**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 | 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-LQFP |
| Supplier Device Package | 144-LQFP (20x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/ez80f91az050sg |

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

| LQFP Pin No | BGA Pin No | Symbol | Function | Signal Direction | Description |
|---|---|---|---|---|---|
| 95 | F10 | PC5 | 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. |
| | | DSR1 | Data Set Ready | Schmitt-trigger input, Active Low | Modem status signal to the UART. This signal is multiplexed with PC5. |
| 96 | G8 | PC6 | 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. |
| | | DCD1 | Data Carrier Detect | Schmitt-trigger input, Active Low | Modem status signal to the UART. This signal is multiplexed with PC6. |
| 97 | E12 | PC7 | 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. |
| | | RI1 | Ring Indicator | Schmitt-trigger input, Active Low | Modem status signal to the UART. This signal is multiplexed with PC7. |
| 98 | E11 | $V_{DD}$ | Power Supply | | Power Supply. |
| 99 | F9 | $V_{SS}$ | Ground | | Ground. |

# 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 Brownout (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 43.

## 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.
- The system clock is enabled and continues to operate.
- The CPU is idle.

### Chip Select x Control Register

The Chip Select x Control register (see Table 29) enables the chip selects, specifies the type of chip select, and sets the number of wait states. The reset state for the Chip Select 0 Control register is E8h when the reset state for three other Chip Select Control registers is 00h.

**Table 29. Chip Select x Control Register (CS0_CTL = 00AAh, CS1_CTL = 00ADh, CS2_CTL = 00B0h, CS3_CTL = 00B3h)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CS0_CTL Reset | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| CS1_CTL Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CS2_CTL Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CS3_CTL Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CPU Access | R/W | R/W | R/W | R/W | R/W | R | R | R |

**Note:** R/W = Read/Write; R = Read Only.

| Bit Position | Value | Description |
|---|---|---|
| [7:5] CSX_WAIT | 000 | 0 wait states are asserted when this chip select is active. |
| | 001 | 1 wait state is asserted when this chip select is active. |
| | 010 | 2 wait states are asserted when this chip select is active. |
| | 011 | 3 wait states are asserted when this chip select is active. |
| | 100 | 4 wait states are asserted when this chip select is active. |
| | 101 | 5 wait states are asserted when this chip select is active. |
| | 110 | 6 wait states are asserted when this chip select is active. |
| | 111 | 7 wait states are asserted when this chip select is active. |
| 4 CSX_IO | 0 | Chip select is configured as a memory chip select. |
| | 1 | Chip select is configured as an I/O chip select. |
| 3 CSX_EN | 0 | Chip select is disabled. |
| | 1 | Chip select is enabled. |
| [2:0] | 000 | Reserved. |

select address space, the memory chip select is not activated. On-chip RAM is not accessible to external devices during bus acknowledge cycles.

## RAM Control Registers

### RAM Control Register

Internal general-purpose RAM is disabled by clearing the GPRAM_EN bit. The default on RESET is for general-purpose RAM to be enabled. See Table 32.

**Table 32. RAM Control Register (RAM_CTL=00B4h)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| CPU Access | R/W | R/W | R | R | R | R | R | R |

**Note:** R/W = Read/Write; R = Read Only.

| Bit Position | Value | Description |
|---|---|---|
| 7<br>GPRAM_EN | 0 | On-chip general-purpose RAM is disabled. |
| | 1 | On-chip general-purpose RAM is enabled. |
| 6<br>ERAM_EN | 0 | On-chip EMAC RAM is disabled. |
| | 1 | On-chip EMAC RAM is enabled. |
| [5:0] | 000000 | Reserved. |

without first erasing it. Otherwise, the burden is on software to ensure that the 31 ms maximum cumulative programming time between erases is not exceeded for a row.

