Zilog - EZ80F93AZ020SC00TR 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	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K 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	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80f93az020sc00tr

Table of Contents

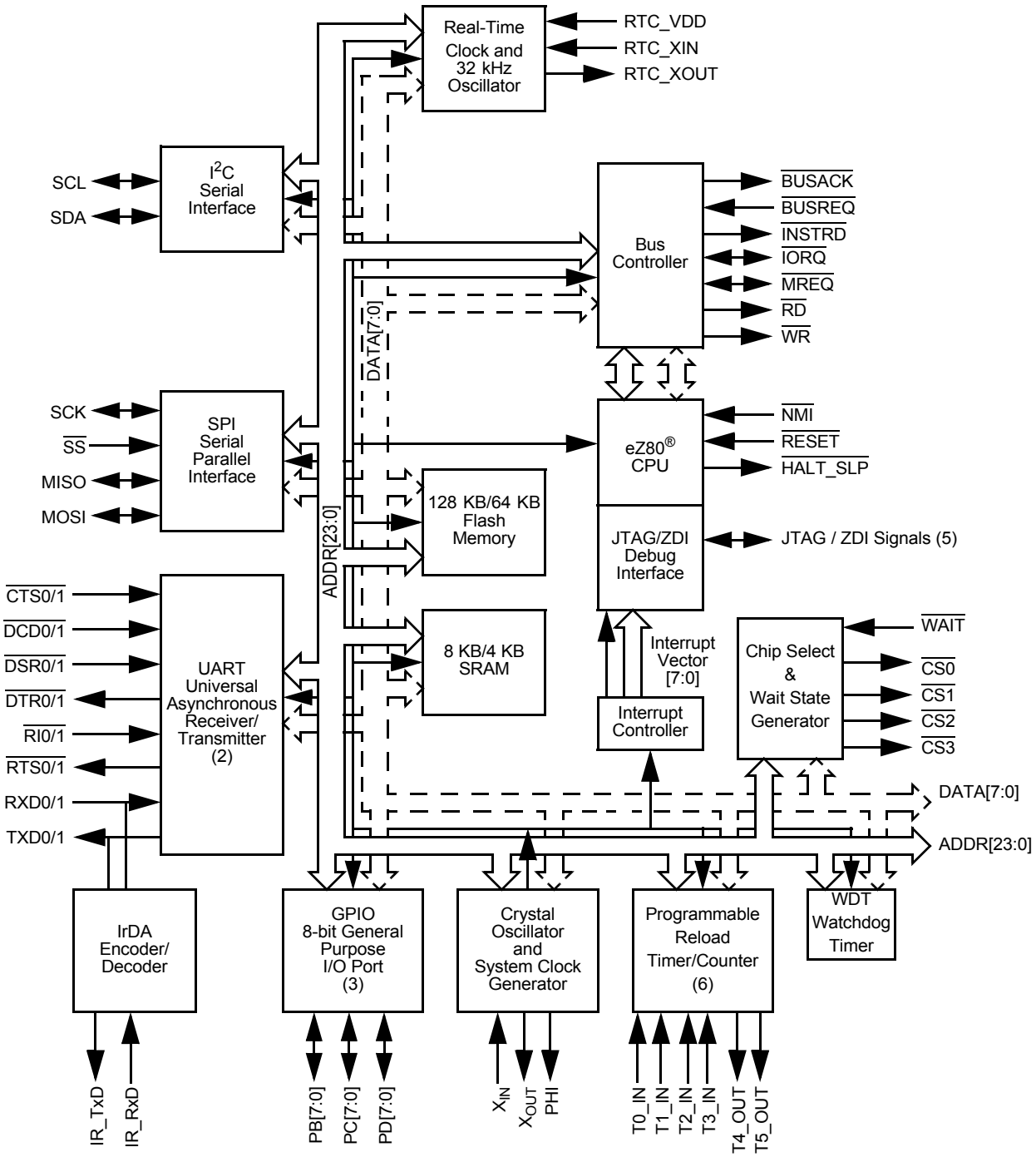

Figure 1.eZ80F92 Block Diagram

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

Bit Position	Value	Description
7 PHI_OFF	1	PHI Clock output is disabled (output is high-impedance).
	0	PHI Clock output is enabled.
6	0	Reserved.
5 PRT5_OFF	1	System clock to PRT5 is powered down.
	0	System clock to PRT5 is powered up.
4 PRT4_OFF	1	System clock to PRT4 is powered down.
	0	System clock to PRT4 is powered up.
3 PRT3_OFF	1	System clock to PRT3 is powered down.
	0	System clock to PRT3 is powered up.
2 PRT2_OFF	1	System clock to PRT2 is powered down.
	0	System clock to PRT2 is powered up.
1 PRT1_OFF	1	System clock to PRT1 is powered down.
	0	System clock to PRT1 is powered up.
0 PRT0_OFF	1	System clock to PRT0 is powered down.
	0	System clock to PRT0 is powered up.

Chip Selects and Wait States

The eZ80F92 device generates four Chip Selects for external devices. Each Chip Select may be programmed to access either memory space or I/O space. The Memory Chip Selects can be individually programmed on a 64 KB boundary. The I/O Chip Selects can each choose a 256-byte section of I/O space. In addition, each Chip Select may be programmed for up to 7 wait states.

Memory and I/O Chip Selects

