous serial data. This signal is multiplexed with PC1.

Table 3. Register Map (Continued)

Address (hex)	Mnemonic	Name	Reset (hex)	CPU Access	Page No
00A9	CS0_UBR	Chip Select 0 Upper Bound Register	FF	R/W	86
00AA	CS0_CTL	Chip Select 0 Control Register	E8	R/W	87
00AB	CS1_LBR	Chip Select 1 Lower Bound Register	00	R/W	85
00AC	CS1_UBR	Chip Select 1 Upper Bound Register	00	R/W	86
00AD	CS1_CTL	Chip Select 1 Control Register	00	R/W	87
00AE	CS2_LBR	Chip Select 2 Lower Bound Register	00	R/W	85
00AF	CS2_UBR	Chip Select 2 Upper Bound Register	00	R/W	86
00B0	CS2_CTL	Chip Select 2 Control Register	00	R/W	87
00B1	CS3_LBR	Chip Select 3 Lower Bound Register	00	R/W	85
00B2	CS3_UBR	Chip Select 3 Upper Bound Register	00	R/W	86
00B3	CS3_CTL	Chip Select 3 Control Register	00	R/W	87
Random Access Memory Control					
00B4	RAM_CTL	RAM Control Register	C0	R/W	94
00B5	RAM_ADDR_U	RAM Address Upper Byte Register	FF	R/W	95
00B6	MBIST_GPR	General Purpose RAM MBIST Control	00	R/W	96
00B7	MBIST_EMR	Ethernet MAC RAM MBIST Control	00	R/W	96
Serial Peripheral Interface					
00B8	SPI_BRG_L	SPI Baud Rate Generator Low Byte Register	02	R/W	209
00B9	SPI_BRG_H	SPI Baud Rate Generator High Byte Register	00	R/W	209
00BA	SPI_CTL	SPI Control Register	04	R/W	210
00BB	SPI_SR	SPI Status Register	00	R	211
00BC	SPI_TSR	SPI Transmit Shift Register	XX	W	212
	SPI_RBR	SPI Receive Buffer Register	XX	R	212
Infrared Encoder/Decoder					
00BF	IR_CTL	Infrared Encoder/Decoder Control	00	R/W	201

Low-Power Modes

The eZ80F91 device provides a range of power-saving features. The highest level of power reduction is provided by SLEEP Mode with all peripherals disabled, including VBO. The next level of power reduction is provided by the HALT instruction. The most basic level of power reduction is provided by the clock peripheral power-down registers.

SLEEP Mode

Execution of the CPU's SLP instruction puts the eZ80F91 device into SLEEP Mode. In SLEEP Mode, the operating characteristics are:

- The primary crystal oscillator is disabled.
- The system clock is disabled.
- The CPU is idle.
- The Program Counter (PC) stops incrementing.
- The 32 kHz crystal oscillator continues to operate and drives the real-time clock and WDT (if WDT is configured to operate from the 32 kHz oscillator).

The CPU is brought out of SLEEP Mode by any of the following operations:

- A RESET via the external $\overline{\text{RESET}}$ pin driven Low.
- A RESET via a real-time clock alarm.
- A RESET via a WDT time-out (if running out of the 32 kHz oscillator and configured to generate a RESET on time-out).
- A RESET via execution of a Debug RESET command.
- A RESET via the Low-Voltage Brown-Out (VBO) detection circuit, if enabled.

After exiting SLEEP Mode, the standard RESET delay occurs to allow the primary crystal oscillator to stabilize. For more information, see Figure 4 on page 40.

HALT Mode

Execution of the CPU's HALT instruction puts the eZ80F91 device into HALT Mode. In HALT Mode, the operating characteristics are:

- The primary crystal oscillator is enabled and continues to operate.

4. Set Px_DDR = 0.

When configured for single edge-triggered interrupt mode (GPIO Mode 9), the value in the Port *x* Data Register determines whether a positive or negative edge causes an interrupt request. 0 in the Port *x* Data Register bit sets the selected pin to generate an interrupt request for falling edges. 1 in the Port *x* Data Register bit sets the selected pin to generate an interrupt request for rising edges. To select Mode 9 from the default mode (Mode 2), observe the following brief procedure.

