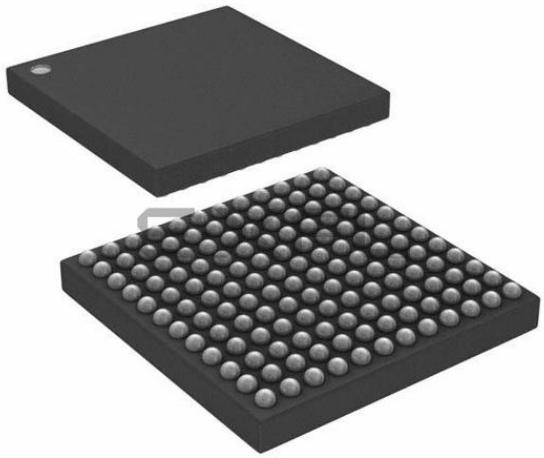

Welcome to E-XFL.COM

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/ez80f91na050sc00tr

Date	Revision Level	Section	Description	Page Number
June 2006	10	Global modifications	Updated for new release.	All
		Pin Identification on the eZ80F91 Device	Table 3: The description of the following pins modified: pins 55, 61, 63 and 69	6
		General-Purpose Input/ Output	GPIO chapter totally rewritten	49
		Chip Selects and Wait States	Input/Output chip select operation modified	65
		Flash Memory	The following sections are modified in Flash memory chapter: Erasing Flash memory, Information page characteristics, Flash Write/Erase protection register, Flash program control registers, and Table 43.	97
		Real-Time Clock Overview	Added a note in real time clock overview section	159
		Universal Asynchronous Receiver/Transmitter	Table 102 and 109 modified	175
		Infrared Encoder/Decoder Control Registers	The field [7:4] modified in Table 111	199
		Zilog Debug Interface	Updated the Introduction section, Added two paragraphs to ZDI Read Memory Registers	231
		On-Chip Oscillators	On page 349, Figure 63: Recommended Crystal Oscillator Configuration, the value of inductance L is changed to 3.3 μH. On page 351, Table 232, changed serial resistance value from 40 kΩ to 50 kΩ	336
		POR and VBO Electrical Characteristics	In Table 235: Min, Typ, and Max values of VBO voltage threshold modified and added IS_{por_vbo} parameter	341
		Ordering Information	Ordering information modified	359

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

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
61	M8	RTC_X_{IN}	Real-Time Clock Crystal Input	Input	This pin is the input to the low-power 32 kHz crystal oscillator for the Real-time clock. If the Real-time clock is disabled or not used, this input must be left floating or tied to VSS to minimize any input current leakage.
62	L8	RTC_X_{OUT}	Real-Time Clock Crystal Output	Bidirectional	This pin is the output from the low-power 32 kHz crystal oscillator for the Real-Time Clock. This pin is an input when the RTC is configured to operate from 50/60 Hz input clock signals and the 32 kHz crystal oscillator is disabled.
63	J7	RTC_V_{DD}	Real-Time Clock Power Supply		Power supply for the Real-Time Clock and associated 32 kHz oscillator. Isolated from the power supply to the remainder of the chip. A battery is connected to this pin to supply constant power to the Real-Time Clock and 32 kHz oscillator. If the Real-time clock is disabled or not used this output must be tied to Vdd.
64	K8	V_{SS}	Ground		Ground.
65	M9	HALT_SLP	HALT and SLEEP Indicator	Output, Active Low	A Low on this pin indicates that the CPU has entered either HALT or SLEEP mode because of execution of either a HALT or SLP instruction.
66	H7	TMS	JTAG Test Mode Select	Input	JTAG Mode Select Input.
67	L9	TCK	JTAG Test Clock	Input	JTAG and ZDI clock input.
68	J8	TRIGOUT	JTAG Test Trigger Output	Output	Active High trigger event indicator.
69	K9	TDI	JTAG Test Data In	Bidirectional	JTAG data input pin. Functions as ZDI data I/O pin when JTAG is disabled. This pin has an internal pull-up resistor in the pad.