Each of the Chip Selects can be enabled for either the memory address space or the I/O address space, but not both. To select the memory address space for a particular Chip Select, CSX_IO (CS*x*_CTL[4]) must be reset to 0. To select the I/O address space for a particular Chip Select, CSX_IO must be set to 1. After RESET, the default is for all Chip Selects to be configured for the memory address space. For either the memory address space or the I/O address space, the individual Chip Selects must be enabled by setting CSx_EN (CS*x*_CTL[3]) to 1.

Memory Chip Select Operation

Operation of each of the Memory Chip Selects is controlled by three control registers. To enable a particular Memory Chip Select, the following conditions must be met:

- The Chip Select is enabled by setting CSx_EN to 1
- The Chip Select is configured for Memory by clearing CSX_IO to 0
- The address is in the associated Chip Select range:

  CS*x*_LBR[7:0] ≤ ADDR[23:16] ≤ CS*x*_UBR[7:0]
- No higher priority (lower number) Chip Select meets the above conditions
- A memory access instruction must be executing

If all of the foregoing conditions are met to generate a Memory Chip Select, then the following actions occur:

- The appropriate Chip Select—$\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, or $\overline{CS3}$—is asserted (driven Low)
- $\overline{MREQ}$ is asserted (driven Low)
- Depending upon the instruction, either $\overline{RD}$ or $\overline{WR}$ is asserted (driven Low)

If the upper and lower bounds are set to the same value (CS*x*_UBR = CS*x*_LBR), then a particular Chip Select is valid for a single 64 KB page.