1. Set Px_DR = 1
2. Set Px_ALT2 = 1
3. Set Px_ALT = 1.
4. Set Px_DDR = 1.

Edge-triggered interrupts are cleared by writing 1 to the corresponding bit of the Px_ALT0 Register. For example, if PD4 has been set up to generate an edge-triggered interrupt, the interrupt is cleared by writing a 1 to Px_ALT0[4].

GPIO Control Registers

Each GPIO port has four registers that controls its operation. The operating mode of each bit within a port is selected by writing to the corresponding bits of these four registers as shown in Table 6 on page 46. These four registers are Port Data Register (Px_DR), Port Data Direction Register (Px_DDR), Port Alternate Register 1 (PX_ALT1), and Port Alternate Register 2 (Px_ALT2). In addition to these four control registers, each port has a Port Alternate Register 0 (Px_ALT0), which is used for clearing edge-triggered interrupts.

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) is driven on the corresponding pins. In all modes, reading from the Port *x* Data registers always returns the sampled current value of the corresponding pins. When the port pins are configured for edge-triggered interrupts or level-sensitive interrupts, the value written to the Port *x* Data Register bit selects the interrupt edge or interrupt level (for more details about GPIO mode selection, see Table 6 on page 46).

Bit	Description (Continued)
[3:0] BUS_CYCLE	**Bus Cycle** 0000: Not valid. 0001: Each bus mode state is 1 eZ80 clock cycle in duration.[1, 2, 3] 0010: Each bus mode state is 2 eZ80 clock cycles in duration. 0011: Each bus mode state is 3 eZ80 clock cycles in duration. 0100: Each bus mode state is 4 eZ80 clock cycles in duration. 0101: Each bus mode state is 5 eZ80 clock cycles in duration. 0110: Each bus mode state is 6 eZ80 clock cycles in duration. 0111: Each bus mode state is 7 eZ80 clock cycles in duration. 1000: Each bus mode state is 8 eZ80 clock cycles in duration. 1001: Each bus mode state is 9 eZ80 clock cycles in duration. 1010: Each bus mode state is 10 eZ80 clock cycles in duration. 1011: Each bus mode state is 11 eZ80 clock cycles in duration. 1100: Each bus mode state is 12 eZ80 clock cycles in duration. 1101: Each bus mode state is 13 eZ80 clock cycles in duration. 1110: Each bus mode state is 14 eZ80 clock cycles in duration. 1111: Each bus mode state is 15 eZ80 clock cycles in duration.

Notes:
1. Setting the BUS_CYCLE to 1 in Intel bus mode causes the ALE pin to not function properly.
2. Use of the external WAIT input pin in Z80 mode requires that BUS_CYCLE is set to a value greater than 1.
3. BUS_CYCLE produces no effect in eZ80 mode.

Bus Arbiter

The Bus Arbiter within the eZ80F91 allows external bus masters to gain control of the CPU memory interface bus. During normal operation, the eZ80F91 device is the bus master. External devices request master use of the bus by asserting the $\overline{\text{BUSREQ}}$ pin. The Bus Arbiter forces the CPU to release the bus after completing the current instruction. When the CPU releases the bus, the Bus Arbiter asserts the $\overline{\text{BUSACK}}$ pin to notify the external device that it can master the bus. When an external device assumes control of the memory interface bus, the bus acknowledge cycle is complete. Table 31 shows the status of the pins on the eZ80F91 device during bus acknowledge cycles.

During a bus acknowledge cycle, the bus interface pins of the eZ80F91 device are used by an external bus master to control the memory and I/O chip selects.

Table 31. eZ80F91 Pin Status During Bus Acknowledge Cycles

Pin Symbol	Signal Direction	Description
ADDR23..ADDR0	Input	Allows external bus master to utilize the chip select logic of the eZ80F91.
CS0	Output	Normal operation.

deactivated by a CPU read of the timer interrupt identification register, TMR*x*_IIR. All bits in that register are reset by the read.

