Zilog - EZ80190AZ050SC 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	0°C ~ 70°C (TA)
Security Features	-
Package / Case	100-LQFP
Supplier Device Package	100-VQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/ez80190az050sc

Programmable Reload Timer Operation

Setting Timer Duration

There are three factors to consider when determining Programmable Reload Timer duration—clock frequency, clock divider ratio, and initial count value. Minimum duration of the timer is achieved by loading `0001h`, because the timer times out on the next clock edge. Maximum duration is achieved by loading `0000h`, because the timer rolls over to `FFFFh` on the next clock edge and then continues counting down to `0000h`.

The time-out period of the PRT is returned by the following equation:

$$\text{PRT Time-Out Period} = \frac{\text{Clock Divider Ratio x Reload Value}}{\text{System Clock Frequency}}$$

SINGLE PASS Mode

In SINGLE PASS mode, when the end-of-count value, `0000h`, is reached, counting halts, the timer is disabled, and the PRT_EN bit resets to 0. To restart the timer, the CPU must reenable the timer by setting the PRT_EN bit to 1. Figure 4 displays an example of a PRT operating in SINGLE PASS mode. Timer register information is listed in Table 3 on page 33.

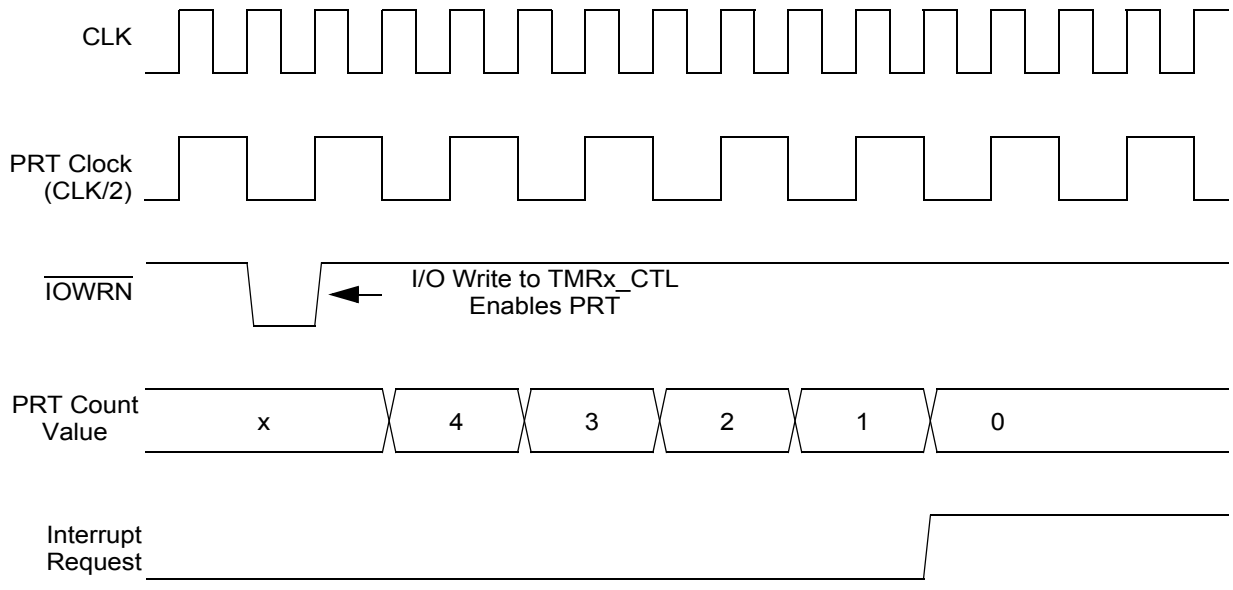

Figure 4. PRT SINGLE PASS Mode Operation Example

The Timer Control register can be read or written to. The timer reload registers are Write Only and are located at the same I/O address as the timer data registers, which are Read Only.

