Zilog - EZ80190AZ050EC00TR Datasheet

Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	eZ80
Number of Cores/Bus Width	1 Core, 8-Bit
Speed	50MHz
Co-Processors/DSP	-
RAM Controllers	-
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 105°C (TA)
Security Features	-
Package / Case	100-LQFP
Supplier Device Package	100-VQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80190az050ec00tr

Table 1. 100-Pin LQFP Pin Identification of the eZ80190 Device (Continued)

Pin No.	Symbol	Function	Signal Direction	Description
89	PA0	GPIO Port A	Input/Output	The PA0 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.
90	PA1	GPIO Port A	Input/Output	The PA1 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.
91	PA2	GPIO Port A	Input/Output	The PA2 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.
92	PA3	GPIO Port A	Input/Output	The PA3 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.
93	PA4	GPIO Port A	Input/Output	The PA4 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.
94	PA5	GPIO Port A	Input/Output	The PA5 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.

Table 1. 100-Pin LQFP Pin Identification of the eZ80190 Device (Continued)

Pin No.	Symbol	Function	Signal Direction	Description
95	PA6	GPIO Port A	Input/Output	The PA6 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.
96	PA7	GPIO Port A	Input/Output	The PA7 pin can be used for GPIO. It can be individually programmed as an input or an output and can also be used individually as an interrupt input. Each Port A pin, when programmed as an output, can be selected to be an open-drain or open-source output.
97	V_{DD}	Power Supply		Power Supply
98	GND	Ground		Ground
99	BUSREQ	Bus Request	Input, Active Low	External devices can force the eZ80190 device to release the bus for their use by driving this line Low. To the CPU, the bus request signal can also originate from internal DMA controllers. In such cases, bus requests from the DMA controllers have a higher priority than a request from an external bus master.
100	PHI	System Clock	Output	The PHI pin is an output driven by the internal system clock. It can be used by the system for synchronization with the eZ80190 device.

Register Map

All on-chip peripheral registers are accessed in the I/O address space. All I/O operations employ 16-bit addresses. The upper byte of the 24-bit address bus is forced to 00h (ADDR[23:16] = 00h) during all I/O operations. All I/O operations using 16-bit addresses within the range of 80h to FFh are routed to the on-chip peripherals; where xx is any value from 00h to FFh. External I/O Chip Selects are not generated if the address space programmed for the I/O Chip Selects overlap the 80h to FFh address range.

Note: Registers at unused addresses within the 80h to FFh range assigned to on-chip peripherals are not implemented. READ access to such addresses return unpredictable values and WRITE access produces no effect. Table 2 lists the register map for the eZ80190 device.

Table 2. Register Map

Address (hex)	Mnemonic	Name	Reset (hex)	CPU Access	Page No.
Programmable Reload Counter/Timers					
80	TMR0_CTL	Timer 0 Control Register	00	R/W	35
81	TMR0_DR_L	Timer 0 Data Register—Low Byte	00	R	36
	TMR0_RR_L	Timer 0 Reload Register—Low Byte	00	W	37
82	TMR0_DR_H	Timer 0 Data Register—High Byte	00	R	36
	TMR0_RR_H	Timer 0 Reload Register—High Byte	00	W	38
83	TMR1_CTL	Timer 1 Control Register	00	R/W	35
84	TMR1_DR_L	Timer 1 Data Register—Low Byte	00	R	36
	TMR1_RR_L	Timer 1 Reload Register—Low Byte	00	W	37
85	TMR1_DR_H	Timer 1 Data Register—High Byte	00	R	36
	TMR1_RR_H	Timer 1 Reload Register—High Byte	00	W	38
86	TMR2_CTL	Timer 2 Control Register	00	R/W	35
87	TMR2_DR_L	Timer 2 Data Register—Low Byte	00	R	36
	TMR2_RR_L	Timer 2 Reload Register—Low Byte	00	W	37
88	TMR2_DR_H	Timer 2 Data Register—High Byte	00	R	36
	TMR2_RR_H	Timer 2 Reload Register—High Byte	00	W	38
89	TMR3_CTL	Timer 3 Control Register	00	R/W	35

When the Bus Enable bit (ENAB) is set to 0, the I^2C bus inputs SCLx. SDAx is ignored and the I^2C module does not respond to any address on the bus. When ENAB is set to 1, the I^2C responds to calls to its slave address and to the general call address if the GCE bit (I2Cx_SAR[0]) is set to 1.