The response of the CPU to this interrupt service request is a function of the CPU's interrupt enable flag, IEF1. For more information about this flag, refer to the eZ80 CPU User Manual (UM0077) available for free download from the Zilog website.

Timer Input Source Selection

Timers 0–3 features programmable input source selection. By default, the input is taken from the eZ80F91's system clock. The timers also use the Real-Time Clock source (50, 60, or 32768THz) as their clock sources. The input source for these timers is set using the timer control register. (TMR*x*_CTL[CLK_SEL])

Timer Output

The timer count is directed to the GPIO output pins, if required. To enable the Timer Output feature, the GPIO port pin must be configured as an output and for alternate functions. The GPIO output pin toggles each time the timer reaches its end-of-count value. In CONTINUOUS Mode operation, enabling the Timer Output feature results in a Timer Output signal period which is twice the timer time-out period. Examples of Timer Output operation are shown in Figure 29 and Table 52. The initial value for the timer output is zero.

Logic to support timer output exists in all timers; but for the eZ80F91 device, only Timer 0 and 2 route the actual timer output to the pins. Because Timer 3 uses the T_{OUT} pins for PWMxN signals, the timer outputs are not available when using complementary PWM outputs. See Table 52 for details.

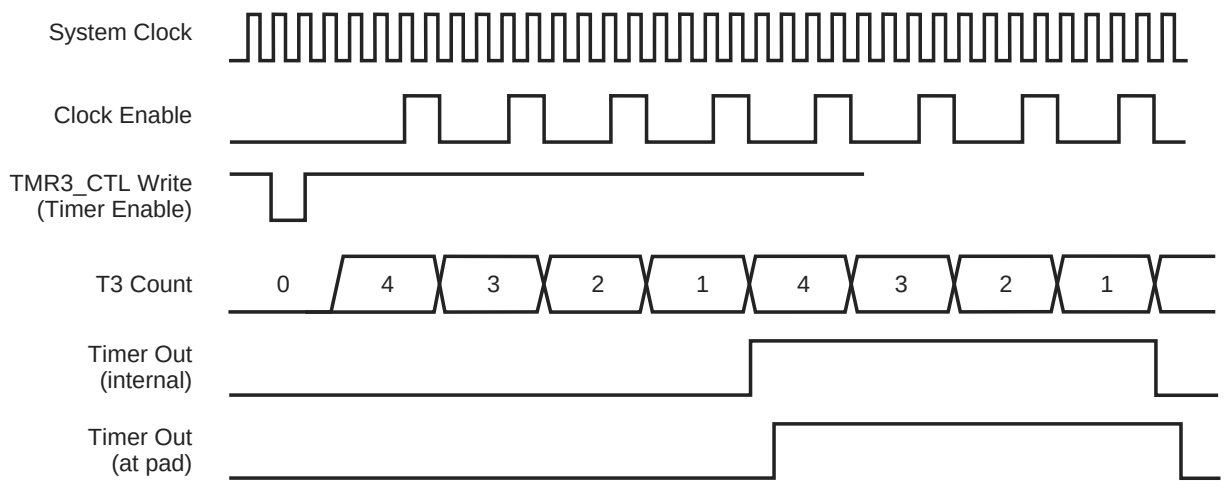

Figure 29. Example: PRT Timer Output Operation

Table 57. Timer Data Low Byte Register (TMR*x*_DR_L)

Bit	7	6	5	4	3	2	1	0
Field	TMR*x*_DR_L							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	TMR0_DR_L = 0063h, TMR1_DR_L = 0068h, TMR2_DR_L = 0072h, TMR3_DR_L = 0077h							

Note: R = read only.

Bit	Description
[7:0] TMR*x*_DR_L	**Timer Data Low Byte** 00h–FFh: These bits represent the low byte of the 2-byte timer data value, {TMR*x*_DR_H[7:0], TMR*x*_DR_L[7:0]}. Bit 7 is bit 7 of the 16-bit timer data value. Bit 0 is bit 0 (lsb) of the 16-bit timer data value.

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.

