UP001907-0910

Product Update

Errata to eZ80190 Microprocessor

eZ80190 Microprocessor with Date Codes 0137 and Later

The errata listed in Table 1 highlights the issues and workaround (if available) related to eZ80190 product with package date codes 0137 and later. The issues are assembled after the 37th week of the year 2001.

Table 1. eZ80190 Microprocessor with Date Codes 0137 and Later

No.	Summary	Detailed Description
1	Breaks during DEBUG mode may occur at incorrect addresses. This error does not affect the part during normal operation.	During DEBUG mode using the ZiLOG Debug Interface (ZDI), all set break points occur as required. However, the CPU can break at program addresses which are not set in ZDI. This situation can occur when a data address is identical to a programmed break point address (for example, during a `LD (HL),A` instruction execution, where (`HL`) is identical to a break address). If this problem is encountered, the suggested workaround is to operate the eZ80190 Webserver at the lower limit of its voltage range (3.0 V) to reduce the false break point address match.
2	SPI operating as a slave in multi-slave configuration can lose transmit data.	If either of the SPI devices on the eZ80190 are configured as a slave device in a multi-slave system, the SPI data can be corrupted by transfers to and from the master and other slaves sharing the same SPI pins. Even though the $\overline{\text{SS}}$ pin is High (not selecting the eZ80190's SPI device), the data is shifted into the SPI's receive buffer. This can overwrite any data that is placed in the SPI's transmit buffer by the eZ80® CPU in preparation for transmit out from the SPI slave. If the SPI devices should be used as slaves in multi-slave systems, the SCK input signal to the eZ80190 should be disabled externally when the $\overline{\text{SS}}$ signal is High. As a result, shifting in of data by the SPI slave receiver is prevented when not selected.
3	A pulse on the SCL line when the I^2C bus is idle and the SDA line is held High causes the I^2C to lock up.	A pulse on the SCL line prior to a START condition or after a STOP condition causes the I^2C to lock up. This situation occurs regardless of the I^2C control register ENAB settings (I2C_CTL). When this situation occurs, an I^2C software reset does not unlock the I^2C bus. **Workarounds** If a lock occurs after the SCL line is released, another device on the I^2C bus can issue a Stop condition by pulsing the SDA line. As a result, the I^2C should unlock. Another option is to initiate an overall SYSTEM RESET of the eZ80190 in order to reset the I^2C block.

eZ80190 Microprocessor with Date Codes 0100–0136

The errata listed in Table 2 highlights the issues and workarounds (if available) related to eZ80190 product with package date codes 0100–0136. The issues are assembled during the first 36 weeks of the year 2001.

Table 2. eZ80190 Microprocessor with Date Codes 0100–0136

No.	Summary	Detailed Description
1	Breaks during DEBUG mode may occur at incorrect addresses. This error does not affect the part during normal operation.	During DEBUG mode using ZiLOG Debug Interface (ZDI), all set break points occur as required. However, the CPU can break at program addresses which are not set in ZDI. This can occur when a data address is identical to or very similar to a programmed break point address (for example during a `LD(HL),A` instruction execution where (`HL`) is similar to a break address). **Workaround** If this problem is encountered, the suggested workaround is to operate the eZ80190 Webserver at the lower limit of its voltage range (3.0 V) as this reduces the false break point address match.
2	SPI operating as a slave in multi-slave configuration can lose transmit data.	If either of the SPI devices on the eZ80190 are configured as a slave device in a multi-slave system, the SPI data can be corrupted by transfers to and from the master and other slaves sharing the same SPI pins. Even though the $\overline{SS}$ pin is high (not selecting the eZ80190's SPI device), the data is shifted into the SPI's receive buffer. This can overwrite any data that is placed in the SPI's transmit buffer by the eZ80® CPU in preparation for transmit out from the SPI slave. If SPI devices should be used as slaves in multi-slave systems, the SCK input signal to the eZ80190 should be disabled externally when the $\overline{SS}$ signal is High. As a result, the shifting in of data by the SPI slave receiver is prevented when not selected.
3	ESD protection does not meet Zilog production standards.	Products with topmark date codes 0100–0136 do not meet ZiLOG standard ESD protection levels. Devices are Mil-Std-883 method 3015 Class 1, less than 2 KV, electrostatic-discharge-sensitive. Use for engineering evaluation and development applications is acceptable. Use in production products is not recommended.
4	A pulse on the SCL line while the I^2C bus is idle and the SDA line is held High causes the I^2C to lock up.	A pulse on the SCL line prior to a Start condition or after a Stop condition causes the I^2C to lock up. This situation occurs irrespective of the I^2C control register ENAB settings (I2C_CTL). When this situation occurs, an I^2C software reset does not unlock the I^2C bus. **Workarounds** If a lock occurs after the SCL line is released, another device on the I^2C bus can issue a Stop condition by pulsing the SDA line. As a result, the I^2C should unlock. Another option is to initiate an overall SYSTEM RESET of the eZ80F91 device to reset the I^2C block.

eZ80190 Microprocessor with Date Codes 00XX

The errata listed in Table 3 highlights the issues and workarounds (if available) related to the eZ80190 product with package date codes 00XX. The issues are assembled in the year 2000.