When the MASTER Mode Start bit (STA) is set to 1, the I^2C enters MASTER mode and sends a START condition on the bus when the bus is free. If the STA bit is set to 1 when the I^2C module is already in MASTER mode and one or more bytes are transmitted, then a repeated START condition is sent. If the STA bit is set to 1 when the I^2C block is being accessed in SLAVE mode, the I^2C completes the data transfer in SLAVE mode and then enters MASTER mode when the bus is released. The STA bit is automatically cleared after a START condition is set. Writing a 0 to this bit produces no effect.

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 behaves 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) and then transmits the START condition. The STP bit is cleared automatically. Writing a 0 to this bit produces no effect.

The I^2C Interrupt Flag (IFLG) is set to 1 automatically when the device enters any of 30 of the possible 31 I^2C states. 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 to 1, 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 (ACK) is set to 1, an acknowledgement 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 I2Cx_SAR is set to 1
- A data byte is received in MASTER or SLAVE mode

When ACK is cleared to 0, a NACK is sent when a data byte is received in MASTER or SLAVE mode. If ACK is cleared to 0 in SLAVE TRANSMIT mode, the byte in the I2Cx_DR register is presumed to be the last byte. After this byte is transmitted, the I^2C block enters state `C8h`, then returns to the IDLE state. The I^2C module does not respond to its slave address unless ACK is set.

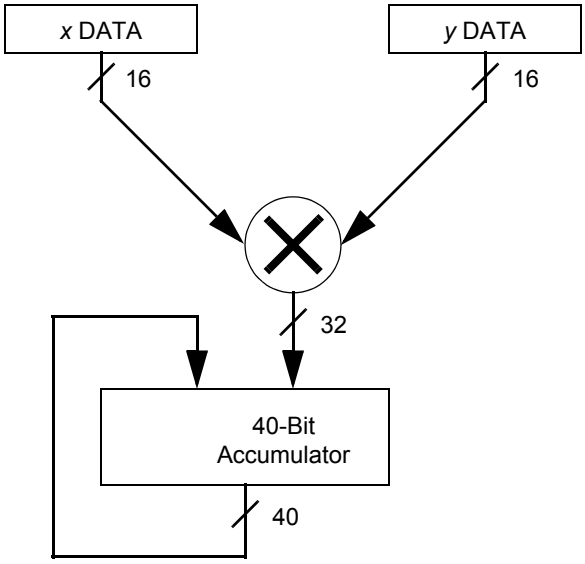


Figure 21. Multiply-Accumulator Block Diagram

Multiply-Accumulator Basic Operation

Figure 22 on page 115 displays a simplified view of the state progression of the MACC when performing calculation on a set of data. The progression begins in the upper left corner with a DATA bank containing the value EMPTY. The CPU loads the MACC control registers to define the next MACC calculation.

If the MACC is not busy with an existing calculation (EMPTY or DONE), the DATA and CALC banks are immediately swapped to initiate the new calculation. If the MACC is busy with an existing calculation, the DATA bank status changes to READY and waits for the MACC to complete the existing calculation. Then, the DATA and CALC banks are swapped to initiate the new calculation.

Assuming the DATA bank is EMPTY or READY when the MACC completes the new calculation, the CALC bank is swapped with the DATA bank. The CPU can then retrieve the result of the new calculation from the accumulator.

Table 67. MACC Control Register (MACC_CTL = E7h)