### Memory Write

A single-byte memory Write operation uses the address bus and data bus of the eZ80F91 device for programming a single data byte to Flash memory. While the CPU executes a Load instruction, the Flash controller asserts the internal WAIT signal to stall the CPU until the Write is complete. A single-byte Write takes between 66 µs and 85 µs to complete. Programming an entire row using memory Writes therefore takes no more than 21.8 ms. This duration of time does not include time required by the CPU to transfer data to the registers, which is a function of the instructions employed and the system clock frequency.

The memory Write function does not support multibyte row programming. Because memory Writes are self-timed, they are performed back-to-back without requiring polling or interrupts.

## Erasing Flash Memory

Erasing bytes in Flash memory returns them to a value of FFh. Both the MASS and PAGE ERASE operations are self-timed by the Flash controller, leaving the CPU free to execute other operations in parallel. The DONE status bit in the Flash Interrupt Control Register are polled by software or used as an interrupt source to signal completion of an Erase operation. If the CPU attempts to access Flash memory while an erase is in progress, the Flash controller forces a wait state until the Erase operation is completed.

### Mass Erase

Performing a MASS ERASE operation on Flash memory erases all bits contained in the main Flash memory array. The information page remains unaffected unless the FLASH_PAGE register bit 7(INFO_EN) is set. This self-timed operation takes approximately 200 ms to complete.

### Page Erase

The smallest erasable unit in Flash memory is a page. The pages to be erased, whether they are the 128 main Flash memory pages or the information page, are determined by the setting of the FLASH_PAGE register. This self-timed operation takes approximately 10 ms to complete.

### Flash Address Upper Byte Register

The FLASH_ADDR_U register defines the upper 6 bits of the Flash memory address space. Changing the value of FLASH_ADDR_U allows on-chip 256 KB Flash memory to be mapped to any location within the 16 MB linear address space of the eZ80F91 device. If on-chip Flash memory is enabled, the Flash address assumes priority over any external Chip Selects. The external Chip Select signals are not asserted if the corresponding Flash address is enabled. Internal Flash memory does not hold priority over internal SRAM. See Table 37.

**Table 37. Flash Address Upper Byte Register (FLASH_ADDR_U = 00F7h)**

| 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 | R |

**Note:** R/W = Read/Write; R = Read Only.

| Bit Position | Value | Description |
|---|---|---|
| [7:2] FLASH_ADDR_U | 00h–FCh | These bits define the upper byte of the Flash address. When on-chip Flash is enabled, the Flash address space begins at address {FLASH_ADDR_U, 00b, 0000h}. On-chip Flash has priority over all external Chip Selects. |
| [1:0] | 00 | Reserved (enforces alignment on a 256 KB boundary). |

### Real-Time Clock Century Register

This register contains the current century count. The value in the RTC_CEN 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 87.

**Table 87. Real-Time Clock Century Register (RTC_CEN = 00E7h)**

| 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_CEN | 0–9 | The tens digit of the current century count. |
| [3:0] CEN | 0–9 | The ones digit of the current century count. |

**Binary Operation (BCD_EN = 0)**

| Bit Position | Value | Description |
|---|---|---|
| [7:0] CEN | 00h–63h | The current century 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 = 00E8h)**

| 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_SEC | 0–5 | The tens digit of the alarm seconds value. |
| [3:0] ASEC | 0–9 | The ones digit of the alarm seconds value. |

| Binary Operation (BCD_EN = 0) | | |
|---|---|---|
| **Bit Position** | **Value** | **Description** |
| [7:0] ASEC | 00h–3Bh | The alarm seconds value. |

### Real-Time Clock Alarm Hours Register

This register contains the alarm hours value. The value in the RTC_AHRS 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 90.

**Table 90. Real-Time Clock Alarm Hours Register (RTC_AHRS = 00EAh)**

| 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_HRS | 0–2 | The tens digit of the alarm hours value. |
| [3:0] AHRS | 0–9 | The ones digit of the alarm hours value. |

| Binary Operation (BCD_EN = 0) | | |
|---|---|---|
| **Bit Position** | **Value** | **Description** |
| [7:0] AHRS | 00h–17h | The alarm hours value. |

completed transmission. When the application makes this determination, it writes the transmit data bytes to the UARTx_THR register. The number of bytes that the application writes depends on whether or not the FIFO is enabled. If the FIFO is enabled, the application writes 16 bytes at a time. If not, the application writes one byte at a time. As a result of the first Write, the interrupt is deactivated. The CPU then waits for the next interrupt. When the interrupt is raised by the UART module, the CPU repeats the same process until it exhausts all of the data for transmission.

