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/ez80f91aza50sg

Pulse-Width Modulation Rising Edge High Byte Register 153
Pulse-Width Modulation Falling Edge Low Byte Register 154
Pulse-Width Modulation Falling Edge High Byte Register 154
Real-Time Clock 155
Real-Time Clock Alarm 156
Real-Time Clock Oscillator and Source Selection 156
Real-Time Clock Battery Backup 156
Real-Time Clock Recommended Operation 156
Real-Time Clock Registers 157
Real-Time Clock Seconds Register 157
Real-Time Clock Minutes Register 158
Real-Time Clock Hours Register 159
Real-Time Clock Day-of-the-Week Register 160
Real-Time Clock Day-of-the-Month Register 161
Real-Time Clock Month Register 162
Real-Time Clock Year Register 163
Real-Time Clock Century Register 164
Real-Time Clock Alarm Seconds Register 165
Real-Time Clock Alarm Minutes Register 166
Real-Time Clock Alarm Hours Register 167
Real-Time Clock Alarm Day-of-the-Week Register 168
Real-Time Clock Alarm Control Register 169
Real-Time Clock Control Register 170
Universal Asynchronous Receiver/Transmitter 172
UART Functional Description 173
UART Functions 173
UART Transmitter 173
UART Receiver 174
UART Modem Control 174
UART Interrupts 175
UART Transmitter Interrupt 175
UART Receiver Interrupts 175
UART Modem Status Interrupt 176
UART Recommended Usage 176
Module Reset 176
Control Transfers to Configure UART Operation 176
Data Transfers 177
Baud Rate Generator 178
Recommended Use of the Baud Rate Generator 179

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

Pin Description

Table 1 lists the pin configuration of the eZ80F91 ASSP device in the 144-BGA package.

Table 1. eZ80F91 144-BGA Pin Configuration

	12	11	10	9	8	7	6	5	4	3	2	1
A	SDA	SCL	PA0	PA4	PA7	COL	TxD0	V_{DD}	Rx_DV	MDC	WPn	A0
B	V_{SS}	PHI	PA1	PA3	V_{DD}	TxD3	Tx_EN	V_{SS}	RxD1	MDIO	A2	A1
C	PB6	PB7	V_{DD}	PA5	V_{SS}	TxD2	Tx_CLK	Rx_ CLK	RxD3	A3	V_{SS}	V_{DD}
D	PB1	PB3	PB5	V_{SS}	CRS	TxD1	Rx_ER	RxD2	A4	A8	A6	A7
E	PC7	V_{DD}	PB0	PB4	PA2	Tx_ER	RxD0	A5	A11	V_{SS}	V_{DD}	A10
F	PC3	PC4	PC5	V_{SS}	PB2	PA6	A9	A17	A15	A14	A13	A12
G	V_{SS}	PC0	PC1	PC2	PC6	PLL_ V_{SS}	V_{SS}	A23	A20	V_{SS}	V_{DD}	A16
H	X_{OUT}	X_{IN}	PLL_ V_{DD}	V_{DD}	PD7	TMS	V_{SS}	D5	V_{SS}	A21	A19	A18
J	V_{SS}	V_{DD}	LOOP FILT_ OUT	PD4	TRIGOUT	RTC_ V_{DD}	NMIn	WRn	D2	CS0n	V_{DD}	A22
K	PD5	PD6	PD3	TDI	V_{SS}	V_{DD}	RESETn	RDn	V_{DD}	D1	CS2n	CS1n
L	PD1	PD2	TRST n	TCK	RTC_ X_{OUT}	BUSACKn	WAITn	Marten	D6	D4	D0	CS3n
M	PD0	V_{SS}	TDO	HALT _ SLPn	RTC_ X_{IN}	BUSREQn	INSTRDn	IORQn	D7	D3	V_{SS}	V_{DD}