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. The value in the RTC_DOM Register is unchanged by a RESET. The current setting of BCD_EN determines whether the values in this register are binary (BCD_EN = 0) or binary-coded decimal (BCD_EN = 1). Access to this register is read-only if the RTC is locked, and read/write if the RTC is unlocked. See Table 84.

Table 84. Real-Time Clock Day-of-the-Month Register (RTC_DOM)

Bit	7	6	5	4	3	2	1	0
Field	TENS_DOM				DOM			
Reset	U	U	U	U	U	U	U	U
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Address	00E4h							

Note: U = unchanged by RESET; R/W* = read only if RTC locked, read/write if RTC unlocked.

Binary-Coded Decimal Operation (BCD_EN = 1)	
Bit	**Description**
[7:4] TENS_DOM	**Day Of The Month: Tens** 0–3: The tens digit of the current day-of-the-month count.
[3:0] DOM	**Day Of The Month: Ones** 0–9: The ones digit of the current day-of-the-month count.

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

Real-Time Clock Alarm Seconds Register

This register contains the alarm seconds value. The value in the RTC_ASEC Register is unchanged by a RESET. The current setting of BCD_EN determines whether the values in this register are binary (BCD_EN = 0) or binary-coded decimal (BCD_EN = 1). See Table 88.

Table 88. Real-Time Clock Alarm Seconds Register (RTC_ASEC)

Bit	7	6	5	4	3	2	1	0
Field	ATEN_SEC				ASEC			
Reset	U	U	U	U	U	U	U	U
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	00E8h							

Note: U = unchanged by RESET; R/W = read/write.

Binary-Coded Decimal Operation (BCD_EN = 1)

Bit	Description
[7:4] ATEN_SEC	**Alarm Seconds: Ten** 0–5: The tens digit of the alarm seconds value.
[3:0] ASEC	**Alarm Seconds: Ones** 0–9: The ones digit of the alarm seconds value.

Binary Operation (BCD_EN = 0)

Bit	Description
[7:0] ASEC	**Alarm Seconds** 00h–3Bh: The alarm seconds value.

- 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 case of 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 CPU reads the UARTx_MSR Register.

UART Recommended Usage

The following standard sequence of events occurs in the UART block of the eZ80F91 device. A description of each follows.

- Module Reset
- Control Transfers to Configure UART Operation
- 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 to Configure UART Operation

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 to ensure that none of the interrupt sources are active. The interrupts are enabled (except for the transmit interrupt) and the application is ready to use the module for transmission/reception.

Bit	Description (Continued)
[4] TCIE	**Transmission Complete Interrupt** 0: Transmission complete interrupt is disabled 1: Transmission complete interrupt is generated when both the transmit hold register and the transmit shift register are empty
[3] MIIE	**Modem Interrupt Input Enable** 0: Modem interrupt on edge detect of status inputs is disabled. 1: Modem interrupt on edge detect of status inputs is enabled.
[2] LSIE	**Line Status Interrupt Input Enable** 0: Line status interrupt is disabled. 1: Line status interrupt is enabled for receive data errors: incorrect parity bit received, framing error, overrun error, or break detection.
[1] TIE	**Transmit Interrupt Input Enable** 0: Transmit interrupt is disabled. 1: Transmit interrupt is enabled. Interrupt is generated when the transmit FIFO/buffer is empty indicating no more bytes available for transmission.
[0] RIE	**Receive Interrupt Input Enable** 0: Receive interrupt is disabled. 1: Receive interrupt and receiver time-out interrupt are enabled. Interrupt is generated if the FIFO/buffer contains data ready to be read or if the receiver times out.

UART Interrupt Identification Register

The read-only UARTx_IIR Register allows you to check whether the FIFO is enabled and the status of interrupts. These registers share the same I/O addresses as the UARTx_FCTL registers. See Tables 99 and 100.

Table 99. UART Interrupt Identification Registers (UART*x*_IIR)

Bit	7	6	5	4	3	2	1	0
Field	FSTS	Reserved			INSTS			INTBIT
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R
Address	UART0_IIR = 00C2h, UART1_IIR = 00D2h							

Note: *x* indicates UART[1:0]; R = read only.