Timer Control Registers

The Timer Control registers, listed in Table 5, are used to control operation of the timer, including enabling the timer, selecting the clock divider, enabling the interrupt, selecting between CONTINUOUS and SINGLE PASS modes, and enabling the automatic reload feature.

Table 5. Timer Control Register (TMR0_CTL = 0080h, TMR1_CTL = 0083h, TMR2_CTL = 0086h, TMR3_CTL = 0089h, TMR4_CTL = 008Ch, TMR5_CTL = 008Fh)

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

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

Bit Position	Value	Description
7 PRT_IRQ	0	The timer has not reached its end-of-count value. This bit is reset to 0 every time the TMRx_CTL register is read.
	1	The timer has reached its end-of-count value. If IRQ_EN is set to 1, an interrupt signal is sent to the CPU. This bit remains 1 until the TMRx_CTL register is read.
6 IRQ_EN	0	Timer interrupt requests are disabled.
	1	Timer interrupt requests are enabled.
5	0	Reserved
4 PRT_MODE	0	The timer operates in SINGLE PASS mode. PRT_EN (bit 0) is reset to 0, and counting stops when the end-of-count value is reached.
	1	The timer operates in CONTINUOUS mode. The timer reload value is written to the counter when the end-of-count value is reached.
[3:2] CLK_DIV	00	Clock ÷ 2 is the timer input source.
	01	Clock ÷ 4 is the timer input source.
	10	Clock ÷ 8 is the timer input source.
	11	Clock ÷ 16 is the timer input source.

Table 16. Port x Alternate Registers 2 (PA_ALT2 = 99h, PB_ALT2 = 9Dh, PC_ALT2 = A1h, PD_ALT2 = A5h)

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.

Chip Selects and Wait States

The eZ80190 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 16-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 four available 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, CS_IO (CS*x*_CTL[4]) must be reset to 0. To select the I/O address space for a particular Chip Select, CS_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 CS_EN (CS*x*_CTL[3]) to 1.

Memory Chip Select Operation

Each of the four Memory Chip Selects features three control registers. To enable a particular Memory Chip Select, the following conditions must be met:

- The Chip Select is enabled by setting CS_EN to 1
- The Chip Select is configured for memory by clearing CS_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 three conditions
- No on-chip RAM is configured for the same address space, because on-chip RAM is prioritized higher than all Memory Chip Selects
- 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:

- A Chip Select—$\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, or $\overline{CS3}$—is activated (driven Low)
- The $\overline{MREQ}$ signal is activated (driven Low)

UART Interrupts

There are five different sources of interrupts from the UART. These five sources of interrupts are:

1. Transmitter
2. Receiver (three different interrupts)
3. Modem status

UART Transmitter Interrupt

The transmitter interrupt is generated if there is no data available for transmission. This interrupt can be disabled using the individual interrupt enable bit, or cleared by writing data into the UARTx_THR register.

UART Receiver Interrupts

A receiver interrupt can be generated by three possible events. The first event, a receiver data ready interrupt event, indicates that one or more data bytes were received and are ready to be read. If the FIFO is enabled, and the trigger level is set, then this interrupt is generated if the number of bytes in the receive FIFO is greater than or equal to the trigger level. If the FIFO is not enabled, the interrupt is generated if the receive buffer contains a data byte. This interrupt is cleared by reading the UARTx_RBR.

The second interrupt source is the receiver time-out. A receiver time-out interrupt is generated when there are fewer data bytes in the receive FIFO than the trigger level. There are no READs and writes to or from the receive FIFO for four consecutive byte times. After the receiver time-out interrupt is generated, it is cleared only after it empties the entire receive FIFO.

The first two interrupt sources from the receiver (data ready and time-out) share an interrupt enable bit.

The third source of a receiver interrupt is a line status error indicating an error in byte reception. This error may result from:

- Incorrect received parity
- Incorrect framing (the stop bit is not detected by the receiver at the end of the byte)
- Receiver overrun condition
- A Break Indication being detected on the receive data input

An interrupt due to one of the above conditions is cleared when the UARTx_LSR register is read. In the case of FIFO mode, a line status interrupt is generated only after the received byte with an error reaches the top of the FIFO and is ready to be read.