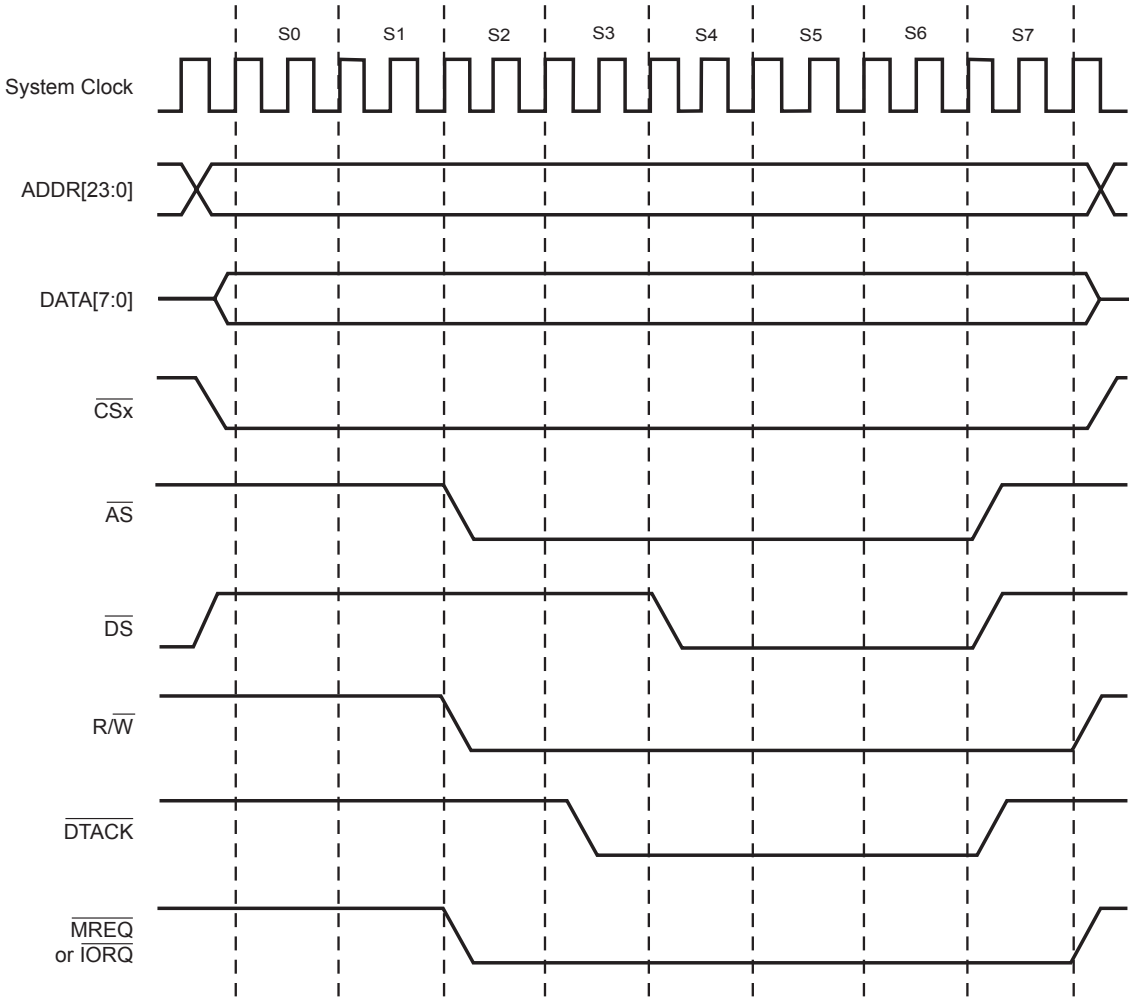

Figure 18. Example: Motorola Bus Mode Write Timing

Switching Between Bus Modes

Each time the bus mode controller must switch from one bus mode to another, there is a one-cycle CPU system clock delay. An extra clock cycle is not required for repeated access in any of the bus modes; nor is it required when the eZ80F92 device switches to eZ80® bus mode. The extra clock cycles are not shown in the timing examples. Due to the asynchronous nature of these bus protocols, the extra delay does not impact peripheral communication.

Chip Select *x* Upper Bound Register

For Memory Chip Selects, the Chip Select *x* Upper Bound registers, listed in Table 23, defines the upper bound of the address range for which the corresponding Chip Select (if enabled) can be active. For I/O Chip Selects, this register produces no effect. The reset state for the Chip Select 0 Upper Bound register is FFh, while the reset state for the other Chip Select upper bound registers is 00h.

Table 23. Chip Select x Upper Bound Register(CS0_UBR = 00A9h, CS1_UBR = 00ACh, CS2_UBR = 00AFh, CS3_UBR = 00B2h)

Bit	7	6	5	4	3	2	1	0
CS0_UBR Reset	1	1	1	1	1	1	1	1
CS1_UBR Reset	0	0	0	0	0	0	0	0
CS2_UBR Reset	0	0	0	0	0	0	0	0
CS3_UBR 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] CSx_UBR	00h–FFh	For Memory Chip Selects (CSx_IO = 0) This byte specifies the upper bound of the Chip Select address range. The upper byte of the address bus, ADDR[23:16], is compared to the values contained in these registers for determining whether a Chip Select signal should be generated.
		For I/O Chip Selects (CSx_IO = 1) No effect.

If the NMI_OUT bit in the WDT_CTL register is set to 1, then upon time-out, the WDT asserts an NMI for CPU processing. The NMI_FLAG bit can be polled by the CPU to determine the source of the NMI event.

