Zilog - EZ80F92AZ020EC00TR Datasheet

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	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	24
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80f92az020ec00tr

Table 1. 100-Pin LQFP Pin Identification of the eZ80F92 Device (Continued)

Pin No	Symbol	Function	Signal Direction	Description
82	PC6	GPIO Port C	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port C pin, when programmed as output, can be selected to be an open-drain or open-source output. Port C is multiplexed with one UART.
	$\overline{DCD1}$	Data Carrier Detect	Input, Active Low	Modem status signal to the UART. This signal is multiplexed with PC6.
83	PC7	GPIO Port C	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port C pin, when programmed as output, can be selected to be an open-drain or open-source output. Port C is multiplexed with one UART.
	$\overline{RI1}$	Ring Indicator	Input, Active Low	Modem status signal to the UART. This signal is multiplexed with PC7.
84	V_{SS}	Ground		Ground.
85	X_{IN}	System Clock Oscillator Input	Input	This pin is the input to the onboard crystal oscillator for the primary system clock. If an external oscillator is used, its clock output should be connected to this pin. When a crystal is used, it should be connected between X_{IN} and X_{OUT}.
86	X_{OUT}	System Clock Oscillator Output	Output	This pin is the output of the onboard crystal oscillator. When used, a crystal should be connected between X_{IN} and X_{OUT}.
87	V_{DD}	Power Supply		Power Supply.

Table 1. 100-Pin LQFP Pin Identification of the eZ80F92 Device (Continued)

Pin No	Symbol	Function	Signal Direction	Description
92	PB4	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	T4_OUT	Timer 4 Out	Output	Programmable Reload Timer 4 timer-out signal. This signal is multiplexed with PB4.
93	PB5	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	T5_OUT	Timer 5 Out	Output	Programmable Reload Timer 5 timer-out signal. This signal is multiplexed with PB5.
94	PB6	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	MISO	Master In, Slave Out	Bidirectional	The MISO line is configured as an input when the CPU is an SPI master device and as an output when CPU is an SPI slave device. This signal is multiplexed with PB6.
95	PB7	GPIO Port B	Bidirectional	This pin can be used for general-purpose I/O. It can be individually programmed as input or output and can also be used individually as an interrupt input. Each Port B pin, when programmed as output, can be selected to be an open-drain or open-source output.
	MOSI	Master Out, Slave In	Bidirectional	The MOSI line is configured as an output when the CPU is an SPI master device and as an input when the CPU is an SPI slave device. This signal is multiplexed with PB7.