Note: Lowercase n suffix indicates an active-low signal in this table only

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

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
100	E10	PB0	GPIO Port B	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 B pin, when programmed as output is selected to be an open-drain or open-source output.
		IC0	Input Capture	Schmitt Trigger input	Input Capture A Signal to Timer 1. This signal is multiplexed with PB0.
		EC0	Event Counter	Schmitt Trigger input	Event Counter Signal to Timer 1. This signal is multiplexed with PB0.
101	D12	PB1	GPIO Port B	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 B pin, when programmed as output is selected to be an open-drain or open-source output.
		IC1	Input Capture	Schmitt Trigger input	Input Capture B Signal to Timer 1. This signal is multiplexed with PB1.
102	F8	PB2	GPIO Port B	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 B pin, when programmed as output is selected to be an open-drain or open-source output.
		SS	SPI Slave Select	Schmitt Trigger input, Active Low	The slave select input line is used to select a slave device in SPI Mode. This signal is multiplexed with PB2.

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

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
106	C12	PB6	GPIO Port B	Bidirectional with Schmitt Trigger input	This pin is be used for GPIO. It is individually programmed as input or output and is also used individually as an interrupt input. Each Port B pin, when programmed as output is selected to be an open-drain or open-source output.
		MISO	SPI Master-In/ Slave-Out	Bidirectional with Schmitt Trigger input	The MISO line is configured as an input when the eZ80F91 device is an SPI master device and as an output when eZ80F91 is an SPI slave device. This signal is multiplexed with PB6.
107	C11	PB7	GPIO Port B	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 B pin, when programmed as output is selected to be an open-drain or open-source output.
		MOSI	SPI Master Out Slave In	Bidirectional with Schmitt Trigger input	The MOSI line is configured as an output when the eZ80F91 device is an SPI master device and as an input when the eZ80F91 device is an SPI slave device. This signal is multiplexed with PB7.
108	B12	V_{SS}	Ground		Ground.
109	A12	SDA	I^2C Serial Data	Bidirectional	This pin carries the I^2C data signal.
110	A11	SCL	I^2C Serial Clock	Bidirectional	This pin is used to receive and transmit the I^2C clock.
111	B11	PHI	System Clock	Output	This pin is an output driven by the internal system clock. It is used by the system for synchronization with the eZ80F91 device.
112	C10	V_{DD}	Power Supply		Power Supply.
113	D9	V_{SS}	Ground		Ground.

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

LQFP Pin No	BGA Pin No	Symbol	Function	Signal Direction	Description
114	A10	PA0	GPIO Port A	Bidirectional	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 A pin, when programmed as output is selected to be an open-drain or open-source output.
		PWM0	PWM Output 0	Output	This pin is used by Timer 3 for PWM 0. This signal is multiplexed with PA0.
		OC0	Output Compare 0	Output	This pin is used by Timer 3 for Output Compare 0. This signal is multiplexed with PA0.
115	B10	PA1	GPIO Port A	Bidirectional	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 A pin, when programmed as output is selected to be an open-drain or open-source output.
		PWM1	PWM Output 1	Output	This pin is used by Timer 3 for PWM 1. This signal is multiplexed with PA1.
		OC1	Output Compare 1	Output	This pin is used by Timer 3 for Output Compare 1. This signal is multiplexed with PA1.
116	E8	PA2	GPIO Port A	Bidirectional	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 A pin, when programmed as output is selected to be an open-drain or open-source output.
		PWM2	PWM Output 2	Output	This pin is used by Timer 3 for PWM 2. This signal is multiplexed with PA2.
		OC2	Output Compare 2	Output	This pin is used by Timer 3 for Output Compare 2. This signal is multiplexed with PA2.