To control and check the modem status, the application sets up the modem by writing to the UARTx_MCTL register and reading the UARTx_MCTL register before starting the process described above.

In RS485 multidrop mode, the first byte of the message is the station address and the rest of the message contains the data for that station. You must set the Even Parity Select (EPS bit 4) and Parity Enable (PEN bit 3) in the UARTx_LCTL before sending the station address. We recommend that in your UART initialization routine set up the UARTx_LCTL register for your data transfer format and set the Parity Enable (PEN bit 3) bit. Each time you want to send a new message you must perform these three steps:

1. Since the UART automatically clears the Even Parity Select (EPS bit 4) bit in the UARTx_LCTL after a byte is sent, before starting a new message you have to wait for the transmitter to go idle. The Transmit Empty (TEMT bit 6) of the UARTx_LSR will be set. If you set the EPS bit of the UARTx_LCTL before the last byte of the previous message is transmitted, the EPS bit will be cleared and the new station address will be sent as data instead of being used as an address.
2. Set the Even Parity Select (EPS bit 4) bit in the UARTx_LCTL register being careful not to alter the other bits in the register sets the address mark. Write station address to the UARTx_THR. The UART will automatically clear the EPS bit after the station address byte is transmitted.
3. Send the rest of the message. Write data to the UART Transmit Holding Register UARTx_THR whenever the Transmit Holding Register Empty (THRE bit 5) in the UARTx_LSR is set.

In multidrop mode, during receiving start address marks, you will see a receive line interrupt (INSTS bits[3:1]) in the IIR register. Read the LSR and check for receive errors only and ignore any parity errors. The parity is only used for address marks in this multidrop mode.

**Receive—**The receiver is always enabled, and it continually checks for the start bit on the RxD input signal. When an interrupt is raised by the UART module, the application reads the UARTx_IIR register and determines the cause for the interrupt. If the cause is a line status interrupt, the application reads the UARTx_LSR register, reads the data byte and then discards the byte or take other appropriate action. If the interrupt is caused by a receive-data-ready condition, the application alternately reads the UARTx_LSR and UARTx_RBR registers and removes all of the received data bytes. It reads the

divide-by-two clock divider. In MASTER mode, the SPI serial clock is one-half the frequency of the clock signal created by the SPI's Baud Rate Generator.

As displayed in Figure 42 and Table 111, four possible timing relations are chosen by using the clock polarity (CPOL) and clock phase CPHA control bits in the SPI Control register. See SPI Control Register on page 208. Both the master and slave must operate with the identical timing, CPOL, and CPHA. The master device always places data on the MOSI line a half-cycle before the clock edge (SCK signal), for the slave device to latch the data.