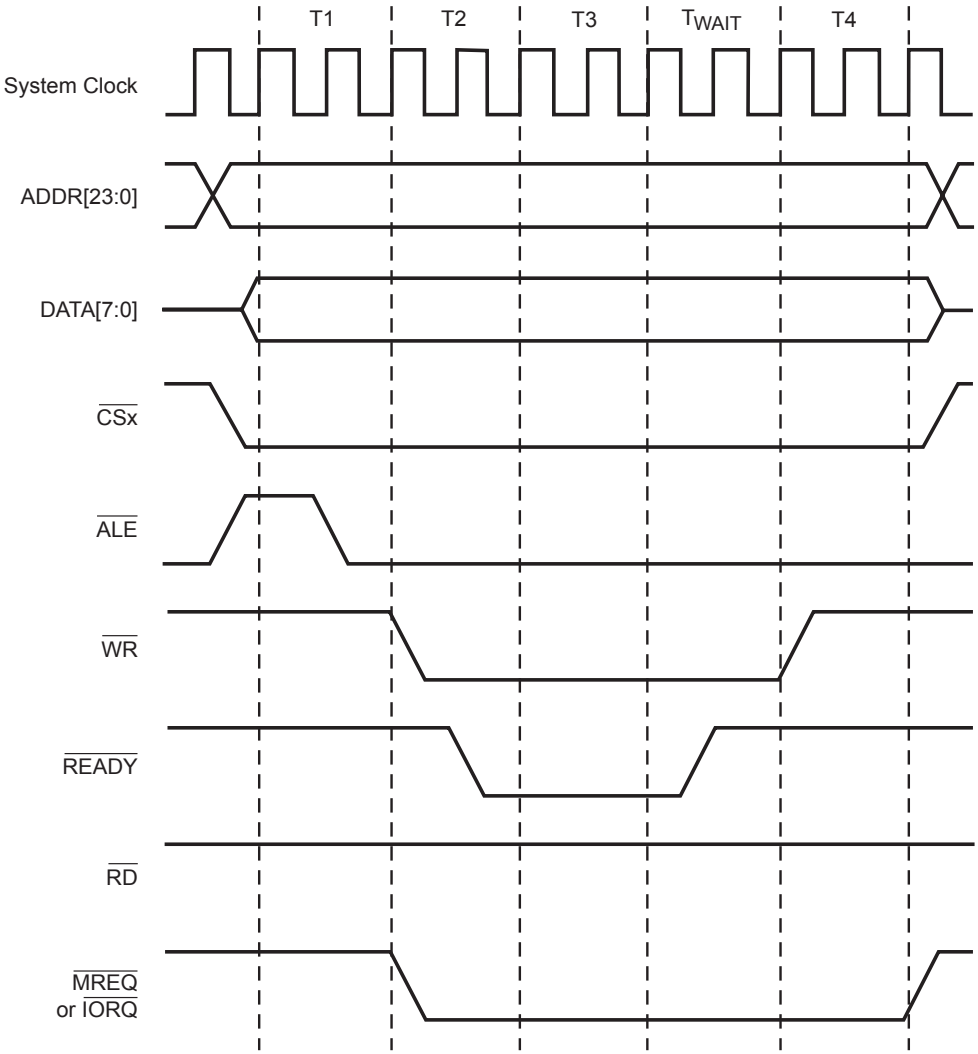

Figure 13. Example: Intel™ Bus Mode Write Timing—Separate Address and Data Buses

Intel™ Bus Mode (Multiplexed Address and Data Bus)

During Read operations with multiplexed address and data, the Intel bus mode employs four states (T1, T2, T3, and T4) as listed in Table 18.

Table 18. Intel Bus Mode Read States (Multiplexed Address and Data Bus)

STATE T1	The Read cycle begins in State T1. The CPU drives the address onto the DATA bus and the associated Chip Select signal is asserted. 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 removes the address from the DATA bus and asserts the $\overline{RD}$ signal. Depending upon 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 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 multiplexed address and data, the Intel bus mode employs four states (T1, T2, T3, and T4) as listed in Table 19.

Table 19. Intel Bus Mode Write States (Multiplexed Address and Data Bus)

STATE T1	The Write cycle begins in State T1. The CPU drives the address onto the DATA bus and 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 removes the address from the DATA bus and drives the Write data onto the DATA bus. The $\overline{WR}$ signal is asserted to indicate a Write operation.
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 Write signal at the beginning of T4 identifying the end of the Write operation. The CPU holds the data and address buses through the end of T4. The bus cycle is completed at the end of T4.

