Product Update

Errata to Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x (Z8 Encore!®)

Z8F640x Family Devices with Date Codes 0239 and Later, without QUAL Topmark

The errata listed in Table 1 are found in Zilog's Z8F640x family products (includes Z8F480x, Z8F320x, Z8F240x, and Z8F160x) without a QUAL topmark and with date codes 0239 and later, where the date code is YYWW (year and week of assembly). When reviewing the following errata, refer to the most recent version of the product specification. Data contained in this document is Preliminary only.

Table 1. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later

Sl No	Summary	Description
1	The On-Chip Debugger's (OCD) Step, Stuff, and Execute instructions do not work if an interrupt is pending.	The OCD's Step, Stuff, and Execute instructions do not work if an interrupt is pending. When in DEBUG mode, the eZ8™ CPU will not acknowledge interrupts or DMA requests. However, if an interrupt or DMA request is pending, the eZ8 CPU will not acknowledge an instruction. If an interrupt is pending and an OCD Step, Stuff, or Execute instruction is executed, the OCD will wait forever for the eZ8 CPU to acknowledge the opcode because of the pending interrupt. **Workaround** The OCD must look at the next instruction before single stepping and take appropriate measures. Instead of executing the Enable Interrupt (EI) instruction, rewrite the PC to the instruction following the EI and then enable interrupts through a register write to the interrupt control register.
2	Extraneous register reads by the eZ8 CPU.	There are several instructions during which the CPU performs extra register reads. Most are addresses the CPU was trying to read, the CPU reads the same register twice. There are a couple instructions where the CPU reads from random addresses. This is not a problem, unless the register being read is affected by a read operation. The registers affected by read operations include the WDTCTL and DMAA_STAT registers and the UART, SPI, and I^2C Receive Data registers. If a read occurs on these registers, receive characters may be lost or the WDT status lost. **Workaround** Do not set RP to %XF. Also, only use the LDX instruction on peripheral registers affected by read operations.
3	SPI does not support single bit data transfers.	The SPI does not function properly when configured for single-bit data transfers. This is not a typical SPI data format. **Workaround** None.

Table 1. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
4	UART Overrun errors may be missed.	Framing Error, Parity Error, Break Detect, and Rx Overrun Error conditions are cleared up on reading the UART Receive Data register. During the time between reading the UART Status register and the UART Receive Data register, it is possible for another character to be received. This causes all UART error flags and the UART Receive Data register to be updated with the new character. Thus making it possible to miss the Overrun Error. The window for this error to occur if a UART Overrun Error occurs between the time the UART Status register is read and the UART Receive Data register is read. If vectored interrupts are used, the UART should be serviced and Receiver Overrun conditions should not occur. If you have long interrupt service routines (ISR) (bad coding style) or are polling the UART instead of using vectored interrupts, Overrun errors become more likely. The window for this problem to occur is still small, yet becomes more probable if UART Receiver Overrun conditions occur frequently. **Workaround** When the user code employs vectored interrupts for the UART and does not have long ISR, this is not a problem. Even for long ISR, the problem can be avoided by, • Nesting the ISR • Adjusting the interrupt masks and re-enabling interrupts
5	Interrupts can be lost if received by the interrupt controller at the same time as a write to the corresponding IRQ register.	Incoming interrupts can be lost if received by the interrupt controller at the same time as a write to the corresponding IRQ register. **Workaround** Clear the Continuous Assertion interrupts using a two-step interrupt service process. In the ISR, first check if the interrupt source (for example, the UART) really has a pending interrupt. If yes: • Process the interrupt as usual. • Clear the interrupt at the source (for example, at the UART). • Do not clear the IRQ register bit (this would make it possible to miss another incoming interrupt). • Execute a return from the ISR. After this first pass through the ISR, the IRQ register bit will still be set to 1. This will cause the interrupt to occur again. When the Z8 Encore! vectors to the interrupt, check if the interrupt source (for example, the UART) really has a pending interrupt. If there is no pending interrupt, immediately execute a return from the ISR.