Table 17. Example: Priority Levels for Maskable Interrupts

Priority	Vector	Source
0	044h	EMAC Tx
1	06Ch	RTC
2	084h	Port A 1
3	040h	EMAC Rx
4	048h	EMAC SYS
5	04Ch	PLL
6	050h	Flash
7	054h	Timer 0
8	058h	Timer 1
9	05Ch	Timer 2

GPIO Port Interrupts

All interrupts are latched. In effect, an interrupt is held even if the interrupt occurs while another interrupt is being serviced and interrupts are disabled, or if the interrupt is of a lower priority. However, before the latched ISR completes its task or reenables interrupts, the ISR must clear the interrupt. For on-chip peripherals, the interrupt is cleared when the data register is accessed. *For GPIO-level interrupts, the interrupt signal must be removed before the ISR completes its task.* For GPIO-edge interrupts (single and dual), the interrupt is cleared by writing a 1 to the corresponding bit position in the Px_ALT0 Register. See the Edge-Triggered Interrupts section on page 50.

Note: For eZ80F91 devices with a ZDI or JTAG revision less than 2, care must be taken using a GPIO data register when it is configured for interrupts. For edge-interrupt modes (modes 6 and 9) as discussed earlier, writing 1 clears the interrupt. However, 1 in the data register also conveys a particular configuration. For example, when the data register Px_DR is set first followed by the Px_ALT2, Px_ALT1, and Px_DDR registers, then the configuration is performed correctly. Writing 1 to the register later to clear interrupts does not change the configuration. For eZ80F91 devices with a ZDI or JTAG revision 2 or later, the clearing of interrupts is accomplished through the new Px_ALT0 registers and the above problem does not exist.

In Mode 9 operation, if the GPIO is already configured for Mode 9 and if the trigger edge must be changed (from falling to rising or from rising to falling), then the configuration must be changed to another mode, such as Mode 2, and then changed back to Mode 9. For example, enter Mode 2 by writing the registers in the sequence PxDR, Px_ALT2,

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 $\overline{\text{WAIT}}$ 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.

Flash Frequency Divider Register

The 8-bit frequency divider allows the programming of Flash memory over a range of system clock frequencies. Flash is programmed with system clock frequencies ranging from 154 kHz to 50 MHz. The Flash controller requires an input clock with a period that falls within the range of 5.1-6.5 µs. The period of the Flash controller clock is set in the Flash Frequency Divider Register. Writes to this register is allowed only after it is unlocked via the FLASH_KEY Register. The Flash Frequency Divider Register value required versus the system clock frequency is shown in Table 39. System clock frequencies outside of the ranges shown are not supported. Register values for the Flash Frequency Divider are shown in Table 40.

Table 39. Flash Frequency Divider Values

System Clock Frequency	Flash Frequency Divider Value
154–196 kHz	1
308–392 kHz	2
462–588 kHz	3
616 kHz–50 MHz	CEILING [System Clock Frequency (MHz) x 5.1 (µs)]*

Note: *The CEILING function rounds fractional values up to the next whole number. For example, CEILING(3.01) is 4.

Table 40. Flash Frequency Divider Register (FLASH_FDIV)

Bit	7	6	5	4	3	2	1	0
Field	FLASH_FDIV							
Reset	0	0	0	0	0	0	0	1
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W
Address	00F9h							

Note: *Key sequence required to enable writes; R/W = read/write, R = read only.

Bit	Description
[7:0] FLASH_FDIV	**Flash Frequency Divider** 01h–FFh: Divider value for generating the required 5.1-6.5 µs Flash controller clock period.

Bit	Description (Continued)
[1:0] WDT_PERIOD	**Watchdog Timer Period**
	00: WDT_CLK = 00: WDT time-out period is 2^{27} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{17} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{15} clock cycles. WDT_CLK = 11: reserved.
	01: WDT_CLK = 00: WDT time-out period is 2^{25} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{14} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{13} clock cycles. WDT_CLK = 11: reserved.
	10: WDT_CLK = 00: WDT time-out period is 2^{22} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{11} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{9} clock cycles. WDT_CLK = 11: reserved.
	11: WDT_CLK = 00: WDT time-out period is 2^{18} clock cycles. WDT_CLK = 01: WDT time-out period is 2^{7} clock cycles. WDT_CLK = 10: WDT time-out period is 2^{5} clock cycles. WDT_CLK = 11: reserved.
Note: When the WDT is enabled, no writes are allowed to the WDT_CTL Register.	