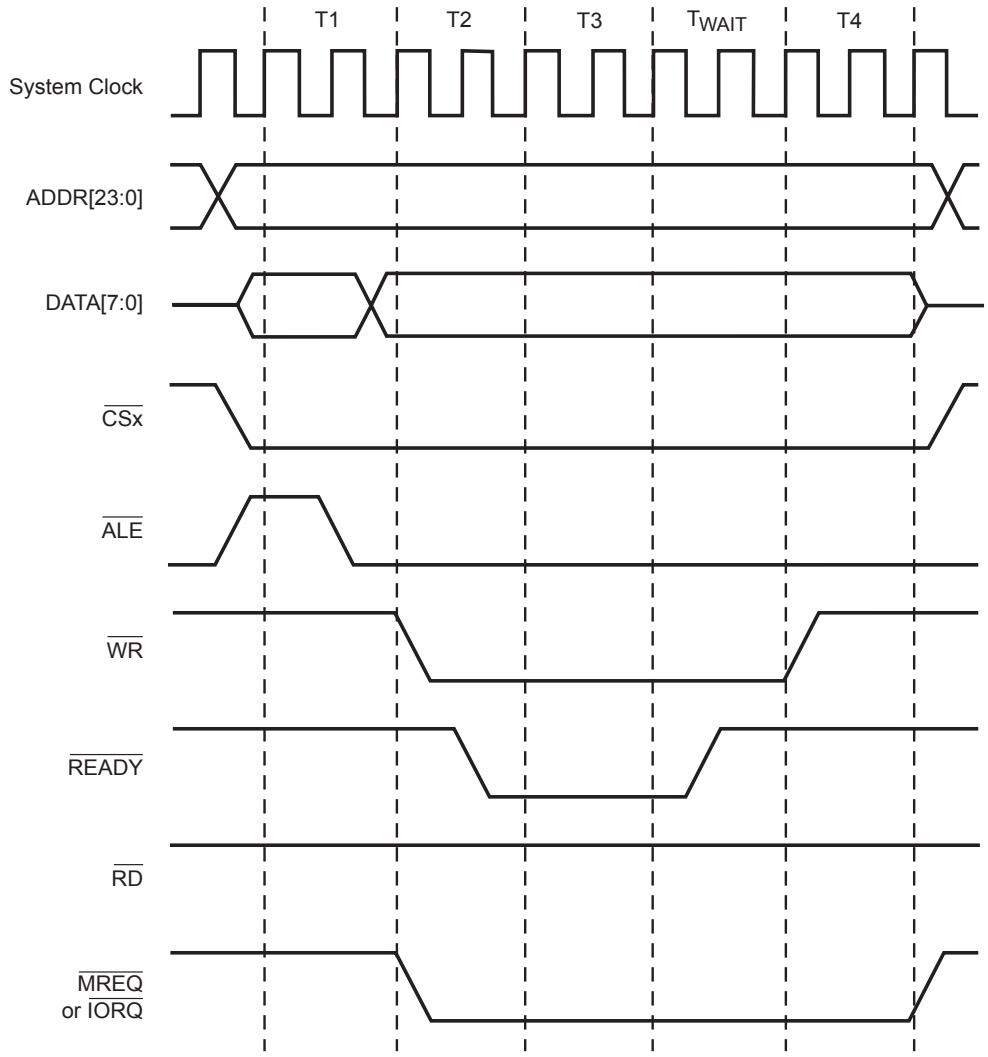

Figure 15. Example: Intel™ Bus Mode Write Timing—Multiplexed Address and Data Bus

Signal timing for Motorola bus mode is displayed for a Read operation in Figure 17 and for a Write operation in Figure 18 on page 66. In these two figures, each Motorola bus mode state is 2 CPU system clock cycles in duration.