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

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
92	G9	PC2	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.
		RTS1	Request to Send	Output, Active Low	Modem control signal from UART. This signal is multiplexed with PC2.
93	F12	PC3	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.
		CTS1	Clear to Send	Schmitt-trigger input, Active Low	Modem status signal to the UART. This signal is multiplexed with PC3.
94	F11	PC4	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.
		DTR1	Data Terminal Ready	Output, Active Low	Modem control signal to the UART. This signal is multiplexed with PC4.

Table 5. Clock Peripheral Power-Down Register 2 (CLK_PPD2 = 00DCh)

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	R	R/W	R/W	R/W	R/W

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

Bit Position	Value	Description
7 PHI_OFF	1	PHI Clock output is disabled (output is high-impedance).
	0	PHI Clock output is enabled.
6 VBO_OFF	1	Voltage Brownout detection circuit is disabled. This reduces DC current consumption in situations where VBO detection is not necessary. Power-On Reset functionality is not affected by this setting.
	0	VBO detection circuit is enabled.
[5:4]	000	Reserved.
3 TIMER3_OFF	1	System clock to TIMER3 is powered down.
	0	System clock to TIMER3 is powered up.
2 TIMER2_OFF	1	System clock to TIMER2 is powered down.
	0	System clock to TIMER2 is powered up.
1 TIMER1_OFF	1	System clock to TIMER1 is powered down.
	0	System clock to TIMER1 is powered up.
0 TIMER0_OFF	1	System clock to TIMER0 is powered down.
	0	System clock to TIMER0 is powered up.

Table 21. Intel Bus Mode Read States—Separate Address and Data Buses (Continued)

STATE T3	During State T3, no bus signals are altered. If the external READY ($\overline{WAIT}$) pin is driven Low at least one CPU system clock cycle prior to the beginning of State T3, additional wait states (T_{WAIT}) are asserted until the READY pin is driven High.
STATE T4	The CPU latches the Read data at the beginning of State T4. The CPU deasserts the $\overline{RD}$ signal and completes the Intel bus mode cycle.

During Write operations with separate address and data buses, the Intel bus mode employs four states—T1, T2, T3, and T4 as listed in Table 22.

Table 22. Intel Bus Mode Write States—Separate Address and Data Buses

STATE T1	The Write cycle begins in State T1. The CPU drives the address onto the address bus, the associated chip select signal is asserted, and the data is driven onto the data bus. The CPU drives the ALE signal High at the beginning of T1. During the middle of T1, the CPU drives ALE Low to facilitate the latching of the address.
STATE T2	During State T2, the CPU asserts the $\overline{WR}$ signal. Depending on the instruction, either the $\overline{MREQ}$ or $\overline{IORQ}$ signal is asserted.
STATE T3	During State T3, no bus signals are altered. If the external READY ($\overline{WAIT}$) pin is driven Low at least one CPU system clock cycle prior to the beginning of State T3, additional wait states (T_{WAIT}) are asserted until the READY pin is driven High.
STATE T4	The CPU deasserts the $\overline{WR}$ signal at the beginning of State T4. The CPU holds the data and address buses till the end of T4. The bus cycle is completed at the end of T4.

Intel bus mode timing is displayed for a Read operation in Figure 13 on page 76 and for a Write operation in Figure 14 on page 77. If the READY signal (external $\overline{WAIT}$ pin) is driven Low prior to the beginning of State T3, additional wait states (T_{WAIT}) are asserted until the READY signal is driven High. The Intel bus mode states are configured for 2 to 15 CPU system clock cycles. In the Figure 13 on page 76 and Figure 14 on page 77, each Intel bus mode state is 2 CPU system clock cycles in duration. Figure 13 on page 76 and Figure 14 on page 77 also display the assertion of one Wait state (T_{WAIT}) by the selected peripheral.