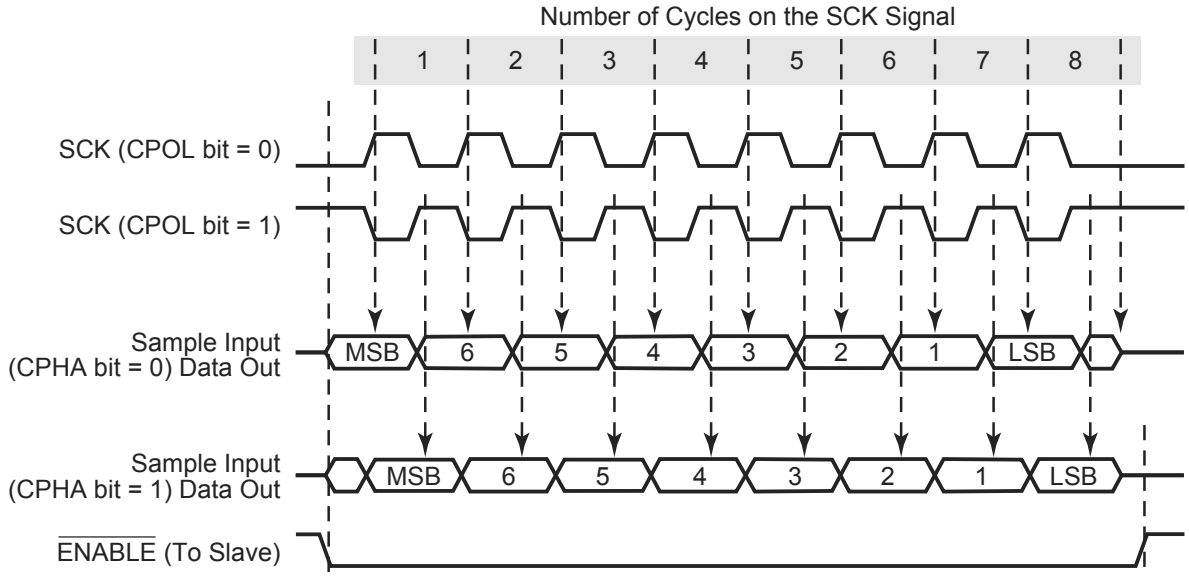


**Figure 42. SPI Timing**

**Table 111. SPI Clock Phase and Clock Polarity Operation**

| CPHA | CPOL | SCK Transmit Edge | SCK Receive Edge | SCK Idle State | $\overline{SS}$ High Between Characters? |
|---|---|---|---|---|---|
| 0 | 0 | Falling | Rising | Low | Yes |
| 0 | 1 | Rising | Falling | High | Yes |
| 1 | 0 | Rising | Falling | Low | No |
| 1 | 1 | Falling | Rising | High | No |

**Table 121. I²C Master Receive Status Codes**

| Code | I²C State | Microcontroller Response | Next I²C Action |
|---|---|---|---|
| 40h | Addr + R transmitted, ACK received | For a 7-bit address, clear IFLG, AAK = 0[1] | Receive data byte, transmit NACK |
| | | Or clear IFLG, AAK = 1 | Receive data byte, transmit ACK |
| | | For a 10-bit address Write extended address byte to data, clear IFLG | Transmit extended address byte |
| 48h | Addr + R transmitted, ACK not received[2] | For a 7-bit address: Set STA, clear IFLG | Transmit repeated START |
| | | Or set STP, clear IFLG | Transmit STOP |
| | | Or set STA & STP, clear IFLG | Transmit STOP then START |
| | | For a 10-bit address: Write extended address byte to data, clear IFLG | Transmit extended address byte |
| 38h | Arbitration lost | Clear IFLG | Return to idle |
| | | Or set STA, clear IFLG | Transmit START when bus is free |
| 68h | Arbitration lost, SLA+W received, ACK transmitted[3] | Clear IFLG, clear AAK = 0 | Receive data byte, transmit NACK |
| | | Or clear IFLG, set AAK = 1 | Receive data byte, transmit ACK |
| 78h | Arbitration lost, General call addr received, ACK transmitted | Same as code 68h | Same as code 68h |
| B0h | Arbitration lost, SLA+R received, ACK transmitted | Write byte to DATA, clear IFLG, clear AAK = 0 | Transmit last byte, receive ACK |
| | | Or write byte to DATA, clear IFLG, set AAK = 1 | Transmit data byte, receive ACK |

**Notes**
1. AAK is an I²C control bit that identifies which ACK signal to transmit.
2. R is defined as the Read bit; that is, the lsb is set to 1.
3. W is defined as the Write bit; that is, the lsb is cleared to 0.

If 10-bit addressing is being used, the slave is first addressed using the full 10-bit address, plus the Write bit. The master then issues a restart followed by the first part of the 10-bit

contains the two status codes 88h or 98h if SLAVE RECEIVE mode is entered with the general call address. The I2C returns to an idle state when the IFLG bit is cleared to 0.