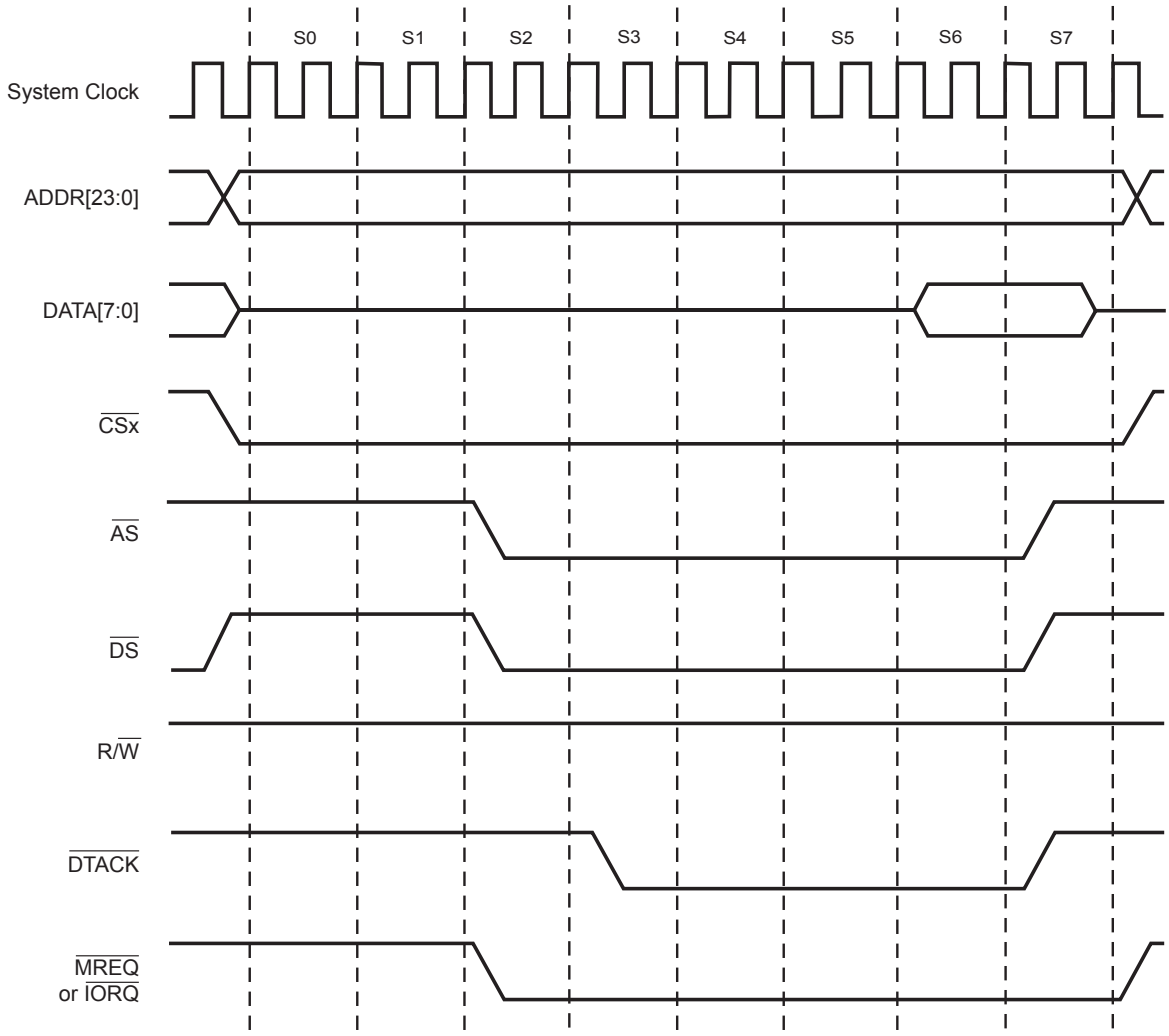

Figure 17. Example: Motorola Bus Mode Read Timing

Programmable Reload Timers

Programmable Reload Timers Overview

The eZ80F92 device features six Programmable Reload Timers (PRT). Each PRT contains a 16-bit downcounter and a 16-bit reload register. In addition, each PRT features a clock divider with four selectable taps for CLK ÷ 4, CLK ÷ 16, CLK ÷ 64, and CLK ÷ 256. Each timer can be individually enabled to operate in either SINGLE PASS or CONTINUOUS mode. The timer can be programmed to start, stop, restart from the current value, or restart from the initial value, and generate interrupts to the CPU.

Four of the Programmable Reload Timers (timers 0–3) feature a selectable clock source input. The input for these timers can be either the system clock or the Real-Time Clock (RTC) source. Timers 0–3 can also be used for event counting, with their inputs received from a GPIO port pin. Output from timers 4 and 5 can be directed to a GPIO port pin.

Each of the six PRTs available on the eZ80F92 device can be controlled individually. They do not share the same counters, reload registers, control registers, or interrupt signals. A simplified block diagram of a programmable reload timer is displayed in Figure 20.

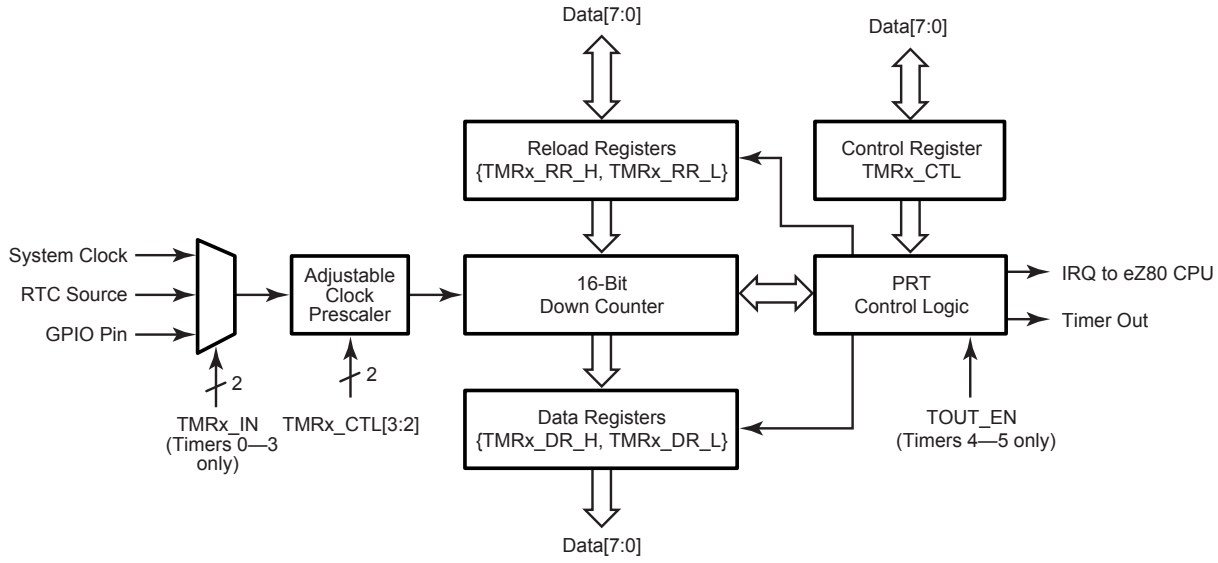

Figure 20.Programmable Reload Timer Block Diagram

Timer Reload Register—High Byte

The Timer Reload Register—High Byte, listed in Table 36, stores the most-significant byte (MSB) of the 2-byte timer reload value. In CONTINUOUS mode, the timer reload value is reloaded into the timer upon end-of-count. When RST_EN (TMRx_CTL[1]) is set to 1 to enable the automatic reload and restart function, the timer reload value is written to the timer on the next rising edge of the clock.