Bit	Description
[7] FSTS	**FIFO Enable** 0: FIFO is disabled. 1: FIFO is enabled.
[6:4]	**Reserved** These bits are reserved and must be programmed to 000.

UART Modem Status Register

This register is used to show the status of the UART signals. See Table 107.

Table 107. UART Modem Status Registers (UART*x*_MSR)

Bit	7	6	5	4	3	2	1	0
Field	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
Reset	U	U	U	U	U	U	U	U
R/W	R	R	R	R	R	R	R	R
Address	UART0_MSR = 00C6h, UART1_MSR = 00D6h							

Note: *x* indicates UART[1:0]; U = undefined; R = read only.

Bit	Description
[7] DCD	**Data Carrier Detect** 0–1: In NORMAL Mode, this bit reflects the inverted state of the $\overline{\text{DCDx}}$ input pin. In LOOPBACK Mode, this bit reflects the value of the UARTx_MCTL[3] = out2.
[6] RI	**Ring Indicator** 0–1: In NORMAL Mode, this bit reflects the inverted state of the $\overline{\text{RIx}}$ input pin. In LOOPBACK Mode, this bit reflects the value of the UARTx_MCTL[2] = out1.
[5] DSR	**Data Set Ready** 0–1: In NORMAL Mode, this bit reflects the inverted state of the $\overline{\text{DSRx}}$ input pin. In LOOPBACK Mode, this bit reflects the value of the UARTx_MCTL[0] = DTR.
[4] CTS	**Clear To Send** 0–1: In NORMAL Mode, this bit reflects the inverted state of the $\overline{\text{CTSx}}$ input pin. In LOOPBACK Mode, this bit reflects the value of the UARTx_MCTL[1] = RTS.
[3] DDCD	**Delta Status Change of $\overline{\text{DCD}}$** 0–1: This bit is set to 1 whenever the $\overline{\text{DCDx}}$ pin changes state. This bit is reset to 0 when the UARTx_MSR Register is read.
[2] TERI	**Trailing Edge Change on $\overline{\text{RI}}$** 0–1: This bit is set to 1 whenever a falling edge is detected on the $\overline{\text{RIx}}$ pin. This bit is reset to 0 when the UARTx_MSR Register is read.
[1] DDSR	**Delta Status Change of $\overline{\text{DSR}}$** 0–1: This bit is set to 1 whenever the $\overline{\text{DSRx}}$ pin changes state. This bit is reset to 0 when the UARTx_MSR Register is read.
[0] DCTS	**Delta Status Change of $\overline{\text{CTS}}$** 0–1: This bit is set to 1 whenever the $\overline{\text{CTSx}}$ pin changes state. This bit is reset to 0 when the UARTx_MSR Register is read.

10 Mbps ENDEC Mode

This mode affects the MII interface between the PHY and the MAC. In ENDEC Mode, the RxCLK and TxCLK clocks are bit clocks instead of the normal nibble clock. In NIBBLE Mode, 4 bits are transferred on each clock. In ENDEC Mode, 1 bit is transferred per clock.

For more information about throughput, see the EMAC and the System Clock section on page 295.

Memory

EMAC memory is the shared Ethernet memory location of the Transmit and Receive buffers. This memory is broken into two parts: the Tx buffer and the Rx buffer. The Transmit Lower Boundary Pointer Register, EmacTLBP, is the register that holds the starting address of the Tx buffer. The Boundary Pointer Register, EmacBP, points to the start of the Rx buffer (end of Tx buffer + 1). The Receive High Boundary Pointer Register, EmacRHBP, points to the end of the Rx buffer + 1. The Tx and Receive buffers are divided into packet buffers of either 256, 128, 64, or 32 bytes. These buffer sizes are selected by EmacBufSize Register bits 7 and 6.

The EmacBlksLeft Register contains the number of Receive packet buffers remaining in the Rx buffer. This buffer is used for software flow control. If the Block_Level is nonzero (bits 5:0 of the EmacBufSize Register), hardware flow control is enabled. If in FULL-DUPLEX Mode, the EMAC transmits a pause control frame when the EmacBlksLeft Register is less than the Block_Level. In HALF-DUPLEX Mode, the EMAC continually transmits a nibble pattern of hexadecimal 5’s to jam the channel.