Watchdog Timer Reset Register

The WDT Reset Register, shown in Table 49, is an 8-bit write-only register. The WDT is reset when an A5h value followed by a 5Ah value is written to this register. Any amount of time occurs between the writing of A5h value and the 5Ah value, so long as the WDT time-out does not occur prior to completion. Any value other than 5Ah written to the WDT Reset Register after the A5h value requires that the sequence of writes (A5h,5Ah) be restarted for the timer to be reset.

Table 49. Watchdog Timer Reset Register (WDT_RR)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	WDT_RR							
Reset	U	U	U	U	U	U	U	U
R/W	W	W	W	W	W	W	W	W
Address	0094h							

Note: U = undefined; W = write only.

Bit	Description
[7:0] WDT_RR	**Watchdog Timer Reset** A5h: The first write value required to reset the WDT prior to a time-out. 5Ah: The second write value required to reset the WDT prior to a time-out. If an A5h, 5Ah sequence is written to WDT_RR, the WDT timer is reset to its initial count value and counting resumes.

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.

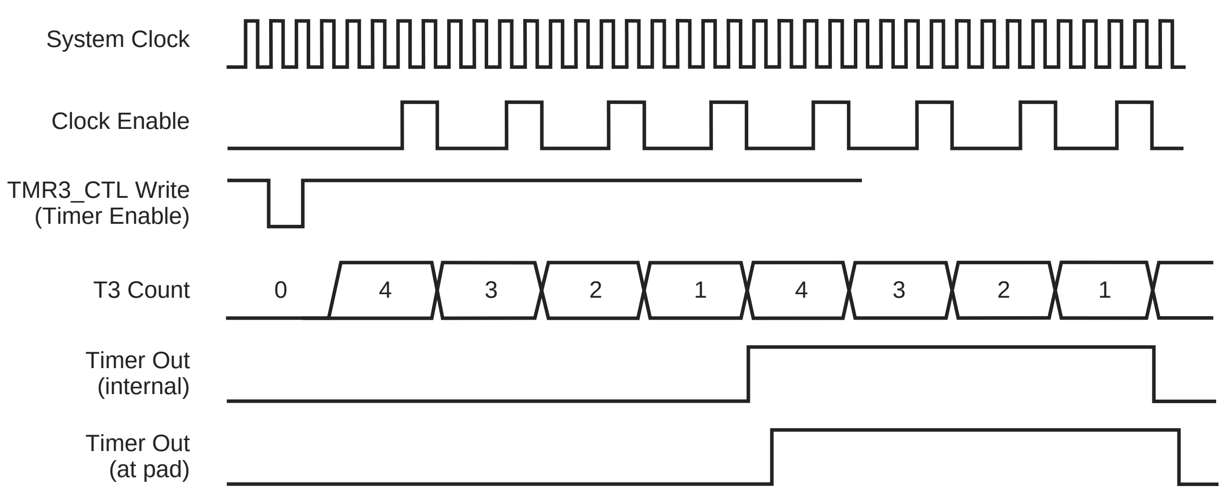

Figure 29. Example: PRT Timer Output Operation

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

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

Note: W = write only.

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

Timer Input Capture Control Register

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

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

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

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

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

Table 111. SPI Clock Phase and Clock Polarity Operation (Continued)

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

SPI Functional Description

When a master transmits to a slave device via the MOSI signal, the slave device responds by sending data to the master via the master's MISO signal. The result is a full-duplex transmission, with both *data out* and *data in* synchronized with the same clock signal. The byte transmitted is replaced by the byte received, eliminating the need for separate transmit-empty and receive-full status bits. A single status bit, SPIF, is used to signify that the I/O operation is complete. See the SPI Status Register section on page 206.

The SPI is double-buffered during reads, but not during writes. If a write is performed during data transfer, the transfer occurs uninterrupted, and the write is unsuccessful. This condition causes the write collision (WCOL) status bit in the SPI_SR Register to be set. After a data byte is shifted, the SPI flag of the SPI_SR Register is set to 1.