## I2C Registers

The section that follows describes each of the eZ80F91 MCU's Inter-Integrated Circuit (I2C) registers.

### Addressing

The CPU interface provides access to seven 8-bit registers: four Read/Write registers, one Read Only register and two Write Only registers, as listed in Table 123.

**Table 123. I2C Register Descriptions**

| Register | Description |
|---|---|
| I2C_SAR | Slave address register |
| I2C_XSAR | Extended slave address register |
| I2C_DR | Data byte register |
| I2C_CTL | Control register |
| I2C_SR | Status register (Read Only) |
| I2C_CCR | Clock Control register (Write Only) |
| I2C_SRR | Software reset register (Write Only) |

### Resetting the I2C Registers

**Hardware Reset—**When the I2C is reset by a hardware reset of the eZ80F91 device, the I2C_SAR, I2C_XSAR, I2C_DR, and I2C_CTL registers are cleared to 00h; while the I2C_SR register is set to F8h.

**Software Reset—**Perform a software reset by writing any value to the I2C Software Reset Register (I2C_SRR). A software reset clears the STP, STA, and IFLG bits of the I2C_CTL register to 0 and sets the I2C back to an idle state.

### I2C Slave Address Register

The I2C_SAR register provides the 7-bit address of the I2C when in SLAVE mode and allows 10-bit addressing in conjunction with the I2C_XSAR register. I2C_SAR[7:1] = SLA[6:0] is the 7-bit address of the I2C when in 7-bit SLAVE mode. When the I2C receives this address after a START condition, it enters SLAVE mode. I2C_SAR[7] corresponds to the first bit received from the I2C bus.

When the register receives an address starting with F7h to F0h (I2C_SAR[7:3] = 11110b), the I2C recognizes that a 10-bit slave addressing mode is being selected. The I2C sends an ACK after receiving the I2C_SAR byte (the device does not generate an interrupt at this

**Table 133. ZDI Write Only Registers (Continued)**

| ZDI Address | ZDI Register Name | ZDI Register Function | Reset Value |
|---|---|---|---|
| 08h | ZDI_ADDR2_L | Address Match 2 Low Byte | XXh |
| 09h | ZDI_ADDR2_H | Address Match 2 High Byte | XXh |
| 0Ah | ZDI_ADDR2_U | Address Match 2 Upper Byte | XXh |
| 0Ch | ZDI_ADDR3_L | Address Match 3 Low Byte | XXh |
| 0Dh | ZDI_ADDR3_H | Address Match 3 High Byte | XXh |
| 0Eh | ZDI_ADDR3_U | Address Match 4 Upper Byte | XXh |
| 10h | ZDI_BRK_CTL | Break Control Register | 00h |
| 11h | ZDI_MASTER_CTL | Master Control Register | 00h |
| 13h | ZDI_WR_DATA_L | Write Data Low Byte | XXh |
| 14h | ZDI_WR_DATA_H | Write Data High Byte | XXh |
| 15h | ZDI_WR_DATA_U | Write Data Upper Byte | XXh |
| 16h | ZDI_RW_CTL | Read/Write Control Register | 00h |
| 17h | ZDI_BUS_CTL | Bus Control Register | 00h |
| 21h | ZDI_IS4 | Instruction Store 4 | XXh |
| 22h | ZDI_IS3 | Instruction Store 3 | XXh |
| 23h | ZDI_IS2 | Instruction Store 2 | XXh |
| 24h | ZDI_IS1 | Instruction Store 1 | XXh |
| 25h | ZDI_IS0 | Instruction Store 0 | XXh |
| 30h | ZDI_WR_MEM | Write Memory Register | XXh |

## ZDI Read Only Registers

Table 134 lists the ZDI Read Only registers. Many of the ZDI Read Only addresses are shared with ZDI Write Only registers.