Watchdog Timer Registers

Watchdog Timer Control Register

The Watchdog Timer Control register, listed in Table 27, is an 8-bit Read/Write register used to enable the Watchdog Timer, set the time-out period, indicate the source of the most recent RESET, and select the required operation upon WDT time-out.

Table 27. Watchdog Timer Control Register; (WDT_CTL = 0093h)

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0/1	0	0	0	0	0
CPU Access	R/W	R/W	R	R/W	R/W	R	R/W	R/W
Note: R = Read only; R/W = Read/Write.								

Bit Position	Value	Description
7 WDT_EN	0	WDT is disabled.
	1	WDT is enabled. When enabled, the WDT cannot be disabled without a RESET.
6 NMI_OUT	0	WDT time-out resets the CPU.
	1	WDT time-out generates a nonmaskable interrupt (NMI) to the CPU.
5 RST_FLAG*	0	RESET caused by external full-chip reset or ZDI reset.
	1	RESET caused by WDT time-out. This flag is set by the WDT time-out, even if the NMI_OUT flag is set to 1. The CPU can poll this bit to determine the source of the RESET or NMI.
[4:3] WDT_CLK	00	WDT clock source is system clock.
	01	WDT clock source is Real-Time Clock source (32 kHz on-chip oscillator or 50/60 Hz input as set by RTC_CTRL[4]).
	10	Reserved.
	11	Reserved.
2	0	Reserved.

Note: *RST_FLAG is only cleared by a non-WDT RESET.

Timer Data Register—Low Byte

This Read Only register returns the Low byte of the current count value of the selected timer. The Timer Data Register—Low Byte, listed in Table 33, can be read while 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, {TMRx_DR_H[7:0], TMRx_DR_L[7:0]}, first read the Timer Data Register—Low Byte and then read the Timer Data Register—High Byte. The Timer Data Register—High Byte value is latched when a Read of the Timer Data Register—Low Byte occurs.

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

Table 33. Timer Data Register—Low Byte(TMR0_DR_L = 0081h, TMR1_DR_L = 0084h, TMR2_DR_L = 0087h, TMR3_DR_L = 008Ah, TMR4_DR_L = 008Dh, or TMR5_DR_L = 0090h)

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] TMRx_DR_L	00h–FFh	These bits represent the Low byte of the 2-byte timer data value, {TMRx_DR_H[7:0], TMRx_DR_L[7:0]}. Bit 7 is bit 7 of the 16-bit timer data value. Bit 0 is bit 0 (lsb) of the 16-bit timer data value.

Timer Data Register—High Byte

This Read Only register returns the High byte of the current count value of the selected timer. The Timer Data Register—High Byte, listed in Table 34 on page 84, can be read while 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, {TMRx_DR_H[7:0], TMRx_DR_L[7:0]}, first read the Timer Data Register—Low Byte and then read the Timer Data Register—High Byte. The Timer Data Register—High Byte value is latched when a Read of the Timer Data Register—Low Byte occurs.

Note: *The timer data registers and timer reload registers share the same address space.*

Table 67. IrDA Physical Layer 1.4 Pulse Durations Specifications (Continued)

Baud Rate	Minimum Pulse Width	Maximum Pulse Width
57600	1.41 s	4.34 s
115200	1.41 s	2.23 s

Receiver Frequency Divider

The IrDA receiver uses a 6-bit frequency divider. The value is derived from the system clock to measure IR_RxD pulses. The IrDA endec detects pulses that are within the IrDA Physical Layer specified minimum and maximum ranges, with system clock frequencies from 5 MHz up to 50 MHz.

The upper four bits of the frequency divider factor are set via the FREQ_DIV bit in the IR_CTL register, based on the following equation:

$$\text{Frequency Divider Factor} = \frac{\text{System Clock Frequency (MHz)}}{\text{Target Frequency of 3.33 MHz}}$$

The remaining lower two bits of the divider are set to 03h. The target frequency corresponds to a period of 1.2 seconds. The FREQ_DIV value must be rounded to the nearest integer and the resulting period of the 6-bit frequency divider must not be larger than 1.4 seconds, which is the IrDA defined minimum pulse width. If the period is greater than 1.4 seconds, FREQ_DIV should be rounded to the next lower integer. The receiver frequency divider value versus the system clock frequency is shown in table, below.