In SPI MASTER Mode, the SCK pin functions as an output. It idles High or Low depending on the CPOL bit in the SPI_CTL Register until data is written to the shift register. Data transfer is initiated by writing to the transmit shift register, SPI_TSR. Eight clocks are then generated to shift the eight bits of transmit data out via the MOSI pin while shifting in eight bits of data via the MISO pin. After transfer, the SCK signal becomes idle.

In SPI SLAVE Mode, the start logic receives a logic Low from the $\overline{SS}$ pin and a clock input at the SCK pin; as a result, the slave is synchronized to the master. Data from the master is received serially from the slave MOSI signal and is loaded into the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel-transferred to the read buffer. During a write cycle, data is written into the shift register. Next, the slave waits for the SPI master to initiate a data transfer, supply a clock signal, and shift the data out on the slave's MISO signal.

If the CPHA bit in the SPI_CTL Register is 0, a transfer begins when the $\overline{SS}$ pin signal goes Low. The transfer ends when $\overline{SS}$ goes High after eight clock cycles on SCK. When the CPHA bit is set to 1, a transfer begins the first time SCK becomes active while $\overline{SS}$ is Low. The transfer ends when the SPI flag is set to 1.

I^2C Clock Control Register

The I^2C_CCR Register is a write-only register. The seven LSBs control the frequency at which the I^2C bus is sampled and the frequency of the I^2C clock line (SCL) when the I^2C is in MASTER Mode. The write-only I^2C_CCR registers share the same I/O addresses as the read-only I2C_SR registers. See Table 130.

Table 130. I^2C Clock Control Registers (I2C_CCR)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	M				N		
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	00CCh							

Note: W = read only.

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6:3] M	**Scalar Value** 0000–1111: I^2C clock divider scalar value; see the equations that follow.
[2:0] N	**Exponential Value** 000–111: I^2C clock divider exponent; see the equations that follow.

The I^2C clocks are derived from the system clock of the eZ80F91 device. The frequency of this system clock is f_{SCK}. The I^2C bus is sampled by the I^2C block at the frequency f_{SAMP} supplied by the following equation:

$$f_{SAMP} = \frac{f_{SCLK}}{2^N}$$

In MASTER Mode, the I^2C clock output frequency on SCL (f_{SCL}) is supplied by the following equation:

$$f_{SCL} = \frac{f_{SCLK}}{10 \bullet (M+1)(2)^N}$$

The use of two separately-programmable dividers allows the MASTER Mode output frequency to be set independently of the frequency at which the I^2C bus is sampled. This feature is particularly useful in multimaster systems because the frequency at which the I^2C

Bus Requests During ZDI Debug Mode

The ZDI block on the eZ80F91 device allows an external device to take control of the address and data bus while the eZ80F91 device is in DEBUG Mode. ZDI_BUSACK_EN causes ZDI to allow or prevent acknowledgement of bus requests by external peripherals. The bus acknowledge occurs only at the end of the current ZDI operation (indicated by a High during the single-bit byte separator). The default reset condition is for bus acknowledgement to be disabled. To allow bus acknowledgement, the ZDI_BUSACK_EN must be written.

When an external bus request ($\overline{\text{BUSREQ}}$ pin asserted) is detected, ZDI waits until completion of the current operation before responding. ZDI acknowledges the bus request by asserting the bus acknowledge ($\overline{\text{BUSACK}}$) signal. If the ZDI block is not currently shifting data, it acknowledges the bus request immediately. ZDI uses the single-bit byte separator of each data word to determine if it is at the end of a ZDI operation. If the bit is a logic 0, ZDI does not assert $\overline{\text{BUSACK}}$ to allow additional data read or write operations. If the bit is a logic 1, indicating completion of the ZDI commands, $\overline{\text{BUSACK}}$ is asserted.