**Table 134. ZDI Read Only Registers**

| ZDI Address | ZDI Register Name | ZDI Register Function | Reset Value |
|---|---|---|---|
| 00h | ZDI_ID_L | eZ80® Product ID Low Byte Register | 08h |
| 01h | ZDI_ID_H | eZ80 Product ID High Byte Register | 00h |
| 02h | ZDI_ID_REV | eZ80 Product ID Revision Register | XXh |

**Table 178. Transmit Descriptor Status**

| Bit | Name | Description |
|---|---|---|
| 15 | TxOwner | 0 = Host (eZ80®) owns, 1 = EMAC owns. |
| 14 | TxAbort | 1 = Packet aborted (not transmitted). |
| 13 | TxBPA | 1 = Back pressure applied. |
| 12 | TxHuge | 1 = Packet size is very large (Pkt_Size > EmacMaxf). |
| 11 | TxLOOR | 1 = Type/Length field is out of range (larger than 1518 bytes). |
| 10 | TxLCError | 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. |
| 9 | TxCrcError | 1 = The packet contains an invalid FCS (CRC). This flag is set when CRCEN = 0 and the last 4 bytes of the packet are not the valid FCS. |
| 8 | TxPktDeferred | 1 = Packet is deferred. |
| 7 | TxXsDfr | 1 = Packet is excessively deferred. (> 6071 nibble times in 100BaseT or 24,287 bit times in 10BaseT). |
| 6 | TxFifoUnderRun | 1 = TxFIFO experiences underrun. Check the TxAbort bit to see if the packet is aborted or retried. |
| 5 | TxLateCol | 1 = A late collision occurs. Collision is detected at a byte count > EmacCfg2[5:0]. Collisions detected before the byte count reaches EmacCfg2[5:0] are early collisions and retried. |
| 4 | TxMaxCol | 1 = The maximum number of collisions occurs. #Collisions > EmacCfg3[3:0]. These packets are aborted. |
| [3:0] | TxNumberOfCollisions | This field contains the number of collisions that occur while transmitting the packet. |

**Table 179. Receive Descriptor Status**

| Bit | Name | Description |
|---|---|---|
| 15 | RxOK | 1 = Packet received intact. |
| 14 | RxAlignError | 1 = An odd number of nibbles is received. |
| 13 | RxCrcError | 1 = The CRC (FCS) is in error. |

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 MCU 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.

Similarly, for 100BaseT Ethernet, the data rate is 100 Mbps, which equates to 12.5 Mbps. If the eZ80F91 MCU 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.

**Table 193. EMAC Non-Back-To-Back IPG Register—Part 2 (EMAC_IPGR2 = 002Fh)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reset | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| CPU Access | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

**Note:** R = Read Only; R/W = Read/Write.

| Bit Position | Value | Description |
|---|---|---|
| 7 | 0 | Reserved. |
| [6:0] IPGR2 | 00h–7Fh | This bit range is a programmable field representing the non-back-to-back interpacket gap. |

### EMAC Maximum Frame Length Register—Low and High Bytes

The 16-bit field resets to 0600h, which represents a maximum Receive frame of 1536 bytes. An untagged maximum size Ethernet frame (packet) is 1518 bytes. A tagged frame adds four bytes for a total of 1522 bytes. If a shorter maximum length restriction is more appropriate, program this field. See Table 194 and Table 195 on page 310.

**Note:** *The default value of 1536 bytes is large enough to cover the largest Ethernet packet, which contains 14 bytes of Ethernet header, 1500 bytes of MAC client data, plus 4 bytes of CRC for a total of 1518 maximum bytes. This value is also large enough to cover VLAN frames with prepended headers up to 18 bytes.*

*VLAN frames have a proprietary header prepended to the Ethernet packet. Setting the DCRCC bit in EMAC_CFG1 will exclude the first 4 bytes—the proprietary header—from the CRC calculation. For VLAN packets, the maximum frame length is 1522, 4 more than for normal Ethernet packets due to the 4 byte prepended header. Normal packets feature a 12 byte header before the MAC client data. For more information about this topic, refer to Figure 3-1 of the IEEE 802.3 specification.*