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 MACC_IE	0	MACC interrupt is disabled.
	1	The MACC interrupt is enabled for the calculation currently being defined. The MACC generates an interrupt request to the CPU when it completes this calculation (DONE).
6 NOISE	0	All NOISE bits added to the accumulator using IN_SHIFT are 0.
	1	All NOISE bits added to the accumulator using IN_SHIFT are 1.
[5:3] OUT_SHIFT	000	No right-shift is performed during READs from the MACC Accumulator registers by the CPU. DATA_OUT[40:0] = MACC_ACx[39:0].
	001	Reads from the MACC Accumulator registers by the CPU are right-shifted by 1 bit with a fill by the sign bit (msb = bit 39). DATA_OUT[40:0] = {2{MACC_ACx[39]}, MACC_ACx[38:1]}.
	010	Reads from the MACC Accumulator registers by the CPU are right-shifted by 2 bits with a fill by the sign bit (msb = bit 39). DATA_OUT[40:0] = {3{MACC_ACx[39]}, MACC_ACx[38:2]}.
	011	Reads from the MACC Accumulator registers by the CPU are right-shifted by 3 bits with a fill by the sign bit (msb = bit 39). DATA_OUT[40:0] = {4{MACC_ACx[39]}, MACC_ACx[38:3]}.
	100	Reads from the MACC Accumulator registers by the CPU are right-shifted by 4 bits with a fill by the sign bit (msb = bit 39). DATA_OUT[40:0] = {5{MACC_ACx[39]}, MACC_ACx[38:4]}.
	101	Reads from the MACC Accumulator registers by the CPU are right-shifted by 5 bits with a fill by the sign bit (msb = bit 39). DATA_OUT[40:0] = {6{MACC_ACx[39]}, MACC_ACx[38:5]}.
	110	Reads from the MACC Accumulator registers by the CPU are right-shifted by 6 bits with a fill by the sign bit (msb = bit 39). DATA_OUT[40:0] = {7{MACC_ACx[39]}, MACC_ACx[38:6]}.
	111	Reads from the MACC Accumulator registers by the CPU are right-shifted by 7 bits with a fill by the sign bit (msb = bit 39). DATA_OUT[40:0] = {8{MACC_ACx[39]}, MACC_ACx[38:7]}.

Bit Position	Value	Description
[2:0] IN_SHIFT	000	No left-shift is performed during writes to the MACC Accumulator registers by the CPU. MACC_ACx[39:0] = DATA_IN[39:0]
	001	Writes to the MACC Accumulator registers by the CPU are left-shifted by 1 bit with 1 NOISE bit filling the lsb. MACC_ACx[39:0] = {DATA_IN[38:0], NOISE}
	010	Writes to the MACC Accumulator registers by the CPU are left-shifted by 2 bits with repeated NOISE bits filling the lsbs. MACC_ACx[39:0] = {DATA_IN[37:0], 2{NOISE}}
	011	Writes to the MACC Accumulator registers by the CPU are left-shifted by 3 bits with repeated NOISE bits filling the lsbs. MACC_ACx[39:0] = {DATA_IN[36:0], 3{NOISE}}
	100	Writes to the MACC Accumulator registers by the CPU are left-shifted by 4 bits with repeated NOISE bits filling the lsbs. MACC_ACx[39:0] = {DATA_IN[35:0], 4{NOISE}}
	101	Writes to the MACC Accumulator registers by the CPU are left-shifted by 5 bits with repeated NOISE bits filling the lsbs. MACC_ACx[39:0] = {DATA_IN[34:0], 5{NOISE}}
	110	Writes to the MACC Accumulator registers by the CPU are left-shifted by 6 bits with repeated NOISE bits filling the lsbs. MACC_ACx[39:0] = {DATA_IN[33:0], 6{NOISE}}
	111	Writes to the MACC Accumulator registers by the CPU are left-shifted by 7 bits with repeated NOISE bits filling the lsbs. MACC_ACx[39:0] = {DATA_IN[32:0], 7{NOISE}}

MACC Accumulator Byte 0 Register

The MACC_AC0 register, listed in Table 68 on page 132, contains the LSB (bits 7:0) of the 40-bit MACC Accumulator.

Table 68. MACC Accumulator Byte 0 Register (MACC_AC0 = E8h)

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 = Undefined; R/W = Read/Write.

MACC Status Register

The MACC_STAT register, listed in Table 73, reflects the current status of the Multiply-Accumulator. Writing a value of 80h to the MACC_STAT register when the CALC bank has completed its calculation (DONE) and the DATA register is not loaded with a new calculation (EMPTY) swaps the banks to allow the pending result to be retrieved.

The eZ80190 device uses two distinct numbered banks, banks 0 and 1. The value in bit 4 of the MACC_STAT register indicates which of these two banks is currently accessible as the DATA bank. In general, there is no requirement for software to monitor which numbered bank is currently the DATA bank and which is the CALC bank.