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.

Information Page Characteristics

As noted earlier, the information page is not accessible using memory access instructions and must be accessed via the FLASH_DATA I/O register. The Flash Page Select Register contains a bit which selects the information page for I/O access.

There are two ways to erase the information page. You must set the FLASH_PAGE register(0x00FC) bit7(INFO_EN) and then you execute either a MASS ERASE (which also erases the entire main Flash memory array) operation or a PAGE ERASE operation.

Flash Control Registers

The Flash Control Register interface contains all the registers used in Flash memory. The definitions in this section describe each register.

Flash Key Register

Writing the two-byte sequence B6h, 49h in immediate succession to this register unlocks the Flash Divider and Flash Write/Erase Protection registers. If these values are not written by consecutive CPU I/O Writes (I/O reads and memory Read/Writes have no effect), the Flash Divider and Flash Write/Erase Protection registers remain locked. This prevents accidental overwrites of these critical Flash control register settings. Writing a value to either the Flash Frequency Divider Register or the Flash Write/Erase Protection Register automatically relocks both of the registers. See Table 35.

Table 35. Flash Key Register (FLASH_KEY = 00F5h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	W	W	W	W	W	W	W	W

Note: W = Write Only.

Bit Position	Value	Description
[7:0] FLASH_KEY	B6h, 49h	Sequential Write operations of the values B6h, 49h to this register will unlock the Flash Frequency Divider and Flash Write/Erase Protection registers.

TMRx_RR_H and TMRx_RR_L. Downcounting continues on the next clock edge and the timer continues to count until disabled. An example of the timer operating in CONTINUOUS mode is displayed in Figure 28. Timer register information is listed in Table 51.

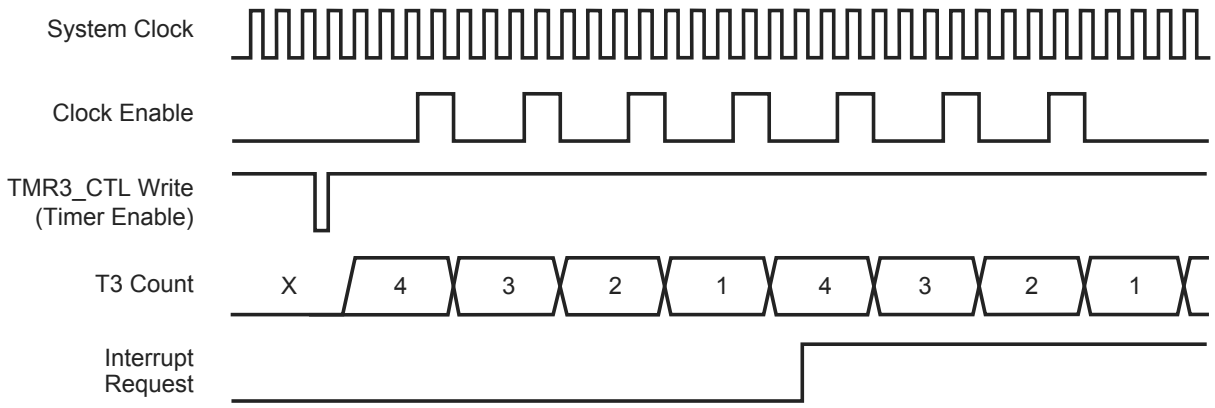

Figure 28. Example: PRT CONTINUOUS Mode Operation

Table 51. Example: PRT CONTINUOUS Mode Parameters

Parameter	Control Register(s)	Value
Timer Enable	TMR*x*_CTL[TIM_EN]	1
Reload	TMR*x*_CTL[RLD]	1
Prescaler Divider = 4	TMR*x*_CTL[CLK_DIV]	00b
CONTINUOUS Mode	TMR*x*_CTL[TIM_CONT]	1
End of Count Interrupt Enable	TMR*x*_IER[IRQ_EOC_EN]	1
Timer Reload Value	{TMR*x*_RR_H, TMR*x*_RR_L}	0004h