Note: *The Timer Data registers and Timer Reload registers share the same address space.*

Table 36. Timer Reload Register—High Byte(TMR0_RR_H = 0082h, TMR1_RR_H = 0085h, TMR2_RR_H = 0088h, TMR3_RR_H = 008Bh, TMR4_RR_H = 008Eh, or TMR5_RR_H = 0091h)

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] TMRx_RR_H	00h–FFh	These bits represent the High byte of the 2-byte timer reload value, {TMRx_RR_H[7:0], TMRx_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 Source Select Register

The Timer Input Source Select register, listed in Table 37 on page 86, sets the input source for Programmable Reload Timer 0–3 (TMR0, TMR1, TMR2, TMR3). Event frequency must be less than one-half of the system clock frequency. When configured for event inputs through the port pins, the Timers decrement on the fifth system clock rising edge following the rising edge of the port pin. The timer event input can arrive from the GPIO port, the real-time clock, or the system clock. The value of the clock divider in the Timer Control Register is ignored when the timer event input is either from the GPIO port pin or the real-time clock source.

Real-Time Clock Alarm

The clock can be programmed to generate an alarm condition when the current count matches the alarm set-point registers. Alarm registers are available for seconds, minutes, hours, and day-of-the-week. Each alarm can be independently enabled. To generate an alarm condition, the current time must match all enabled alarm values. For example, if the day-of-the-week and hour alarms are both enabled, the alarm only occurs at the specified hour on the specified day. The alarm triggers an interrupt if the interrupt enable bit, INT_EN, is set. The alarm flag, ALARM, and corresponding interrupt to the CPU are cleared by reading the RTC_CTRL register.

Alarm value registers and alarm control registers can be written at any time. Alarm conditions are generated when the count value matches the alarm value. The comparison of alarm and count values occurs whenever the RTC count increments (one time every second). The RTC can also be forced to perform a comparison at any time by writing a 0 to the RTC_UNLOCK bit (RTC_UNLOCK is not required to be changed to a 1 first).

Real-Time Clock Oscillator and Source Selection

The RTC count is driven by either an external 32 kHz on-chip oscillator or a 50/60 Hz power-line frequency input connected to the 32 kHz RTC_X_{OUT} pin. An internal divider compensates for each of these options. The clock source and power-line frequencies are selected in the RTC_CTRL register. Writing to the RTC_CTRL register resets the clock divider.

Real-Time Clock Battery Backup

The power supply pin (RTC_V_{DD}) for the Real-Time Clock and associated low-power 32 kHz oscillator is isolated from the other power supply pins on the eZ80F92 device. To ensure that the RTC continues to keep time in the event of loss of line power to the application, a battery can be used to supply power to the RTC and the oscillator via the RTC_V_{DD} pin. All V_{SS} (ground) pins should be connected together on the printed circuit assembly.

Real-Time Clock Recommended Operation

Following a RESET from a powered-down condition, the counter values of the RTC are undefined and all alarms are disabled.

After a RESET from a powered-down condition, the following procedure is recommended:

- Write to RTC_CTRL to set RTC_UNLOCK and CLK_SEL
- Write values to the RTC count registers to set the current time

- Write values to the RTC alarm registers to set the appropriate alarm conditions
- Write to RTC_CTRL to clear the RTC_UNLOCK bit; clearing the RTC_UNLOCK bit resets and enables the clock divider

Real-Time Clock Registers

The Real-Time Clock registers are accessed via the address and data bus using I/O instructions. RTC_UNLOCK controls access to the RTC count registers. When unlocked (RTC_UNLOCK = 1), the RTC count is disabled and the count registers are Read/Write. When locked (RTC_UNLOCK = 0), the RTC count is enabled and the count registers are Read Only. The default, at RESET, is for the RTC to be locked.

Real-Time Clock Seconds Register

This register contains the current seconds count. The value in the RTC_SEC 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 38.

Table 38. Real-Time Clock Seconds Register; (RTC_SEC = 00E0h)

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] TEN_SEC	0–5	The tens digit of the current seconds count.
[3:0] SEC	0–9	The ones digit of the current seconds count.