Table 68. Frequency Divider Values

System Clock	FREQ_DIV
< 5.0 MHz	00h*
5.0–7.8 MHz	01h
7.8–10.8 MHz	02h
10.8–13.6 MHz	03h
13.6–25 MHz	FLOOR[4-bit Frequency Divider Factor]
25–50 MHz	ROUND[4-bit Frequency Divider Factor]

Note: *The frequency divider is disabled when set to 00h.

Table 71. SPI Clock Phase and Clock Polarity Operation

CPHA	CPOL	SCK Transmit Edge	SCK Receive Edge	SCK Idle State	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

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 resulting implication is a full-duplex transmission, with both data out and data in synchronized with the same clock signal. Thus the byte transmitted is replaced by the byte received and eliminates the requirement for separate transmit-empty and receive-full status bits. A single status bit, SPIF, is used to signify that the I/O operation is completed, see the SPI Status Register (SPI_SR) on page 137.

The SPI is double-buffered on Read, but not on Write. 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 SPIF flag of the SPI_SR register is set.

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 8 bits of transmit data out the MOSI pin while shifting in 8 bits of data on the MISO pin. After transfer, the SCK signal idles.

In SPI SLAVE mode, the start logic receives a logic Low from the $\overline{SS}$ pin and a clock input at the SCK pin, and the slave is synchronized to the master. Data from the master is received serially from the slave MOSI signal and loads 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, then 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 $\overline{SS}$ pin signal goes Low and 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 and the transfer ends when the SPIF flag gets set.

Table 79. I²C 10-Bit Master Transmit Status Codes

Code	I²C State	Microcontroller Response	Next I²C Action
38h	Arbitration lost	Clear IFLG	Return to idle
		Or set STA, clear IFLG	Transmit START when bus 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
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
D0h	Second Address byte + W transmitted, ACK received	Write byte to DATA, clear IFLG	Transmit data byte, receive ACK
		Or set STA, clear IFLG	Transmit repeated START
		Or set STP, clear IFLG	Transmit STOP
		Or set STA & STP, clear IFLG	Transmit STOP then START
D8h	Second Address byte + W transmitted, ACK not received	Same as code D0h	Same as code D0h

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

After each data byte is transmitted, the IFLG is 1 and one of the status codes listed in Table 80 is in the I2C_SR register.

Table 80. I²C Master Transmit Status Codes For Data Bytes

Code	I²C State	Microcontroller Response	Next I²C Action
28h	Data byte transmitted, ACK received	Write byte to DATA, clear IFLG	Transmit data byte, receive ACK
		Or set STA, clear IFLG	Transmit repeated START
		Or set STP, clear IFLG	Transmit STOP
		Or set STA & STP, clear IFLG	Transmit START then STOP

I²C Clock Control Register

The I2C_CCR register is a Write Only register. The seven LSBs control the frequency at which the I²C bus is sampled and the frequency of the I²C clock line (SCL) when the I²C is in MASTER mode. The Write Only I2C_CCR registers share the same I/O addresses as the Read Only I2C_SR registers. See Table 90.

Table 90. I²C Clock Control Registers(I2C_CCR = 00CCh)

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 = Read only.								

Bit Position	Value	Description
7	0	Reserved.
[6:3] M	0000–1111	I²C clock divider scalar value.
[2:0] N	000–111	I²C clock divider exponent.

The I²C clocks are derived from the CPU system clock. The frequency of the CPU system clock is f_{SCK}. The I²C bus is sampled by the I²C block at the frequency f_{SAMP} supplied by:

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

In MASTER mode, the I²C clock output frequency on SCL (f_{SCL}) is supplied by:

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

The use of two separately-programmable dividers allows the MASTER mode output frequency to be set independently of the frequency at which the I²C bus is sampled. This feature is particularly useful in multimaster systems because the frequency at which the I²C bus is sampled must be at least 10 times the frequency of the fastest master on the bus to ensure that START and STOP conditions are always detected. By using two programmable clock divider stages, a high sampling frequency can be ensured while allowing the MASTER mode output to be set to a lower frequency.