*If a proprietary header is allowed, this field must be adjusted accordingly. For example, if 12 byte headers are prepended to frames, MAXF must be set to 1524 bytes to allow the maximum VLAN tagged frame plus the 12 byte header. The default value of 1536 is large enough to cover the largest Ethernet packet: 14 bytes of Ethernet header, 1500 bytes of MAC client data, plus 4 bytes of CRC for a total of 1518 bytes maximum. It is also large enough to cover VLAN packets with prepended headers up to 18 bytes. The following formulas illustrate:*

**Ethernet Packet—** Maximum frame size = normal Ethernet packet – 14 (Ethernet header) + 1500 (MAC client data) + 4 (CRC) = 1518 bytes

**Table 213. EMAC Receive Read Pointer Register—High Byte (EMAC_RRP_H = 004Ah)**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CPU Access | R | R | R | R/W | R/W | R/W | R/W | R/W |

**Note:** R = Read Only, R/W = Read/Write.

| Bit Position | Value | Description |
|---|---|---|
| [7:0] EMAC_RRP_H | 00h–FFh | These bits represent the High byte of the 2-byte EMAC Receive Read Pointer value, {EMAC_RRP_H, EMAC_RRP_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. |

### EMAC Buffer Size Register

The lower six bits of this register set the level at which the EMAC either transmits a pause control frame or jams the Ethernet bus, depending on the mode selected. When each of these bits contain a zero, this feature is disabled.

In FULL-DUPLEX mode, a Pause Control Frame is transmitted as a One-shot operation. The software must free up a number of Rx buffers so that the number of buffers remaining, EmacBlksLeft, is greater than TCPF_LEV.

In HALF-DUPLEX mode, the EMAC jams the Ethernet by sending a continuous stream of hexadecimal 5s (`5fh`). When the software frees up the Rx buffers and the number of buffers remaining, EmacBlksLeft, is greater than TCPF_LEV, the EMAC stops jamming.

**Table 233. DC Characteristics**

| | | $T_A$ = 0 ºC to 70 ºC | | | $T_A$ = –40 ºC to 105 ºC | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Minimum** | **Typ[2]** | **Maximum** | **Minimum** | **Typ[2]** | **Maximum** | **Units** | **Conditions** |
| $V_{DD}$ | Supply Voltage | 3.0 | 3.3 | 3.6 | 3.0 | 3.3 | 3.6 | V | |
| $V_{IL}$ | Low Level Input Voltage | –0.3 | | 0.3 x $V_{DD}$ | –0.3 | | 0.3 x $V_{DD}$ | V | |
| $V_{IH}$ | High Level Input Voltage | 0.7x$V_{DD}$ | | 5.5 | 0.7x$V_{DD}$ | | 5.5 | V | |
| $V_{OL}$ | Low Level Output Voltage | | | 0.4 | | | 0.4 | V | $V_{DD}$ = 3.0 V; $I_{OL}$ = 1 mA |
| $V_{OH}$ | High Level Output Voltage | 2.4 | | | 2.4 | | | V | $V_{DD}$ = 3.0 V; $I_{OH}$ = –1 mA |
| $V_{RTC}$ | RTC Supply Voltage | 2.0 | | 3.6 | 2.0 | | 3.6 | V | |
| $I_{IL}$ | Input Leakage Current | –10 | | +10 | –10 | | +10 | µA | $V_{DD}$ = 3.6 V; $V_{IN}$ = $V_{DD}$ or $V_{SS}$[1] |
| $I_{TL}$ | Open-drain Leakage Current | –10 | | +10 | –10 | | +10 | µA | $V_{DD}$ = 3.6 V |
| $I_{CCa}$ | Active Current | | | | | 26 | 40 | mA | @ 10 MHz |
| | | | | | | 52 | 80 | mA | @ 20 MHz |
| | | | | | | 137 | 190 | mA | @ 50 MHz |
| $I_{CCh}$ | HALT Mode Current | | | | | 15 | 20 | mA | @ 10 MHz |
| | | | | | | 27 | 40 | mA | @ 20 MHz |
| | | | | | | 75 | 100 | mA | @ 50 MHz |
| $I_{CCs}$ | SLEEP Mode Current | | 2.5 | 20 | | 2.5 | 95 | µA | VBO_OFF=1 (VBO disabled) |
| $I_{RTC}$ | RTC Supply Current | | 2.5 | 10 | | 2.5 | 10 | µA | Supply current into $V_{RTC}$ |

[1]This condition excludes all pins with on-chip pull-ups when driven Low.
[2]Values in Typical column are for Vdd = 3.3 V and $T_A$ = 25 ºC.