Timer Interrupts

The terminal count flag (TMR*x*_IIR[EOC]) is set to 1 whenever the timer reaches `0000h`, its end-of-count value in SINGLE PASS mode, or when the timer reloads the start value in CONTINUOUS mode. The terminal count flag is only set when the timer reaches `0000h` (or reloads) from `0001h`. The timer interrupt flag is not set to 1 when the timer is loaded with the value `0000h`, which selects the maximum time-out period.

The CPU is programmed to poll the EOC bit for the time-out event. Alternatively, an interrupt service request signal is sent to the CPU by setting the TMR*x*_IER[EOC] bit to 1.

And when the end-of-count value (0000h) is reached, the EOC bit is set to 1 and an interrupt service request signal is passed to the CPU. The interrupt service request signal is 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 on www.zilog.com.

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 32768 Hz) 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 is displayed in Figure 29 on page 126 and listed in Table 52 on page 126. 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 on page 126 for details.

Timer Data Register—Low Byte

The Timer x Data Register—Low Byte returns the Low byte of the current count value of the selected timer. The Timer Data Register—Low Byte (see Table 57) 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 Register—Low Byte, followed by the Timer Data Register—High Byte. The Timer Data Register—High Byte value is latched into temporary storage when a Read of the Timer Data Register—Low Byte occurs.

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

Table 57. Timer Data Register—Low Byte (TMR0_DR_L = 0063h, TMR1_DR_L = 0068h, TMR2_DR_L = 0072h, TMR3_DR_L = 0077h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	R	R	R	R	R	R	R	R

Note: R = Read only.

Bit Position	Value	Description
[7:0] TMR_DR_L	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.

Multi-PWM Power-Trip Mode

When enabled, the Multi-PWM power-trip feature forces the enabled PWM outputs to a predetermined state when an interrupt is generated from an external source via IC0, IC1, IC2, or IC3. One or multiple external interrupt sources are enabled at any given time. If multiple sources are enabled, any of the selected external sources trigger an interrupt. Configuring the PWM_CTL3 register enables or disables interrupt sources. See Table 75 on page 156.

The possible interrupt sources for a Multi-PWM power-trip are:

- IC0—digital input
- IC1—digital input
- IC2—digital input
- IC3—digital input

When the power-trip is detected, TMR3_PWM_CTL3[PTD] is set to 1 to indicate detection of the power-trip. A value of 0 signifies that no power-trip is detected.

The PWMs are released only after a power-trip when TMR3_PWM_CTL3[PTD] is written back to 0 by software. As a result, you are allowed to check the conditions of the motor being controlled before releasing the PWMs. The explicit release also prevents noise glitches after a power-trip from causing an accidental exit or re-entry of the PWM power-trip state.

The programmable power-trip states of the PWMs are globally grouped for the PWM outputs and the inverting $\overline{\text{PWM}}$ outputs. Upon detection of a power-trip, the PWM outputs are forced to either a High state, a Low state, or high-impedance. The settings for the power-trip states are made with power-trip control bits TMR3_PWM_CTL3[PT_LVL], TMR3_PWM_CTL3[PT_LVL_N], and TMR3_PWM_CTL3[PT_TRI].

Multi-PWM Control Registers

Pulse-Width Modulation Control Register 1

The PWM Control Register 1 (see Table 73) controls PWM function enables.

Table 73. PWM Control Register 1 (PWM_CTL1 = 0079h)

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

Note: R/W = Read/Write.