SPI Registers

There are four registers in the Serial Peripheral Interface which provide control, status, and data storage functions. These registers are called the SPI Control register (SPIx_CTL), SPI Status register (SPIx_SR), SPI Receive Buffer register (SPIx_RBR), and SPI Transmit Shift register (SPIx_TSR), where the x in each register name is either 0 or 1 depending on which UZI device the SPI is located within. The SPI registers are described in this section.

SPI Control Register

The SPI Control Register, listed in Table 39, is used to control and set up the serial peripheral interface.

Table 39. SPI Control Register (SPI0_CTL = B6h, SPI1_CTL = BAh)

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

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

Bit Position	Value	Description
7 IRQ_EN	0	The SPI system interrupt is disabled.
	1	The SPI system interrupt is enabled.
6	0	Reserved—must be 0.
5 SPI_EN	0	The SPI is disabled.
	1	The SPI is enabled.
4 MASTER_EN	0	When enabled, the SPI operates as a slave.
	1	When enabled, the SPI operates as a master.
3 CPOL	0	The master SCK pin idles in a Low (0) state.
	1	The master SCK pin idles in a High (1) state.
2 CPHA	0	$\overline{SS}$ must go High after a transfer of every byte of data.
	1	$\overline{SS}$ can remain Low to transfer any number of data bytes.
[1:0]	00b	Reserved—must be 0.

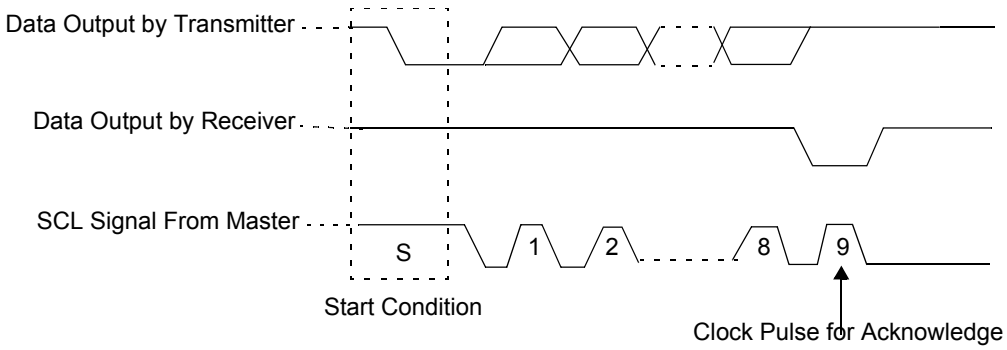

Figure 19. I²C Acknowledge

Clock Synchronization

All masters generate their own clocks on the SCL line to transfer messages on the I²C bus. Data is only valid during the High period of each clock.

Clock synchronization is performed using the wired AND connection of the I²C interfaces to the SCL line, meaning that a High-to-Low transition on the SCL line causes the relevant devices to start counting from their Low period. When a device clock goes Low, it holds the SCL line in that state until the clock High state is reached. See Figure 20 on page 96. The Low-to-High transition of this clock, however, may not change the state of the SCL line if another clock still exists within its Low period. The SCL line is held Low by the device with the longest Low period. Devices with shorter Low periods enter a High WAIT state during this time.

When all devices complete counting off their Low periods, the clock line goes High. There is no difference between the device clocks and the state of the SCL line; therefore, all of the devices begin counting their High periods. The first device to complete its High period again pulls the SCL line Low. In this way, a synchronized SCL clock is generated with its Low period determined by the device with the longest clock Low period, and its High period determined by the device with the shortest clock High period.

Table 55. I²C Clock Control Registers (I2C0_CCR = CCh, I2C1_CCR = DCh)

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 by the eZ80190 device's system clock, which provides a frequency of f_{sclk}. The I²C bus is sampled by the I²C block at the frequency f_{samp} in the following equation.