Potential Hazards of Enabling Bus Requests During DEBUG Mode

There are some potential hazards that you must be aware of when enabling external bus requests during ZDI DEBUG Mode. First, when the address and data bus are being used by an external source, ZDI must only access ZDI registers and internal CPU registers to prevent possible bus contention. The bus acknowledge status is reported in the ZDI_BUS_STAT Register. The $\overline{\text{BUSACK}}$ output pin also indicates the bus acknowledge state.

A second hazard is that when a bus acknowledge is granted, the ZDI is subject to any wait states that are assigned to the device currently being accessed by the external peripheral. To prevent data errors, ZDI must avoid data transmission while another device is controlling the bus.

Finally, exiting ZDI DEBUG Mode while an external peripheral controls the address and data buses, as indicated by $\overline{\text{BUSACK}}$ assertion produces unpredictable results.

ZDI Status Register

The ZDI Status Register, shown in Table 147, provides current information about the eZ80F91 device and the CPU.

Table 147. ZDI Status Register (ZDI_STAT)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	ZDI_ACTIVE	Reserved	HALT_SLP	ADL	MADL	IEF1	Reserved	
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	ZDI_STAT = 03h in the ZDI Register read-only address space							

Note: R = read only.

Bit	**Description**
[7] ZDI_ACTIVE	**ZDI Mode** 0: The CPU is not functioning in ZDI Mode. 1: The CPU is currently functioning in ZDI Mode.
[6]	**Reserved** This bit is reserved and must be programmed to 0.
[5] HALT_SLP	**HALT/SLEEP Modes** 0: The CPU is not currently in HALT or SLEEP Mode. 1: The CPU is currently in HALT or SLEEP Mode.
[4] ADL	**Z80 MEMORY Mode** 0: The CPU is operating in Z80 MEMORY Mode (ADL bit = 0). 1: The CPU is operating in ADL MEMORY Mode (ADL bit = 1).
[3] MADL	**MIXED MEMORY Mode** 0: The CPU's MIXED-MEMORY Mode (MADL) bit is reset to 0. 1: The CPU's MIXED-MEMORY Mode (MADL) bit is set to 1.
[2] IEF1	**Interrupt Enable Flag 1** 0: The CPU's Interrupt Enable Flag 1 is reset to 0. Maskable interrupts are disabled. 1: The CPU's Interrupt Enable Flag 1 is set to 1. Maskable interrupts are enabled.
[1:0]	**Reserved** These bits are reserved and must be programmed to 00.

Op Code Map

Tables 168 through 174 list the hex values for each of the eZ80 instructions.