Binary Operation (BCD_EN = 0)		
Bit Position	**Value**	**Description**
[7:0] SEC	00h–3Bh	The current seconds count.

Real-Time Clock Century Register

This register contains the current century count. See Table 45.

Table 45. 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. See Table 46.

Table 46. 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.

When the Clock Phase bit (CPHA) is set to 0, the shift clock is the logical OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go High between successive characters in an SPI message. When CPHA is set to 1, $\overline{SS}$ can remain Low for several SPI characters. In cases where there is only one SPI slave, its $\overline{SS}$ line could be tied Low as long as CPHA is set to 1. See (SPI_CTL) on page 136 for more information about CPHA.

Serial Clock

The Serial Clock (SCK) is used to synchronize data movement both in and out of the device through its MOSI and MISO pins. The master and slave are each capable of exchanging a byte of data during a sequence of eight clock cycles. As SCK is generated by the master, the SCK pin becomes an input on a slave device. The SPI contains an internal 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 31 and Table 71 on page 133, four possible timing relations may be chosen by using control bits CPOL and CPHA in the SPI Control register. See the SPI Control Register (SPI_CTL) on page 136. Both the master and slave must operate with the identical timing, clock polarity (CPOL), and clock phase (CPHA). The master device always places data on the MOSI line a half-cycle before the clock edge (SCK signal) so that the slave device latches the data.

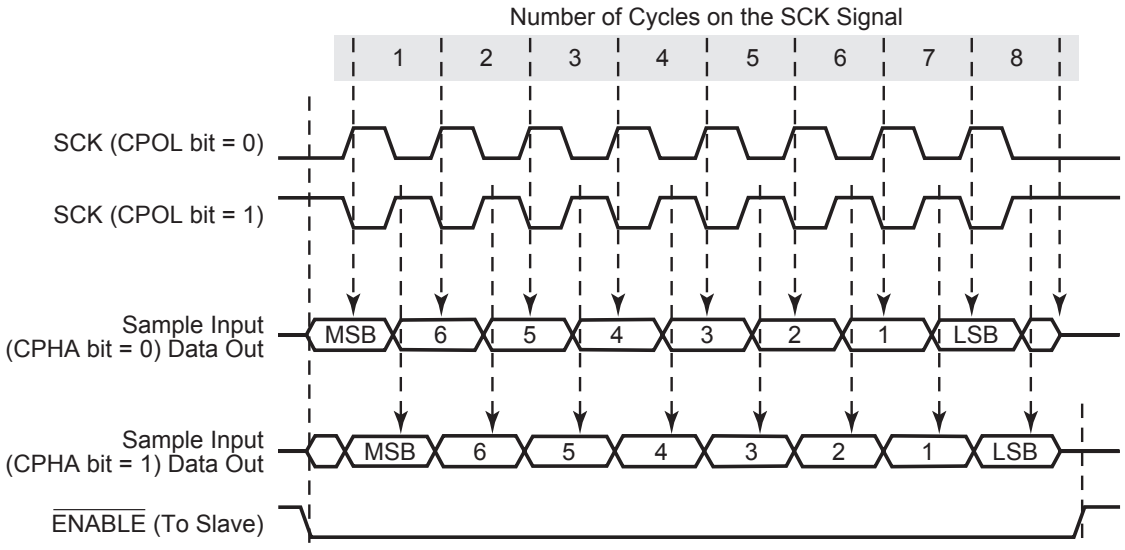

Figure 31. SPI Timing

Table 81. I²C Master Receive Status Codes (Continued)

Code	I²C State	Microcontroller Response	Next I²C Action
48h	Addr + R transmitted, ACK not received	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	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

R = Read bit; that is, the lsb is set to 1.

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 address again, but with the Read bit. The status code then becomes `40h` or `48h`. It is the responsibility of the slave to remember that it had been selected prior to the restart.

If a repeated START condition is received, the status code is `10h` instead of `08h`.

After each data byte is received, the IFLG is set and one of the status codes listed in Table 82 is in the I2C_SR register.

ZDI Write Data Registers

These three registers are used in the ZDI Write Only register address space to store the data that is written when a Write instruction is sent to the ZDI Read/Write Control register (ZDI_RW_CTL). The ZDI Read/Write Control register is located at ZDI address 16h immediately following the ZDI Write Data registers. As a result, the ZDI Master is allowed to write the data to {ZDI_WR_U, ZDI_WR_H, ZDI_WR_L} and the Write command in one data transfer operation. See Table 98.