Table 73. MACC Status Register (MACC_STAT = EDh)

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

Note: R = Read; W = Write.

Bit Position	Value	Description
[7:5]	000	Reserved.
4 BANK	0	The current DATA bank is Bank 0. The current DATA bank is always reset to Bank 0 when both banks are EMPTY.
	1	The current CALC bank is Bank 1.
[3:2] CALC_STAT	00	The CALC bank is EMPTY. No calculation is set up for execution.
	01	Invalid.
	10	The CALC bank is IN PROGRESS. The MACC is currently executing on a data set.
	11	The CALC bank is DONE. The MACC has completed execution of the operations defined by the CALC bank control registers. The result is stored in the CALC bank accumulator registers. The CALC bank must be swapped with the DATA bank to allow the CPU to access the result.

Bit Position	Value	Description
[7:0] DMAx_SAR_L, DMAx_SAR_H, or DMAx_SAR_U	00h–FFh	The 2 sets of DMA Source address registers contain the memory location addresses for the source of the data transfer. The 24-bit addresses are returned by {DMAx_SAR_U, DMAx_SAR_H, DMAx_SAR_L}, where x is either 0 or 1.

DMA Destination Address Registers

This group of registers holds the 24-bit address of the current destination memory location. Depending upon settings within the DMA Control registers' DMA_CTL fields, the 24-bit address values can automatically be incremented, decremented, or unchanged following transfer of each byte of data. See Table 78.

Table 78. DMA Destination Address Registers DMA0_DAR_L = F1h, DMA0_DAR_H = F2h, DMA0_DAR_U = F3h, DMA1_DAR_L = FAh, DMA1_DAR_H = FBh, DMA1_DAR_U = FCh

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 = Undefined; R/W = Read/Write.

Bit Position	Value	Description
[7:0] DMAX_DAR_L, DMAX_DAR_H, or DMAX_DAR_U	00h–FFh	The 2 sets of DMA Destination address registers contain the memory location addresses for the destination of the data transfer. The 24-bit addresses are returned by {DMAx_DAR_U, DMAx_DAR_H, DMAx_DAR_L} where x is either 0 or 1.

DMA Byte Count Registers

The two pairs of DMA Byte Count registers, listed in Table 79 on page 145, contain the number of bytes to be transferred by the DMA channels. The 16-bit value, {DMAx_BC_H, DMAx_BC_L}, is decremented after each transfer. The DMA transfer is complete when the value decrements to 0000h. One to 65535 bytes can be transferred.

ZDI Single-Bit Byte Separator

Following each 8-bit ZDI data transfer, a single-bit byte separator is used. The ZDA pin should be forced High (1) prior to the ZCL rising edge for this 9th bit. For most ZDI operations, the ZDI register address automatically increments during this single-bit byte separator period. The same read or write operation as just completed can then be immediately performed on the next ZDI register. If a different operation or register address is required, a ZDI START signal during the byte separator bit can be used to terminate the previous read or write operation and signify initiation of a new ZDI register operation.

ZDI Register Addressing

Following a START signal, the ZDI master must output the ZDI register address. All data transfers with the ZDI block use special ZDI registers. The ZDI control registers that reside in the ZDI register address space should not be confused with the eZ80190 device peripheral registers that reside in the I/O addressing space of the eZ80190 device.

Many locations in the ZDI control register address space are shared by two registers, one for Read Only access and one for Write Only access. As an example, a read from ZDI register address `00h` returns the Product ID Low Byte while a write to this same location, `00h`, stores the Low byte of one of the address match values used for generating break points.

The format for a ZDI address is seven bits of address, followed by one bit for read or write control, and completed by a single-bit byte separator in which ZDA must be 1. The data separator time period is used to allow the ZDI master to send a new ZDI START signal, if necessary. The ZDI executes a read or write operation depending on the state of the R/W bit (0 = write, 1 = read). Figure 30 displays the timing for address writes to ZDI registers.