Four pointers are defined for reading and writing the Tx and Rx buffers. The Transmit Write Pointer, TWP, is a software pointer that points to the next available packet buffer. The TWP is reset to the value stored in EmacTLBP. The Transmit Read Pointer, TRP, is a hardware pointer in the Transmit Direct Memory Access Register, TxDMA, that contains the address of the next packet to be transmitted. It is automatically reset to the EmacTLBP. The Receive Write Pointer, RWP, is a hardware pointer in the Receive Direct Memory Access Register, RxDMA, which contains the storage address of the incoming packet. The RWP pointer is automatically initialized to the Boundary Pointer registers. The Receive Read Pointer, RRP, is a software pointer to the address location in which the next packet must be read from. The RRP pointer must be initialized to the Boundary Pointer registers. For the hardware flow control to function properly, the software must update the hardware RRP (EmacRrp) pointer whenever the software version is updated. The RxDMA uses RWP and the RRP to determine how many packet buffers remain in the Rx buffer.

Arbiter

The arbiter controls access to EMAC memory. It prioritizes the requests for memory access between the CPU, the TxDMA, and the RxDMA. The TxDMA offers two levels of priority: a high priority when the TxFIFO is less than half full and a Low priority when the

Similarly, for 100BaseT Ethernet, the data rate is 100 Mbps, which equates to 12.5 Mbps. If the eZ80F91 ASSP is operating in FULL-DUPLEX Mode over 100BaseT, the data rate for RX data and TX data is 12.5Mbps. 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 (12.5 + 12.5) x 2 = 50 MHz while transferring Ethernet packets to and from the physical layer. Consequently, 50 MHz is the minimum system clock speed that the eZ80 CPU requires to sustain EMAC data transfers while not including any software overhead or additional eZ80 tasks.

The FIFO functionality of the EMAC operates at any frequency as long as the user application avoids overrun and underrun errors via higher-level flow control. Actual application requirements will dictate Ethernet modes of operation (FULL-DUPLEX, HALF-DUPLEX, etc.). Because each user and application is different, it becomes your responsibility to control the data flow with these parameters. Under ideal conditions, the system clock will operate somewhere between 5 MHz and 50 MHz to handle the EMAC data rates.

EMAC Operation in HALT Modes

When the CPU is in HALT Mode, the eZ80F91 device's EMAC block cannot be disabled as other peripherals. Upon receipt of an Ethernet packet, a maskable Receive interrupt is generated by the EMAC block, just as it would be in a non-halt mode. Accordingly, the processor wakes up and continues with the user-defined application.

EMAC Registers

After a system reset, all EMAC registers are set to their default values. Any writes to unused registers or register bits are ignored and reads return a value of 0. For compatibility with future revisions, unused bits within a register must always be written with a value of 0. Read/write attributes, reset conditions, and bit descriptions of all of the EMAC registers are provided in this section.

EMAC Test Register

The EMAC Test Register, shown in Table 181, allows test functionality of the EMAC block. Available test modes are defined for bits [6:0].

EMAC Transmit Lower Boundary Pointer High and Low Byte Registers

The EMAC Transmit Lower Boundary Pointer is set to the start of the Transmit buffer in EMAC shared memory. See Tables 206 and 207.

Table 206. EMAC Transmit Lower Boundary Pointer Low Byte Register (EMAC_TLBP_L)

Bit	7	6	5	4	3	2	1	0
Field	EMAC_TLBP_L							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R
Address	0042h							

Note: R/W = read/write.

Bit	Description
[7:0] EMAC_TLBP_L	**Transmit Lower Boundary Pointer Low Byte** 00h–FFh: These bits represent the low byte of the two-byte Transmit Lower Boundary Pointer value, {EMAC_TLBP_H, EMAC_TLBP_L}. Bit 7 is bit 7 of the 16-bit value. Bit 0 is bit 0 (lsb) of the 16-bit value.