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

In MASTER mode, the I²C clock output frequency on SCLx (f_{scl}) is provided by:

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

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. These dividers are particularly useful in multimaster systems because the I²C bus sampling frequency 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.

Bus Clock Speed

The I²C bus is defined for bus clock speeds up to 100 kbps (400 kbps in FAST mode).

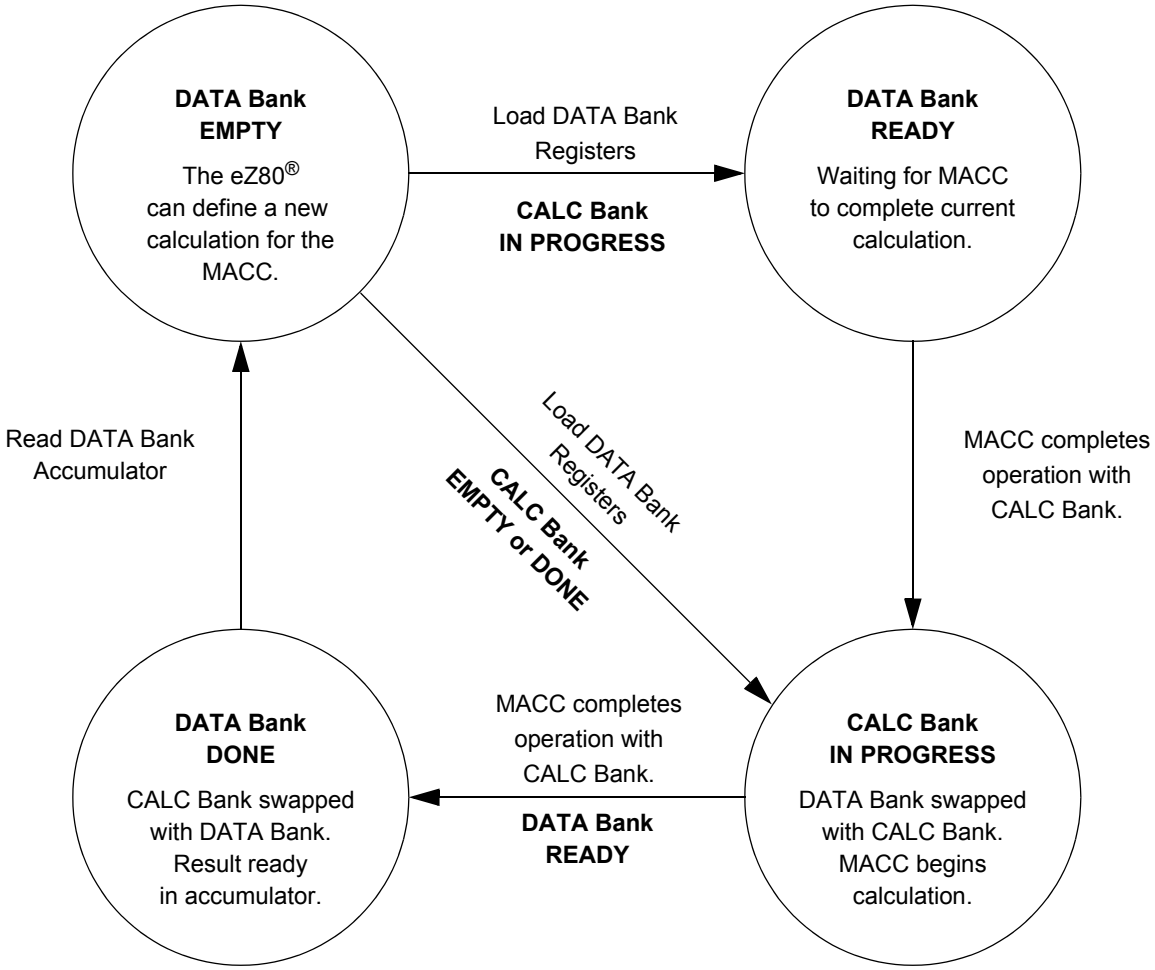

Figure 22. Simplified MACC Status Progression

Software Control of the MACC

The Multiply-Accumulator is designed so that CPU software can set up a calculation by writing to the MACC registers using a single OTI2R block output instruction. For details refer to *eZ80® CPU User Manual (UM0077)*. Depending upon the calculation required, this calculation may require writing to all of the MACC control registers, or just a partial subset.