Table 3. eZ80190 Microprocessor with Date Codes 00XX

No.	Summary	Detailed Description

1	PUSH and POP AF does not function as described in the *eZ80® CPU User Manual (UM0077)*.	POP AF functions differently than POP BC/DE/HL. The PUSH AF and POP AF instructions on the eZ80® operate as follows:
		PUSH AF operation:
		`SPL ← SPL – 3`
		`(SPL) ← dummy`
		`(SPL + 1) ← F`
		`(SPL + 2) ← A`
		POP AF operation:
		`dummy ← (SPL)`
		`F ← (SPL – 1)`
		`A ← (SPL + 2)`
		`SPL ← SPL + 3`
		In the following sequence:
		`PUSH AF`
		`POP BC`
		B gets the value F and C gets a dummy value, whereas for the following sequence:
		`PUSH DE`
		`POP BC`
		B gets the value of D and C gets the value of E.

2	The overflow flag is not set properly on 16/24-bit SBC operation for some boundary cases.	For 16/24 bit operations, the overflow flag is not set if the most negative number (that is, `8000h`) is subtracted from `00h`. Below is an example that fails in ADL = 1 mode:
		`ID A, 0`
		`ID HL,%ff8000`
		`ID DE,%800000`
		`OR A,A`
		`SBC HL, DEA`
		
		In the above example, after the operation, F is found to be 6 (that is, V = 1 and N = 1). Because the operation is to Subtract a negative value from a negative value, no overflow should occur and V = 0.

3	When POP (AF, BC, DE, HL) is immediately followed by any instruction that uses the same register, the upper 8 bits are not correct.	In ADL mode, some instructions following a POP instruction do not work properly if those instructions use the same register pair that the POP instruction uses.
		`EX 1: POP BC`
		`PUSH BD`
		`EX2: POP HL`
		`ADD A, (HL)`
		Workaround
		Place a NOP between the POP and the subsequent instruction.

Table 3. eZ80190 Microprocessor with Date Codes 00XX (Continued)

No.	Summary	Detailed Description
4	When using the I^{2}C interface, consider items 4–9 in this table.	SLAVE mode pulls the SCL line Low while the I^{2}C Master drives it High. The I^{2}C in SLAVE mode, in the acknowledgement clock pulse, pulls the SCL line Low while MASTER drives the SCL line High. This condition truncates the SCL High cycle and appears to users as a possible glitch. The I^{2}C SLAVE operation ensures that the truncated SCL High period is always 5 I^{2}C CLKs, minimum. Adhering to I^{2}C protocol, the SCL line should not be asserted Low by the slave until the master drives SCL Low itself – avoiding any possible malfunction of the SCL clock line.
5	When using the I^{2}C interface, consider items 4–9 in this table.	The 10-bit addressing (READ) protocol is implemented wrongly as published in the eZ80190 Webserver Product Specification (PS0066), which contains the following: 10-bit read: –[S], 1st address (7)+ rd, 2nd address(8) Read Data[8] … [P] Replace the above with: 10-bit write: –[S], 1st address (7)+ wr, 2nd address(8) Write Data[8] … [P] 10-bit read: –[S], 1st address (7)+ wr, 2nd address(8), [RS], 1st address + rd, Read Data(8) … [P]
6	When using the I^{2}C interface, consider items 4–9 in this table.	SCL is not asserted after the transmission of the last byte in SLAVE mode. The I^{2}C interrupts, but does not drive the SCL line Low and instead holds the I^{2}C bus. This condition allows the master device to continue using the I^{2}C bus, thereby corrupting the data and status registers of the Slave, if the controlling CPU is slow.
7	When using the I^{2}C interface, consider items 4–9 in this table.	If the interrupt is cleared while SCL is High, a corrupt address is transmitted. The master start sent interrupt is generated from the falling edge of SDA (2 CLK delay). If the interrupt is serviced before SCL is driven to 0, the address transmitted by the I^{2}C contains a data bit 7 that is forced to 1. If the address is `2Eh`, then `AEh` is transmitted. If the interrupt is serviced after SCL is driven to 0, then the address is transmitted as expected.
8	When using the I^{2}C interface, consider items 4–9 in this table.	The interrupt line is not being cleared following Start and Stop conditions, when larger clock divide ratios are used. In the Stop case (stop bit detected), a status condition `A0h` interrupt with a large divide ratio can recur after clearing.
9	When using the I^{2}C interface, consider items 4–9 in this table.	The stop bit is not sent when larger clock divide ratios greater than or equal to 32 are used. If the interrupt line is cleared while SCL High, and the stop bit to be sent is requested (that is, write `D0h` to the CNTRL register), then the stop bit mechanism starts incorrectly.
10	MIXED mode operation is not recommended.	The `JP.IL` instruction does not work if used in non-ADL mode with a non-zero MBASE to JUMP to the ADL mode. Use `JP.LIL` to enter ADL mode when MBASE is non-zero.