Table 98. ZDI Write Data Registers(ZDI_WR_U = 13h, ZDI_WR_H = 14h, and ZDI_WR_L = 15h in the ZDI Register Write Only Address Space)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	W	W	W	W	W	W	W	W
Note: X = Undefined; W = Write.								

Bit Position	Value	Description
[7:0] ZDI_WR_L, ZDI_WR_H, or ZDI_WR_L	00h–FFh	These registers contain the data that is written during execution of a Write operation defined by the ZDI_RW_CTL register. The 24-bit data value is stored as {ZDI_WR_U, ZDI_WR_H, ZDI_WR_L}. If less than 24 bits of data are required to complete the required operation, the data is taken from the LSBs.

ZDI Read/Write Control Register

The ZDI Read/Write Control register is used in the ZDI Write Only Register address to read data from, write data to, and manipulate the CPU's registers or memory locations. When this register is written, the eZ80F92 device immediately performs the operation corresponding to the data value written as listed in Table 99 on page 177. When a Read operation is executed via this register, the requested data values are placed in the ZDI Read Data registers {ZDI_RD_U, ZDI_RD_H, ZDI_RD_L}. When a Write operation is executed via this register, the Write data is taken from the ZDI Write Data registers {ZDI_WR_U, ZDI_WR_H, ZDI_WR_L}. See Table 99 on page 177. Refer to the *eZ80® CPU User Manual (UM0077)* for information regarding the CPU registers.

Table 99. ZDI Read/Write Control Register Functions(ZDI_RW_CTL = 16h in the ZDI Register Write Only Address Space)

Hex Value	Command	Hex Value	Command
00	Read {MBASE, A, F} ZDI_RD_U ← MBASE ZDI_RD_H ← F ZDI_RD_L ← A	80	Write AF MBASE ← ZDI_WR_U F ← ZDI_WR_H A ← ZDI_WR_L
01	Read BC ZDI_RD_U ← BCU ZDI_RD_H ← B ZDI_RD_L ← C	81	Write BC BCU ← ZDI_WR_U B ← ZDI_WR_H C ← ZDI_WR_L
02	Read DE ZDI_RD_U ← DEU ZDI_RD_H ← D ZDI_RD_L ← E	82	Write DE DEU ← ZDI_WR_U D ← ZDI_WR_H E ← ZDI_WR_L
03	Read HL ZDI_RD_U ← HLU ZDI_RD_H ← H ZDI_RD_L ← L	83	Write HL HLU ← ZDI_WR_U H ← ZDI_WR_H L ← ZDI_WR_L
04	Read IX ZDI_RD_U ← IXU ZDI_RD_H ← IXH ZDI_RD_L ← IXL	84	Write IX IXU ← ZDI_WR_U IXH ← ZDI_WR_H IXL ← ZDI_WR_L
05	Read IY ZDI_RD_U ← IYU ZDI_RD_H ← IYH ZDI_RD_L ← IYL	85	Write IY IYU ← ZDI_WR_U IYH ← ZDI_WR_H IYL ← ZDI_WR_L
06	Read SP In ADL mode, SP = SPL In Z80® mode, SP = SPS	86	Write SP In ADL mode, SP = SPL In Z80 mode, SP = SPS
07	Read PC ZDI_RD_U ← PC[23:16] ZDI_RD_H ← PC[15:8] ZDI_RD_L ← PC[7:0]	87	Write PC PC[23:16] ← ZDI_WR_U PC[15:8] ← ZDI_WR_H PC[7:0] ← ZDI_WR_L
08	Set ADL ADL ← 1	88	Reserved
09	Reset ADL ADL ← 0	89	Reserved

ZDI Write Memory Register

A Write to the ZDI Write Memory register causes the eZ80F92 device to write the 8-bit data to the memory location specified by the current address in the program counter. In Z80® MEMORY mode, this address is {MBASE, PC[15:0]}. In ADL MEMORY mode, this address is PC[23:0]. The program counter, PC, increments after each data Write. However, the ZDI register address does not increment automatically when this register is accessed. As a result, the ZDI master is allowed to write any number of data bytes by writing to this address one time followed by any number of data bytes. See Table 102.

Table 102. ZDI Write Memory Register(ZDI_WR_MEM = 30h in the ZDI Register Write Only Address Space)

Bit	7	6	5	4	3	2	1	0
Reset	X	X	X	X	X	X	X	X
CPU Access	W	W	W	W	W	W	W	W
Note: X = Undefined; W = Write.								