Similarly, the MACC is designed so that eZ80® CPU software can read the results of a calculation from the MACC Accumulator registers using a single INI2R block input

Table 59. State Progression of the MACC During Operation (Continued)

Current State			Next State	
DATA Bank	**CALC Bank**	**Operation**	**DATA Bank**	**CALC Bank**
DONE	IN PROGRESS	If the MACC completes execution of the current calculation, the CALC bank status changes from IN PROGRESS to DONE.	DONE	DONE
DONE	IN PROGRESS	1. Read the result from the MACC Accumulator registers using the INI2R instruction. When the INI2R instruction completes and the last read is from any byte of MACC_ACx, the DATA bank status changes from DONE to EMPTY. 2. Any read from MACC_AC4 produces the same effect. 3. Any write to any MACC register except for MAC_STAT produces the same effect.	EMPTY	IN PROGRESS
DONE	DONE	1. Read the result from the MACC Accumulator registers using the INI2R instruction. When the INI2R instruction completes and the last read is from any byte of MACC_ACx, the DATA bank status changes from DONE to EMPTY. 2. Any read from MACC_AC4 produces the same effect. 3. Any write to any MACC register except for MAC_STAT produces the same effect.	EMPTY	DONE

IN_SHIFT and OUT_SHIFT

The Multiply-Accumulator on the eZ80190 device features two additional functions, IN_SHIFT and OUT_SHIFT, that can be useful in many DSP operations. Both of these optional functions are controlled by the MACC Control register, MACC_CTL.

IN_SHIFT Function

The IN_SHIFT field, bits 2:0 of the MACC_CTL register, defines the magnitude of the left-shift that is performed when the CPU writes a starting value to the MACC Accumulator registers MACC_AC0, MACC_AC1, MACC_AC2, MACC_AC3, and MACC_AC4. The MACC automatically handles the shift of the 40-bit value as it is written as a succession of 8-bit values. The writes can be left-shifted 0 to 7 bits depending upon the value of IN_SHIFT. The NOISE field, bit 6 of the MACC_CTL register, sets the value used to fill the lsbs vacated during the left-shift operation.

Bit Position	Value	Description
[7:0] MACC_*x*RELOAD	00h–FFh	The reload address for MACC RAM *x* DATA.

MACC Length Register

The MACC_LENGTH register, listed in Table 63, defines the total number of *x* and *y* data pairs that are multiplied and accumulated for the MACC operation.

Table 63. MACC Length Register (MACC_LENGTH = E3h)

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

Note: R/W = Read/Write.

Bit Position	Value	Description
[7:0] MACC_LENGTH	00h–FFh	Total number of address pairs to be multiplied and accumulated for the current MACC operation.

MACC *y* DATA Starting Address Register

The MACC_*y*START register, listed in Table 64, defines the starting address for the MACC to read 16-bit values from the *y* DATA for performing its calculations.

Table 64. MACC *y* DATA Starting Address Register (MACC_*y*START = E4h)

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

Note: R/W = Read/Write.

Bit Position	Value	Description
[7:0] MACC_*y*START	00h–FFh	Starting address for the *y* DATA of MACC RAM.

MACC Accumulator Byte 3 Register

The MACC_AC3 register, listed in Table 71, contains bits 31:24 of the 40-bit MACC Accumulator.

Table 71. MACC Accumulator Byte 3 Register (MACC_AC3 = EBh)

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] MACC_AC3	00h–FFh	MACC Accumulator bits 31:24.

MACC Accumulator Byte 4 Register

The MACC_AC4 register contains the MSB (bits 39:32) of the 40-bit MACC Accumulator. Reading this register changes the status of the DATA bank to EMPTY. Also, if the CALC bank status is DONE, reading this register swaps the banks. In this case, the ending status of the DATA bank is DONE while the CALC bank is EMPTY.