Table 3. eZ80190 Microprocessor with Date Codes 00XX (Continued)

No.	Summary	Detailed Description
11	Instructions that are blank in the opcode map do not automatically trap.	Execution of opcodes 30–37 of the CB opcode map do not generate an illegal instruction trap.
12	Avoid using multiple Wait states and ADL mode-switching instructions.	The instruction SBC.S/L functions differently than as specified in the *eZ80® CPU User Manual (UM0077)*, when ADL = 1. That is, if `BC = 0E6h SBC.S HL, BC` (opcode 52 ed 42) sets the zero flag, as the 16-bit operation `HL – BC = 0`. The zero flag is not set after operation. **Caution:** There are many other instructions which fail when working with multiple Wait states and ADL mode switches. The list includes all instructions which use .S or .IS in ADL mode and .L or .IL in non-ADL mode on any operand.
13	Incorrect edge detect on ring indicator pin.	The 16550 MIMIC specification uses the rising edge of the $\overline{\text{RI}}$ pin to generate the interrupt. The eZ80190 design currently uses the falling edge of the pin NRI.
14	Avoid using continuously asserted interrupt signals.	If the EI instruction is executed with wait states when an interrupt request is continuously asserted, the IEF1 and IEF2 flags are set immediately at the end of an instruction instead at the end of next instruction. The program then jumps to Interrupt Service Routine (ISR). After going to ISR again, the IEF1 and IEF2 flags settle. Due to this condition, one more ISR is executed if the interrupt request is still active at the beginning of the ISR. To prevent nested interrupts, use zero wait states and edge-triggered interrupts, or ensure that the interrupt is not active before leaving the ISR. Re-enabling the interrupts outside the ISR may work in some cases. If a level-sensitive IRQ gets inactive while the RETI instruction POPs the return address, the interrupt controller sends an interrupt request with vector `5Eh` (this address is unused and reserved in the eZ80190). Because there is probably not a defined interrupt vector address `5Eh`, the code jumps to an unexpected location to perform the ISR. The eZ80190 interrupt controller requires that the interrupt source be active until the start of the ISR. Refer to the *eZ80190 Webserver Product Specification (PS0066)*. This problem occurs in bothV1.0 and V1.1.
15	Avoid using the Halt instruction.	If the opcode register contains 76 (Halt opcode) and if a bus acknowledge is simultaneously given for the bus master (which is asserted as a bus request), the CPU bus gets into the Halt state after the CPU de-asserts the bus acknowledge. If an $\overline{\text{NMI}}$ is asserted, the CPU must come out of Halt mode, but the CPU decodes the Halt instruction again and re-enters HALT mode.
16	Avoid using RETI.S/ RETI.L in mixed mode.	RETI.S/RETI.L – when MIXED mode is set (STMIX), RETI.S/RETI.L does not POP the ADL byte that is pushed onto the stack. RETI.S/.L with STMIX PUSHes and POPs the ADL byte to the stack. However, the wrong value is pushed onto the stack when switching from ADL = 0 to ISR (3 instead of 2), so when RETI.S/.L is executed, the mode stays set to ADL = 1. If the value on the stack is modified to be the correct value, RETI.S/.L works correctly.

Table 3. eZ80190 Microprocessor with Date Codes 00XX (Continued)

No.	Summary	Detailed Description
17	Avoid using INC SP before return interrupt.	INC SP followed by RETI – INC SP followed by RETI makes the program hang.
18	Avoid using LD.s to modify the stack pointer in ADL mode.	LD.S SP,HL – the error is related to the LD.S SP,HL (ADL = 1) instruction. This instruction modifies the SPL instruction and not SPS. Simulations find that LD.S SP,HL ADL=1 works (HL $\rightarrow$ SPS). There is a problem with LD.L SP,HL ADL=0 (HL $\rightarrow$ SPS not SPL).

Warning: DO NOT USE IN LIFE SUPPORT