Bit Position	Value	Description
7 PAIR_EN	0	Global disable of the $\overline{\text{PWM}}$ outputs (PWM outputs enabled only).
	1	Global enable of the PWM and $\overline{\text{PWM}}$ output pairs.
6 PT_EN	0	Disable power-trip feature.
	1	Enable power-trip feature.
5 MM_EN	0	Disable Master mode.
	1	Enable Master mode.
4 pwm3_en	0	Disable PWM generator 3.
	1	Enable PWM generator 3.
3 pwm2_en	0	Disable PWM generator 2.
	1	Enable PWM generator 2.
2 pwm1_en	0	Disable PWM generator 1.
	1	Enable PWM generator 1.
1 PWM0_EN	0	Disable PWM generator 0.
	1	Enable PWM generator 0.
0 mpwm_en	0	Disable Multi-PWM mode.
	1	Enable Multi-PWM mode.

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.

Universal Asynchronous Receiver/Transmitter

The UART module implements all of the logic required to support the asynchronous communications protocol. The module also implements two separate 16-byte-deep FIFOs for both transmission and reception. A block diagram of the UART is displayed in Figure 36.

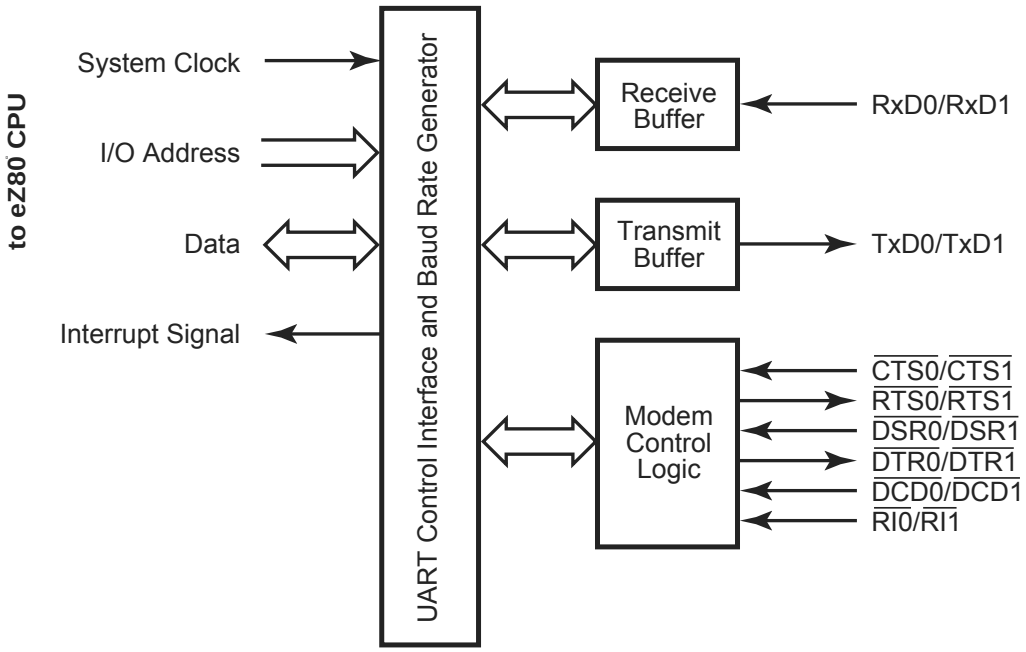

Figure 36. UART Block Diagram

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

- 5-, 6-, 7-, 8- or 9-bit data transmission.
- Even/odd, space/mark, address/data, or no parity bit generation and detection.
- Start and stop bit generation and detection (supports up to two stop bits).
- Line break detection and generation.
- Receiver overrun and framing errors detection.
- Logic and associated I/O to provide modem handshake capability.

UART FIFO Control Register

This register is used to monitor trigger levels, clear FIFO pointers, and enable or disable the FIFO. The UARTx_FCTL registers share the same I/O addresses as the UARTx_IIR registers. See Table 101.