Writing to the MACC_AC4 register, listed in Table 72, changes the status of the DATA bank from EMPTY to READY. If the MACC is ready to begin a new calculation, the banks are swapped and the new calculation begins (CALC bank status becomes IN PROGRESS).

Table 72. MACC Accumulator Byte 4 Register (MACC_AC4 = ECh)

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] MACC_AC4	00h–FFh	MACC Accumulator bits 39:32.

DMA Transfer Modes

There are two modes of operation for the DMA channels. The DMA can transfer data in BURST mode or CYCLE-STEAL mode. The data transfer mode is controlled by the BURST bit in the DMA Control registers (DMAx_CTL[4]).

In BURST mode, the DMA controller takes control of the bus within the eZ80190 device for the entire time period required to complete the data transfer. The CPU is idled while the DMA controller completes its BURST mode data transfer.

The default operation for the DMA controller is CYCLE-STEAL mode in which the DMA controller requests and then gains access to the bus for the transfer of only one byte at a time. After the transfer of each byte, the DMA returns control of the bus back to the CPU. The DMA then waits for the CPU to complete 8 clock cycles before again requesting control of the bus. As a result, other activities can proceed while the DMA is transferring data in the background. CYCLE-STEAL mode slows down the processing of the main program task of the CPU. See Figure 25.

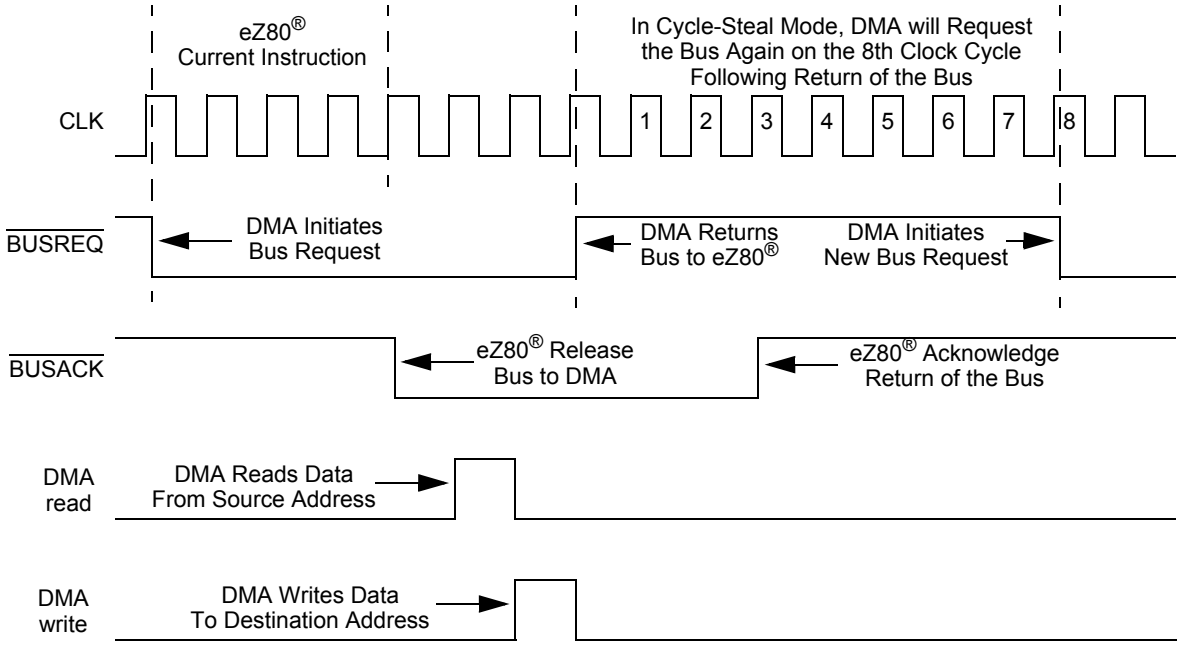

Figure 25. DMA CYCLE-STEAL Timing

ZDI Block READ

A block READ operation is initiated the same as a single-byte read; however, the ZDI master continues to clock in the next byte from the ZDI slave as the ZDI slave continues to output data. The ZDI register address counter increments with each read. If the ZDI register address reaches the end of the Read Only ZDI register address space (20h), the address stops incrementing. Figure 34 displays ZDI block READ timing.