ZDI Block Write

The Block Write operation is initiated in the same manner as the single-byte Write operation, but instead of terminating the Write operation after the first data byte is transferred, the ZDI master can continue to transmit additional bytes of data to the ZDI slave on the eZ80F92 device. After the receipt of each byte of data the ZDI register address increments by 1. If the ZDI register address reaches the end of the Write Only ZDI register address space (30h), the address stops incrementing. Figure 43 displays the timing for ZDI Block Write operations.

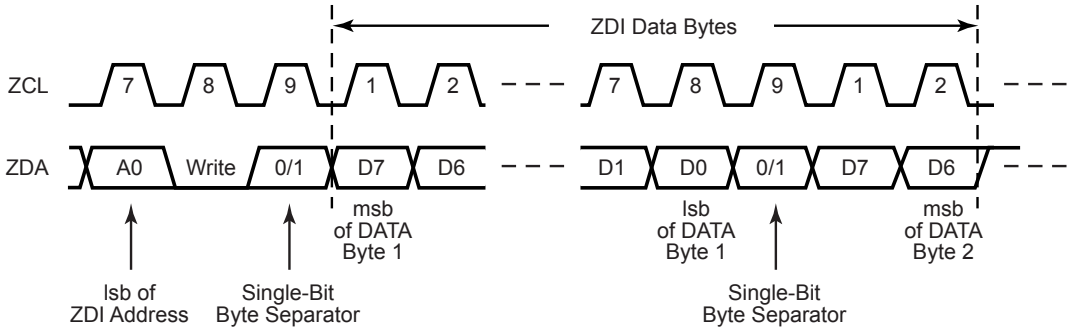

Figure 43.ZDI Block Data Write Timing

ZDI Read Operations

ZDI Single-Byte Read

Single-byte Read operations are initiated in the same manner as single-byte Write operations, with the exception that the R/$\overline{W}$ bit of the ZDI register address is set to 1. Upon receipt of a slave address with the R/$\overline{W}$ bit set to 1, the eZ80F92 device's ZDI block loads the selected data into the shifter at the beginning of the first cycle following the single-bit data separator. The msb is shifted out first. Figure 44 displays the timing for ZDI single-byte Read operations.

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

Hex Value	Command	Hex Value	Command
0A	Exchange CPU register sets AF ← AF' BC ← BC' DE ← DE' HL ← HL'	8A	Reserved
0B	Read memory from current PC value, increment PC	8B	Write memory from current PC value, increment PC

The eZ80® CPU's alternate register set (A', F', B', C', D', E', HL') cannot be read directly. The ZDI programmer must execute the exchange instruction (EXX) to gain access to the alternate eZ80 CPU register set.

Table 109. ZDI Read Memory Register(ZDI_RD_MEM = 20h in the ZDI Register Read Only Address Space)

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] ZDI_RD_MEM	00h–FFh	8-bit data read from the memory address indicated by the CPU's program counter. In Z80® MEMORY mode, 8-bit data is transferred out from address {MBASE, PC[15:0]}. In ADL Memory mode, 8-bit data is transferred out from address PC[23:0].

Table 145. DC Characteristics (Continued)

		T_A = 0 ºC to 70 ºC		T_A = 0 ºC to 105 ºC			
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Units**	**Conditions**
I_{RTC}	RTC Supply Current	2.5	10 Typical	2.5	10 Typical	µA	Supply current into RTC_V_{DD}; SLEEP mode[2].

Notes
1. This condition excludes all pins with on-chip pull-ups when driven Low.
2. RTC current increases when the eZ80F92 device is not in SLEEP mode as the RTC_VDD pin supplies power to system clock buffers within the Real-Time Clock circuit.

POR and VBO Electrical Characteristics

Table 146 lists the Power-On Reset and Voltage Brownout characteristics of the eZ80F92 device.