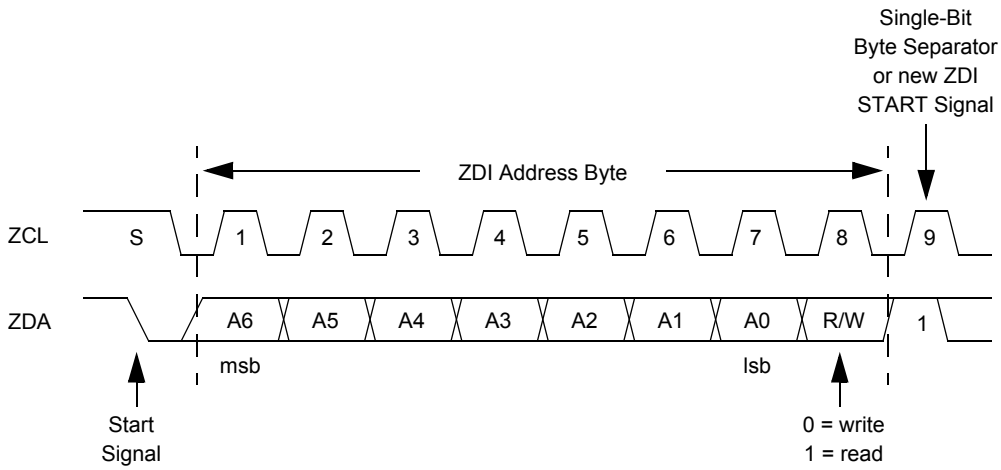


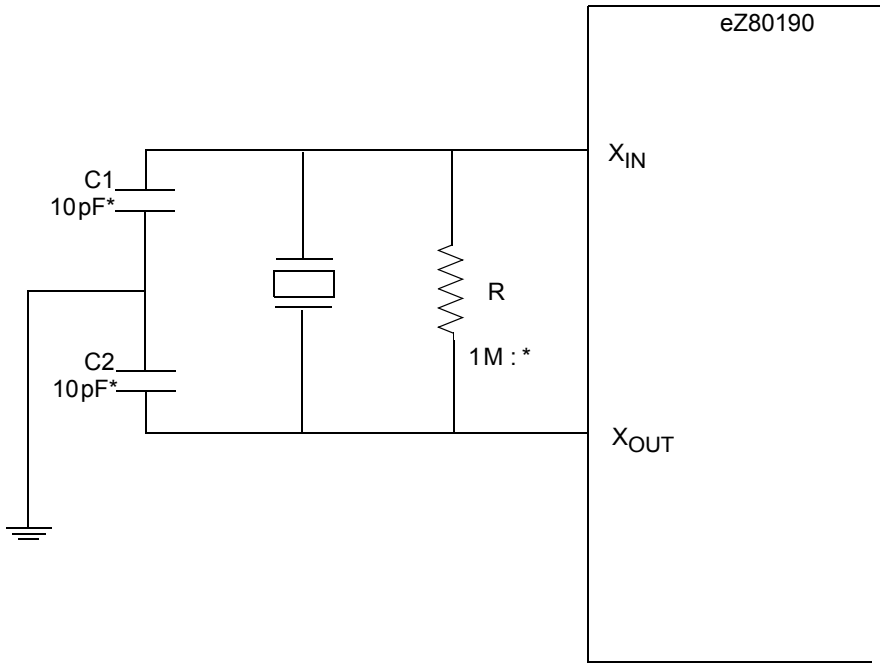
Figure 30. ZDI Address Write Timing

Crystal Oscillator

The eZ80190 device features an on-chip crystal oscillator that supplies clocks to the internal eZ80® CPU core, to peripherals, and to the external pin. The clock circuitry uses the three dedicated pins X_{IN}, X_{OUT}, and PHI.

The external clock/oscillator (X_{IN}) input features two clock-generation options. X_{IN} may be used to interface the internal oscillator to an external oscillator (see Figure 35). Typical circuit parameters are C1 = C2 = 10 pF and R = 1 M : using a a parallel resonant crystal.

X_{IN} can also accept a CMOS-level clock input. The oscillator output (X_{OUT}) connects the internal crystal oscillator to an external crystal oscillator. If an external clock is used, X_{OUT} should be left unconnected. The PHI pin, which drives the high-speed system clock, may be used to synchronize other peripherals to the eZ80190 device system clock.



Note: *These values are typical values only. Actual values must be tuned for the crystal and the frequency of operation.

Figure 35. Crystal Oscillator

DC Characteristics

Table 113 lists the Direct Current characteristics of the eZ80190 device.