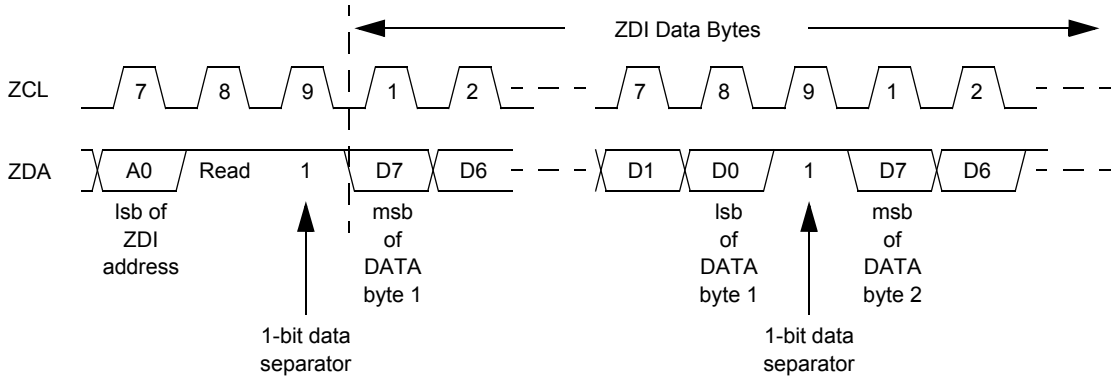

Figure 34. ZDI Block Data Read Timing

Operation Of The eZ80190 Device During ZDI Breakpoints

If the ZDI forces the CPU to break, only the CPU suspends operation. The system clock continues to operate and drive other peripherals. Those peripherals that can operate autonomously from the CPU may continue to operate, if so enabled. For example, the WDT and Programmable Reload Timers continue to count during a ZDI breakpoint.

When using the ZDI interface, any write or read operations of peripheral registers in the I/O address space produces the same effect as read or write operations using the CPU. Because many register Read/Write operations exhibit secondary effects, such as clearing flags or causing operations to commence, the effects of the Read/Write operations during a ZDI break must be taken into consideration. As an example, reading or writing the MACC Accumulator Byte 4 register can cause a bank switch to occur within the MACC.

ZDI Write Only Registers

Table 81 on page 154 lists the ZDI Write Only registers. Many of the ZDI Write Only addresses are shared with ZDI Read Only registers.