Table 207. EMAC Transmit Lower Boundary Pointer High Byte Register (EMAC_TLBP_H) *

Bit	7	6	5	4	3	2	1	0
Field	EMAC_TLBP_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	0043h							

Note: R/W = read/write.

Bit	Description
[7:0] EMAC_TLBP_H	**Transmit Lower Boundary Pointer High Byte** 00h–FFh: These bits represent the high byte of the two-byte Transmit Lower Boundary Pointer value, {EMAC_TLBP_H, EMAC_TLBP_L}. Bit 7 is bit 15 (msb) of the 16-bit value. Bits 7:5 default to 000 on reset; bit 0 is bit 8 of the 16-bit value.

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

Table 241. External I/O Read Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Min.**	**Max.**
T_1	PHI Clock Rise to ADDR Valid Delay	–	7.3
T_2	PHI Clock Rise to ADDR Hold Time	1.0	–
T_3	DATA Valid to PHI Clock Rise Setup Time	0.5	–
T_4	PHI Clock Rise to DATA Hold Time	0.0	–
T_5	PHI Clock Rise to $\overline{CSx}$ Assertion Delay	2.0	8.5
T_6	PHI Clock Rise to $\overline{CSx}$ Deassertion Delay	0.0	6.0
T_7	PHI Clock Rise to $\overline{IORQ}$ Assertion Delay	2.6	7.0
T_8	PHI Clock Rise to $\overline{IORQ}$ Deassertion Delay	1.0	6.3
T_9	PHI Clock Rise to $\overline{RD}$ Assertion Delay	2.7	7.0
T_{10}	PHI Clock Rise to $\overline{RD}$ Deassertion Delay	0.5	6.3

External I/O Write Timing

Figure 71 and Table 242 show the timing for external I/O writes. PHI clock rise/fall to signal transition timing is independent of the particular bus mode employed (eZ80, Z80, Intel, or Motorola).

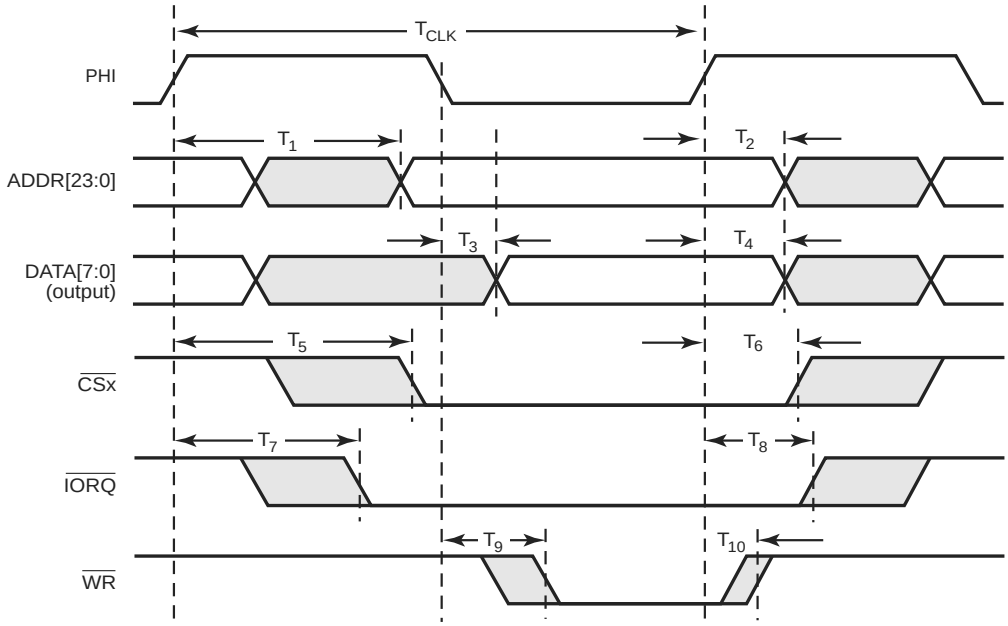

Figure 71. External I/O Write Timing