Table 113. DC Characteristics

		Standard Temperature Range = 0 °C to 70 °C		Extended Temperature Range = –40 °Cto 105 °C			
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Units**	**Conditions**
V_{DD}	Supply Voltage	3.0	3.6	3.0	3.6	V	
V_{IL}	Low Level Input Voltage	–0.3	0.8V	–0.3	0.8V	V	
V_{IH}	High Level Input Voltage	$0.7xV_{DD}$	5.5	$0.7xV_{DD}$	5.5	V	
V_{OL}	Low Level Output Voltage		0.4		0.4	V	V_{DD} = 3.0 V; I_{OL} = 1 mA
V_{OH}	High Level Output Voltage	2.4		2.4		V	V_{DD} = 3.0 V; I_{OH} = ,— mA
I_{IL}	Input Leakage Current	–10	+10	–10	+10	PA	V_{DD} = 3.6 V; V_{IN} = V_{DD} or V_{SS}*
I_{TL}	Tri-State Leakage Current	–10	+10	–10	+10	PA	V_{DD} = 3.6 V

* This condition excludes the ZDA and ZCL pins, when driven Low, due to the presence of on-chip pull-ups.

In the following pages, Figure 36 displays the typical current consumption of the eZ80190 device versus the number of WAIT states while operating 25 °C, 3.3 V, and with either a 1 MHz or 5 MHz system clock. Figure 37 displays the typical current consumption of the eZ80190 device versus the number of WAIT states while operating 25 °C, 3.3 V, and with either a 20 MHz or 50 MHz system clock. Figure 38 displays the typical current consumption of the eZ80190 device versus the system clock frequency while operating 25 °C, 3.3 V, and using 0, 2, or 7 WAIT states. Figure 39 displays the typical current consumption of the eZ80190 device versus the system clock frequency while operating at 3.3 V, 7 WAIT states, and with either a 5 MHz, 20 MHz, or 50 MHz system clock.

External I/O Read Timing

Figure 42 and Table 117 display the timing for external I/O reads.