Bit Position	Value	Description
4 BRK_ADDR0	0	The ZDI break, upon matching break address 0, is disabled.
	1	The ZDI break, upon matching break address 0, is enabled. ZDI asserts a break when the CPU address, ADDR[23:0], matches the value in the ZDI Address Match 1 registers, {ZDI_ADDR0_U, ZDI_ADDR0_H, ZDI_ADDR0_L}. If the IGN_LOW_0 bit is set to 1, ZDI asserts a break with the upper two bytes of the CPU address, ADDR[23:8], and matches the value in the ZDI Address Match 0 High and Low Byte registers, {ZDI_ADDR0_U, ZDI_ADDR0_LH}. The lower byte of the address is ignored. Breaks can only occur on an instruction boundary. If the address is not the beginning of an instruction, then the break occurs at the end of the current instruction.The break is implemented by setting the BRK_NEXT bit to 1. The BRK_NEXT bit must be reset to 0 to release the break.
2 IGN_LOW_1	0	The *Ignore the Low byte* function of the ZDI Address Match 1 registers is disabled. If BRK_ADDR1 is set to 1, ZDI initiates a break when the entire 24-bit address, ADDR[23:0], matches the 3-byte value {ZDI_ADDR1_U, ZDI_ADDR1_H, ZDI_ADDR1_L}.
	1	The *Ignore the Low byte* function of the ZDI Address Match 1 registers is enabled. If BRK_ADDR1 is set to 1, ZDI initiates a break when only the upper 2 bytes of the 24-bit address, ADDR[23:8], match the 2-byte value {ZDI_ADDR1_U, ZDI_ADDR1_H}. As a result, a break can occur anywhere within a 256-byte page.
1 IGN_LOW_0	0	The *Ignore the Low byte* function of the ZDI Address Match 1 registers is disabled. If BRK_ADDR0 is set to 1, ZDI initiates a break when the entire 24-bit address, ADDR[23:0], matches the 3-byte value {ZDI_ADDR0_U, ZDI_ADDR0_H, ZDI_ADDR0_L}.
	1	The *Ignore the Low byte* function of the ZDI Address Match 1 registers is enabled. If the BRK_ADDR1 is set to 0, ZDI initiates a break when only the upper 2 bytes of the 24-bit address, ADDR[23:8], match the 2 bytes value {ZDI_ADDR0_U, ZDI_ADDR0_H}. As a result, a break can occur anywhere within a 256-byte page.
0 SINGLE_STEP	0	ZDI SINGLE STEP mode is disabled.
	1	ZDI SINGLE STEP mode is enabled. ZDI asserts a break following execution of each instruction.

eZ80® CPU Instruction Set

Table 95 through Table 104 list the eZ80 CPU instructions available for use with the eZ80190 device. The instructions are grouped by class. More detailed information is available in *eZ80® CPU User Manual (UM0077)*.

Note: *The Sleep (SLP) instruction is not supported on the eZ80190 device. Executing a SLP instruction causes the eZ80190 device to behave as if it has received a two-cycle NOP instruction.*

Table 95. Arithmetic Instructions

Mnemonic	Instruction
ADC	Add with Carry
ADD	Add without Carry
CP	Compare with Accumulator
DAA	Decimal Adjust Accumulator
DEC	Decrement
INC	Increment
MLT	Multiply
NEG	Negate Accumulator
SBC	Subtract with Carry
SUB	Subtract without Carry

Table 96. Bit Manipulation Instructions

Mnemonic	Instruction
BIT	Bit Test
RES	Reset Bit
SET	Set Bit

Table 99. Input/Output Instructions (Continued)

Mnemonic	Instruction
OUTI (OTIR)	Output to I/O and Increment (with Repeat)
OUTI2 (OTI2R)	Output to I/O and Increment (with Repeat)
TSTIO	Test I/O

Table 100. Load Instructions

Mnemonic	Instruction
LD	Load
LEA	Load Effective Address
PEA	Push Effective Address
POP	Pop
PUSH	Push

Table 101. Logical Instructions

Mnemonic	Instruction
AND	Logical AND
CPL	Complement Accumulator
OR	Logical OR
TST	Test Accumulator
XOR	Logical Exclusive OR

Table 102. Processor Control Instructions

Mnemonic	Instruction
CCF	Complement Carry Flag
DI	Disable Interrupts
EI	Enable Interrupts
HALT	Halt

Electrical Characteristics

Absolute Maximum Ratings

Stresses greater than those listed in Table 112 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 should be tied to one of the supply voltages (V_{DD} or V_{SS}).

Table 112. Absolute Maximum Ratings

Parameter	Min	Max	Units	Notes
Ambient temperature under bias	–40	+105	C	1
Storage temperature	–65	+150	C	
Voltage on any pin with respect to V_{SS}	–0.3	+6.0	V	2
Voltage on V_{DD} pin with respect to V_{SS}	–0.3	+6.0	V	
Total power dissipation		520	mW	
Maximum current out of V_{SS}		145	mA	
Maximum current into V_{DD}		145	mA	
Maximum output current from active output pin	–8	+8	mA	

Notes:
1. Operating temperature is listed in Table 113.
2. This voltage applies to all pins except where otherwise noted.

E

F

G