Table 146. POR and VBO Electrical Characteristics

		T_A = 0 ºC to +105 ºC				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Unit**	**Conditions**
V_{VBO}	VBO Voltage Threshold	2.40	2.55	2.85	V	$V_{CC} = V_{VBO}$
V_{POR}	POR Voltage Threshold	2.45	2.65	2.90	V	$V_{CC} = V_{POR}$
V_{HYST}	POR/VBO Hysteresis	50	100	150	mV	
T_{ANA}	POR/VBO analog RESET duration	40		100	µs	
T_{VBO_MIN}	VBO pulse reject period		10		µs	
V_{CCRAMP}	V_{CC} ramp rate requirements to guarantee proper RESET occurs	0.1		100	V/ms	

Typical Current Consumption Under Various Operating Conditions

In the following pages, Figure 53 on page 225 displays the typical current consumption of the eZ80F92 device versus the number of WAIT states while operating 25 ºC, 3.3 V, and with either a 5 MHz, 10 MHz, 15 MHz, or 20 MHz system clock. Figure 54 on page 226 displays the typical current consumption of the eZ80F92 device versus the system clock frequency while operating 25 ºC, 3.3 V, and using 0, 2, or 7 WAIT states. Figure 55 on page 227 displays the typical current consumption of the eZ80F92 device versus temperature while operating at 3.3 V, 7 WAIT states, and with either a 5 MHz, 10 MHz, 15 MHz or 20 MHz system clock. Figure 56 on page 228 displays the typical current consumption

Table 149. External Write Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
T_1	Clock Rise to ADDR Valid Delay	—	13
T_2	Clock Rise to ADDR Hold Time	2.0	—
T_3	Clock Fall to Output DATA Valid Delay	—	11
T_4	Clock Rise to DATA Hold Time	2.0	—
T_5	Clock Rise to $\overline{CSx}$ Assertion Delay	2.0	19.0
T_6	Clock Rise to $\overline{CSx}$ Deassertion Delay	2.0	18.0
T_7	Clock Rise to $\overline{MREQ}$ Assertion Delay	2.0	16.0
T_8	Clock Rise to $\overline{MREQ}$ Deassertion Delay	2.0	16.0
T_9	Clock Fall to $\overline{WR}$ Assertion Delay	1.8	6.5
T_{10}	Clock Rise to $\overline{WR}$ Deassertion Delay*	1.6	6.5
	$\overline{WR}$ Deassertion to ADDR Hold Time	0.25	—
	$\overline{WR}$ Deassertion to DATA Hold Time	0.25	—
	$\overline{WR}$ Deassertion to $\overline{CSx}$ Hold Time	0.25	—
	$\overline{WR}$ Deassertion to $\overline{MREQ}$ Hold Time	0.25	—

Note: *At the conclusion of a Write cycle, deassertion of $\overline{WR}$ always occurs before any change to ADDR, DATA, $\overline{CSx}$, or $\overline{MREQ}$.

Zilog Debug Interface Timing

Figure 65 and Table 155 display timing information for TCK, TDI, TDO, TMS pins.

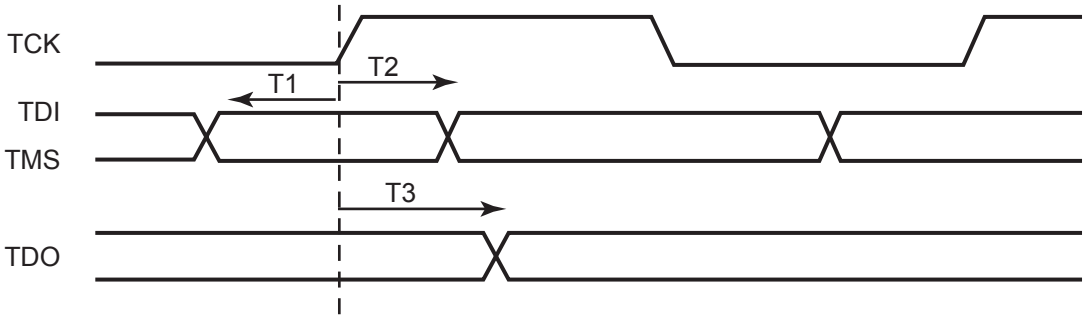

Figure 65.ZDI Timing

Table 155. ZDI Timing Specifications

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
T_{TCK}	TCK Period	2 x T_{XIN}	
T_1	TDI/TMS setup to TCK Rise	4	
T_2	TDI/TMS hold after TCK Rise Fall	4	
T_3	TCK Rise to TDO change		10

D

E