Table 168. Op Code Map: First Op Code

Legend

Lower Op Code Nibble

Upper Op Code Nibble

4

A AND A,H Mnemonic

First Operand   Second Operand

Lower Nibble (Hex) / **Upper Nibble (Hex)**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LD BC,Mmn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRC A
1	DJNZ d	LD DE,Mmn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR d	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRA
2	JR NZ,d	LD HL,Mmn	LD (Mmn),HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,d	ADD HL,HL	LD HL,(Mmn)	DEC HL	INC L	DEC L	LD L,n	CPL
3	JR NC,d	LD SP,Mmn	LD (Mmn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF	JR CF,d	ADD HL,SP	LD A,(Mmn)	DEC SP	INC A	DEC A	LD A,n	CCF
4	.SIS suffix	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	.LIS suffix	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
5	LD D,B	LD D,C	.SIL suffix	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	.LIL suffix	LD E,H	LD E,L	LD E,(HL)	LD E,A
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A
9	SUB A,B	SUB A,C	SUB A,D	SUB A,E	SUB A,H	SUB A,L	SUB A,(HL)	SUB A,A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A
A	AND A,B	AND A,C	AND A,D	AND A,E	AND A,H	AND A,L	AND A,(HL)	AND A,A	XOR A,B	XOR A,C	XOR A,D	XOR A,E	XOR A,H	XOR A,L	XOR A,(HL)	XOR A,A
B	OR A,B	OR A,C	OR A,D	OR A,E	OR A,H	OR A,L	OR A,(HL)	OR A,A	CP A,B	CP A,C	CP A,D	CP A,E	CP A,H	CP A,L	CP A,(HL)	CP A,A
C	RET NZ	POP BC	JP NZ,Mmn	JP Mmn	CALL NZ,Mmn	PUSH BC	ADD A,n	RST 00h	RET Z	RET	JP Z,Mmn	See Table 169	CALL Z,Mmn	CALL Mmn	ADC A,n	RST 08h
D	RET NC	POP DE	JP NC,Mmn	OUT (n),A	CALL NC,Mmn	PUSH DE	SUB A,n	RST 10h	RET CF	EXX	JP CF,Mmn	IN A,(n)	CALL CF,Mmn	See Table 170	SBC A,n	RST 18h
E	RET PO	POP HL	JP PO,Mmn	EX (SP),HL	CALL PO,Mmn	PUSH HL	AND A,n	RST 20h	RET PE	JP (HL)	JP PE,Mmn	EX DE,HL	CALL PE,Mmn	See Table 171	XOR A,n	RST 28h
F	RET P	POP AF	JP P,Mmn	DI	CALL P,Mmn	PUSH AF	OR A,n	RST 30h	RET M	LD SP,HL	JP M,Mmn	EI	CALL M,Mmn	See Table 172	CP A,n	RST 38h

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

Table 214. EMAC Receive Read Pointer High Byte Register (EMAC_RRP_H)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	EMAC_RRP_H							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Address	004Ah							

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

Bit	Description
[7:0] EMAC_RRP_H	**Receive Read Pointer High Byte** 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.

UART Modem Control 174
    Register 188
UART Modem Status Interrupt 176
UART Modem Status Register 191
UART Receive Buffer Register 182
UART Receiver 174
UART Receiver Interrupts 175
UART Recommended Usage 176
UART Registers 181
UART Scratch Pad Register 192
UART Transmit Holding Register 181
UART Transmitter 173
UART Transmitter Interrupt 175
Universal Asynchronous Receiver/Transmitter 172
Usage, JTAG Boundary Scan 263

V

VBO 38, 39, 339
    pulse reject period 339
    Voltage Threshold 339
VCC 2, 39, 339
    ramp rate 339
VCO 266, 273
VLAN tagged frame 308
Voltage Brown-Out 339
    Reset 39
voltage signal, high 97
voltage, input 266
voltage, peak-to-peak 273
voltage, supply 2, 48, 209, 267, 336, 337
voltage-controlled oscillator 265
    PLL 266

W

wait 1, 9, 80
wait condition 109
WAIT Input Signal 66
wait pin, external 68, 69
wait state 69, 76, 349, 350
Wait State Timing for Read Operations 349
Wait State Timing for Write Operations 350
wait states 55, 65, 73, 76, 85, 238
Watchdog Timer 1, 41, 111, 112, 237
    Control Register 113
    Operation 112
    Registers 113
    Reset Register 116
    time-out 38, 41, 42
WCOL—see write collision 201, 202, 206
WDT 38, 41, 111, 112, 113
    clock source 111, 112, 114
    oscillator 113
    time-out 111, 113, 114, 116
    time-out period 112, 115
WP 24
WP pin 94, 104, 105, 106
WR 8, 63, 65, 69, 73, 76
WR assertion delay 346, 348
write collision 201, 202
    SPI 206

X

XIN input pin 332
XOUT output pin 332

Z

Z80 BUS Mode 68
ZCL 232, 235, 243
ZDA 232, 243, 257
ZDI 230, 231, 256
    Address Match Registers 241
    Block Read 237
    Block Write 235
    Break Control Register 242
    Bus Control Register 248
    Bus Status Register 254
    Clock and Data Conventions 232
    clock pin 232
    data pin 232
    debug control 256
    Master Control Register 244
    Read Memory Register 254
    Read Operations 236
    Read Register Low, High, and Upper 253