Table 101. UART FIFO Control Registers (UART0_FCTL = 00C2h, UART1_FCTL = 00D2h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
CPU Access	W	W	W	W	W	W	W	W

Note: W = Write Only.

Bit Position	Value	Description
[7:6] TRIG	00	Receive FIFO trigger level set to 1. Receive data interrupt is generated when there is 1 byte in the FIFO. Valid only if FIFO is enabled.
	01	Receive FIFO trigger level set to 4. Receive data interrupt is generated when there are 4bytes in the FIFO. Valid only if FIFO is enabled.
	10	Receive FIFO trigger level set to 8. Receive data interrupt is generated when there are 8 bytes in the FIFO. Valid only if FIFO is enabled.
	11	Receive FIFO trigger level set to 14. Receive data interrupt is generated when there are 14 bytes in the FIFO. Valid only if FIFO is enabled.
[5:3]	000b	Reserved—must be 000b.
2 CLRTxF	0	Transmit Disable. This register bit works differently than the standard 16550 UART. This bit must be set to transmit data. When it is reset the transmit FIFO logic is reset along with the associated transmit logic to keep them in sync. This bit is now persistent–it does not self clear and it must remain at 1 to transmit data.
	1	Transmit Enable.
1 CLRRxF	0	Receive Disable. This register bit works differently than the standard 16550 UART. This bit must be set to receive data. When it is reset the receive FIFO logic is reset along with the associated receive logic to keep them in sync and avoid the previous version's lookup problem. This bit is now persistent–it does not self clear and it must remain at 1 to receive data.
	1	Receive Enable.

Data Validity

The data on the SDA line must be stable during the High period of the clock. The High or Low state of the data line changes only when the clock signal on the SCL line is Low, as displayed in Figure 43.

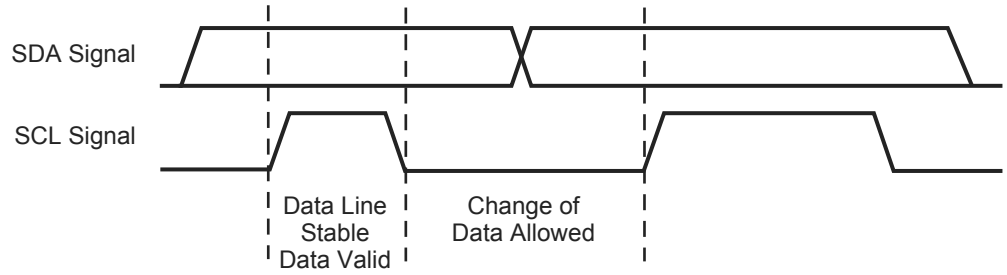

Figure 43. I²C Clock and Data Relationship

START and STOP Conditions

Within the I²C bus protocol, unique situations arise which are defined as START and STOP conditions. Figure 44 displays a High-to-Low transition on the SDA line while SCL is High, indicating a START condition. A Low-to-High transition on the SDA line while SCL is High defines a STOP condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after a START condition. The bus is considered to be free for a defined time after a STOP condition.