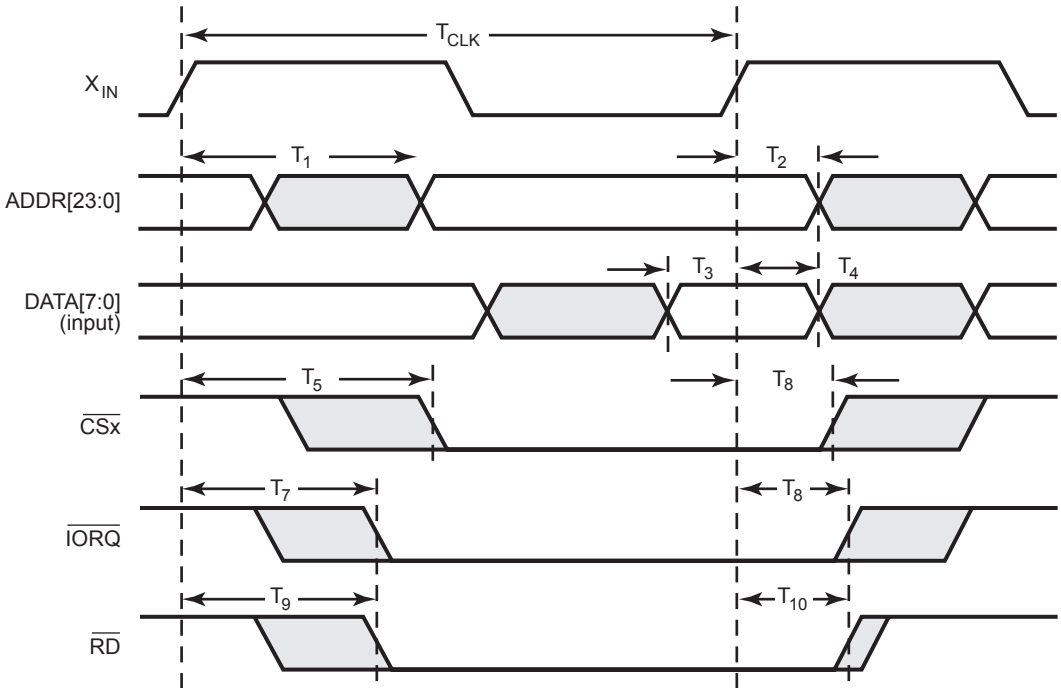


Figure 42. External I/O Read Timing

Table 117. External I/O Read Timing

		Delay (ns)	
Parameter	**Description**	**Min**	**Max**
T_1	Clock Rise to ADDR Valid Delay	—	10.2
T_2	Clock Rise to ADDR Hold Time	1.6	—
T_3	Input DATA Valid to Clock Rise Setup Time	0.0	—
T_4	DATA Hold Time from Clock Rise	5.0	—
T_5	Clock Rise to $\overline{CSx}$ Assertion Delay	3.0	10.5
T_6	Clock Rise to $\overline{CSx}$ Deassertion Delay	3.0	9.7
T_7	Clock Rise to $\overline{IORQ}$ Assertion Delay	2.1	10.3
T_8	Clock Rise to $\overline{IORQ}$ Deassertion Delay	4.1	7.9

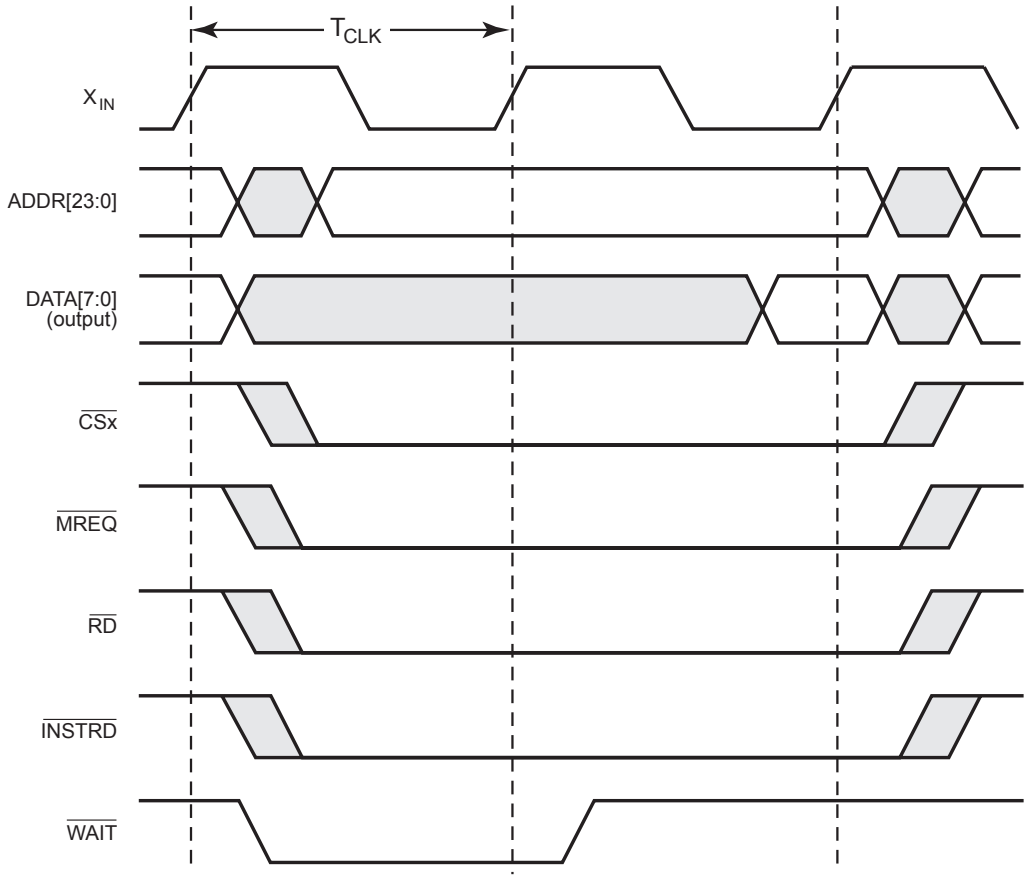


Figure 44. Wait State Timing for Read Operations

Wait State Timing for Write Operations

Figure 45 on page 192 displays the extension of the memory access signals using a single WAIT state for a write operation. The $\overline{\text{WAIT}}$ signal is not delivered to a pin on the eZ80190 device and is displayed here for informational purposes only.