## POR and VBO Electrical Characteristics

Table 234 on page 341 lists the Power-On Reset and Voltage Brownout characteristics of the eZ80F91 device.

Figure 67 displays the typical current consumption of the eZ80F91 device versus Vdd while operating in SLEEP mode (units in microamps, $10^{-6}$A); all peripherals off, and VBO disabled.

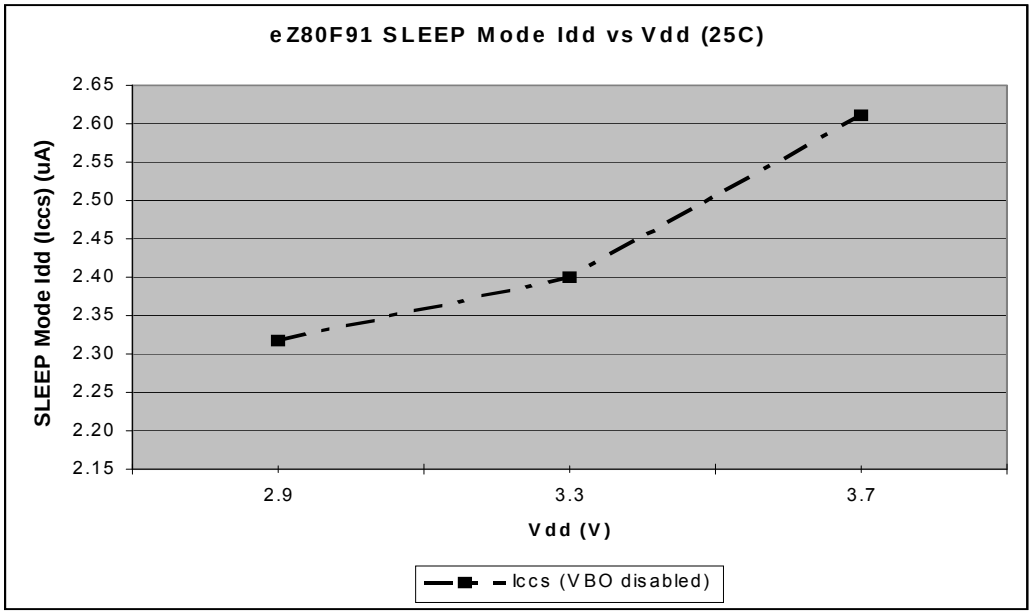


**Figure 67. $I_{CC}$ vs. Vdd During SLEEP Mode**

## AC Characteristics

This section provides information about the AC characteristics and timing of the eZ80F91 device. All AC timing information assumes a standard load of 50 pF on all outputs. See Table 236.

**Table 236. AC Characteristics**

| | | $T_A$ = 0 ºC to 70 ºC | | $T_A$ = –40 ºC to 105 ºC | | | |
|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Minimum** | **Maximum** | **Minimum** | **Maximum** | **Units** | **Conditions** |
| $T_{XIN}$ | System Clock Cycle Time | 20 | 1000 | 20 | 1000 | ns | $V_{DD}$ = 3.0–3.6 V |
| $T_{XINH}$ | System Clock High Time | 8 | | 8 | | ns | $V_{DD}$ = 3.0–3.6 V; $T_{CLK}$ = 20 ns |
| $T_{XINL}$ | System Clock Low Time | 8 | | 8 | | ns | $V_{DD}$ = 3.0–3.6 V; $T_{CLK}$ = 20 ns |