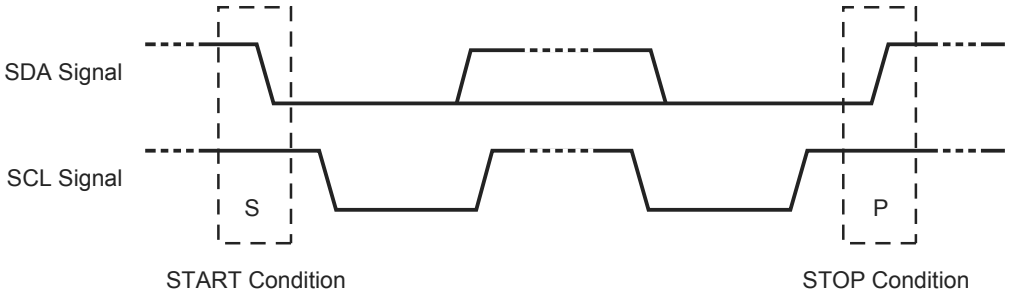

Figure 44. START and STOP Conditions In I²C Protocol

If the Master Mode Stop bit (STP) is set to 1 in MASTER mode, a STOP condition is transmitted on the I^2C bus. If the STP bit is set to 1 in SLAVE mode, the I^2C module operates as if a STOP condition is received, but no STOP condition is transmitted. If both STA and STP bits are set, the I^2C block first transmits the STOP condition (if in MASTER mode), then transmits the START condition. The STP bit is cleared to 0 automatically. Writing a 0 to this bit produces no effect.

The I^2C Interrupt Flag (IFLG) is set to 1 automatically when any of 30 of the possible 31 I^2C states is entered. The only state that does not set the IFLG bit is state F8h. If IFLG is set to 1 and the IEN bit is also set, an interrupt is generated. When IFLG is set by the I^2C, the Low period of the I^2C bus clock line is stretched and the data transfer is suspended. When a 0 is written to IFLG, the interrupt is cleared and the I^2C clock line is released.

When the I^2C Acknowledge bit (AAK) is set to 1, an acknowledge is sent during the acknowledge clock pulse on the I^2C bus if:

- Either the whole of a 7-bit slave address or the first or second byte of a 10-bit slave address is received.
- The general call address is received and the General Call Enable bit in I^2C_SAR is set to 1.
- A data byte is received while in MASTER or SLAVE modes.

When AAK is cleared to 0, a NACK is sent when a data byte is received in MASTER or SLAVE mode. If AAK is cleared to 0 in SLAVE TRANSMIT mode, the byte in the I^2C_DR register is assumed to be the final byte. After this byte is transmitted, the I^2C block enters the C8h state, then returns to an idle state. The I^2C module does not respond to its slave address unless AAK is set to 1. See Table 127 on page 226.

Table 127. I^2C Control Register (I2C_CTL = 00CBh)

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 IEN	0	I^2C interrupt is disabled.
	1	I^2C interrupt is enabled.
6 ENAB	0	The I^2C bus (SCL/SDA) is disabled and all inputs are ignored.
	1	The I^2C bus (SCL/SDA) is enabled.

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.

Bit Position	Value	Description
[2:0] CLKS	Programmable divisor that produces MDC from SCLK. MDC is the management data clock pin, which clocks MDIO data to and from the PHY. Its frequency is SCLK divided by the MDC clock divider.	
	000	MDC = SCLK ÷ 4.
	001	MDC = SCLK ÷ 4.
	010	MDC = SCLK ÷ 6.
	011	MDC = SCLK ÷ 8.
	100	MDC = SCLK ÷ 10.
	101	MDC = SCLK ÷ 14.
	110	MDC = SCLK ÷ 20.
	111	MDC = SCLK ÷ 28.

EMAC PHY Configuration Data Register—Low and High Byte

The Low and High bytes of the EMAC PHY Configuration Data Register represents the configuration data written to the external PHY. The EMAC_CTLD_H and EMAC_CTLD_L registers form a 16-bit register. These registers are loaded with data to be sent via the MDIO pin to the PHY. The PHY is selected by setting the EMAC_FIAD. The register inside the PHY is selected by setting EMAC_RGAD. See Table 199 and Table 200 on page 315.

Table 199. EMAC PHY Configuration Data Register—Low Byte (EMAC_CTLD_L = 003Ch)

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

Note: R/W = Read/Write.

Bit Position	Value	Description
[7:0] EMAC_CTLD_L	00h–FFh	These bits represent the Low byte of the 2 byte PHY configuration data value, {EMAC_CTLD_H, EMAC_CTLD_L}. Bit 7 is bit 7 of the 16 bit value. Bit 0 is bit 0 (lsb) of the 16 bit value.

J

L

M