Bit Position	Value	Description
[7:0] ZDI_WR_MEM	00h–FFh	The 8-bit data that is transferred to the ZDI slave following a Write to this address is written to the address indicated by the current program counter. The program counter is incremented following each 8 bits of data. In Z80 MEMORY mode, ({MBASE, PC[15:0]}) ← 8 bits of transferred data. In ADL MEMORY mode, (PC[23:0]) ← 8 bits of transferred data.

Electrical Characteristics

Absolute Maximum Ratings

Stresses greater than those listed in Table 144 may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, unused inputs must be tied to one of the supply voltages (V_{DD} or V_{SS}).

Table 144. Absolute Maximum Ratings

Parameter	Min	Max	Units	Notes
Ambient temperature under bias (ºC)	–40	+105	C	1
Storage temperature (ºC)	–65	+150	C	
Voltage on any pin with respect to V_{SS}	–0.3	+5.5	V	2
Voltage on V_{DD} pin with respect to V_{SS}	–0.3	+3.6	V	
Total power dissipation		520	mW	
Maximum current out of V_{SS}		145	mA	
Maximum current into V_{DD}		145	mA	
Maximum current on input and/or inactive output pin	–25	+25	µA	
Maximum output current from active output pin	–8	+8	mA	
Flash memory writes to same single address	-	2	-	3
Flash memory data retention	100	-	Years	
Flash memory write/erase endurance	10,000		Cycles	4

Notes
1. Operating temperature is specified in DC Characteristics.
2. This voltage applies to all pins except where noted otherwise.
3. Before next erase operation.
4. Write cycles.

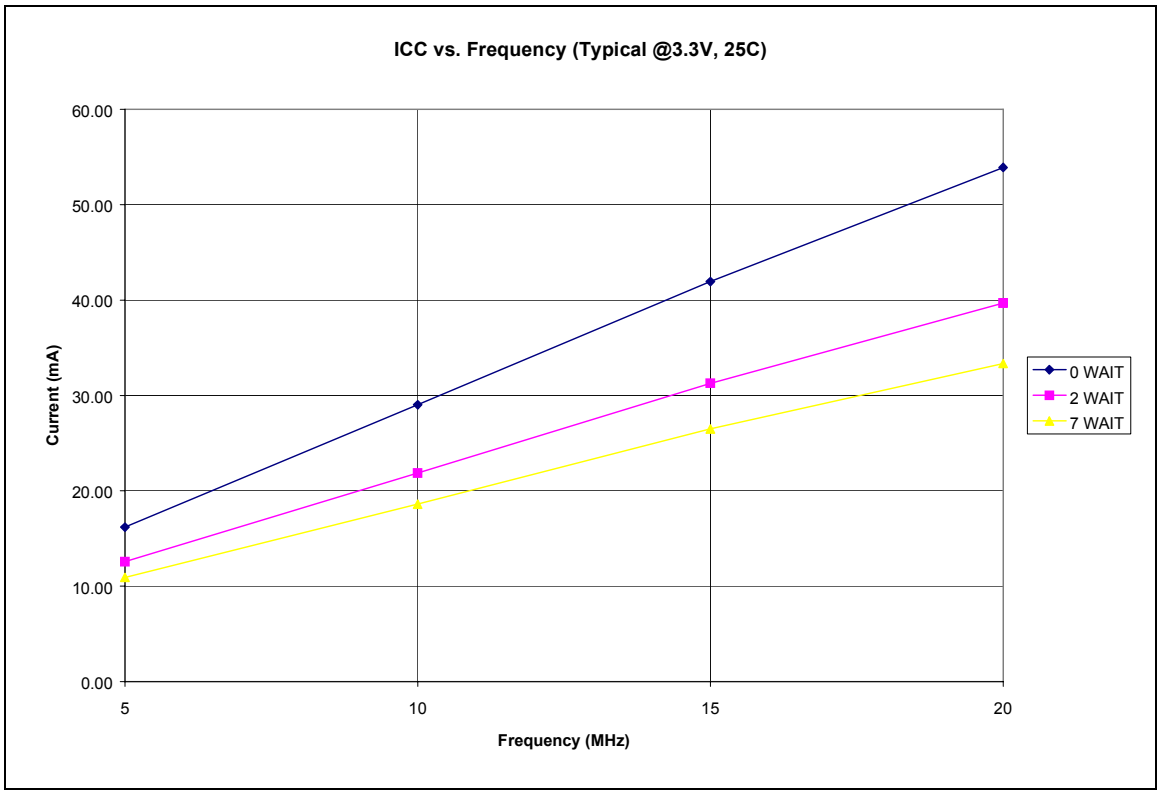

Figure 54. I_{CC} Versus Frequency as a Function of WAIT States

T

U

V