Table 1. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
6	The TXST bit in the SPISTAT register does not assert until the transmission actually starts.	When data is written to the SPIDATA register to be transmitted, the TXST bit in the SPISTAT register does not assert until the transmission actually starts which results in a short delay (delay is dependent on the baud rate). It is possible for software to poll the TXST bit and see a 0 before the transmission has started. Software may erroneously conclude that the data has already been transmitted. **Workaround** User code can poll the IRQ bit in the SPISTAT register. Even when the SPI interrupt is disabled, the IRQ bit will assert at the end of the data transaction and remain asserted until cleared by software.
7	Reading the UART Status 1 register through the OCD always returns the value 00H.	The UART Status 1 register is cleared when read. When the OCD reads the register it holds the read for multiple system clock cycles, thereby clearing the value before completing the read. Thus, the value returned through the OCD is always 00H. This issue affects only OCD operation and does not affect normal operation. **Workaround** Issue a CPU command through the OCD to transfer the UART Status 1 register data to a Register File location. Then the desired UART Status 1 register data can be read from the Register File.
8	When used as simple timers, the Baud Rate Generators in the UARTs, I^2C, and SPI generate a spurious interrupt at the beginning of the count.	When the Baud Rate Counters for UARTs, I^2C, and SPI are placed in timer mode they immediately generate an interrupt. This is because the counters are incorrectly reset to 0001H rather than the reload value. Since 0001H is the reload state, it initiates an interrupt request. **Workaround** • Use one of the four other Timers rather than the Baud Rate Generators in timer mode. • Delay enabling the interrupt for these counters until the count value has progressed beyond 0001H. • Write the ISR so that is disregards the first interrupt. • Clear the associated interrupt request in the Interrupt Control shortly after starting the timer.

Table 1. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
9	SPI operating as a Slave in a multi-Slave system can lose transmit data.	If the SPI devices on the Z8 Encore!® is configured as a Slave device in a multi-Slave system, the SPI data can be corrupted by transfers to and from the Master to other Slaves sharing the same SPI pins. Even though the $\overline{SS}$ input pin is High (that is, not selecting the Z8 Encore!'s SPI device), the data will be shifted into the SPI's receive buffer. This can overwrite any data that has been placed in the SPI's transmit buffer by the eZ8 CPU in preparation for transmit out from the SPI Slave. **Workaround** If it is desired to use the SPI device as a Slave in multi-Slave systems, the SCK input signal to the Z8 Encore! should be disabled externally when the $\overline{SS}$ signal is High. This will prevent shifting in of data by the SPI Slave receiver when not selected.
10	ADC output is inaccurate for input values below approximately 20 mV.	The output from the ADC can vary widely when the input signal drops below about 20 mV. **Workaround** Measure analog inputs only above 20 mV.
11	eZ8 CPU opcode timing is incorrect for three Load instructions.	The following instructions have timing errors in which an extra (unused) clock cycle is inserted during instruction execution: Instruction / Opcode / Spec Cycles / Actual Cycles LD / E4 / 2 / 3 LD / E7 / 3 / 4 LD / E5 / 3 / 4 **Workaround** None. This issue is not likely to affect the user code. If the user code has software timing loops, it is possible that future Z8 Encore! products will have different loop timing using the same code. Due the pipelined nature of the eZ8 CPU, timing loops are less likely to be employed than on older Z8® products.
12	GPIO Port pins draw current when input voltage exceeds one diode drop above the supply voltage.	For the 5 V-input tolerant GPIO pins, when the input voltage exceeds approximately 4.0 V (for a 3.3 V supply voltage), current will be drawn by the input pin. **Workaround** For GPIO pins that toggle at low frequencies, a 10 kΩ resistor can be placed between the GPIO pin and the external driver. This will limit the current into the pin to about 150 µA. For higher frequencies, a 1 kΩ resistor should be used. This will limit the input current to the pin to about 1.5 mA.

Table 1. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
13	With the SPI configured for multi-master operation, an occurrence of multi-master collision will not be detected.	With the SPI configured for multi-master operation, a multi-master collision, should it occur, will not be detected by the Z8 Encore!'s SPI device. **Workaround** A GPIO pin, configured as an input with interrupt on a falling edge, could potentially be used to detect multi-master collisions. If the interrupt occurs with the SPI configured as a Master, the user software could determine that a multi-master collision has likely occurred.
14	The UART Receiver and Transmitter incorrectly test for the setting of the Parity Enable bit when in Multiprocessor (MP) mode.	The UART Receiver and Transmitter incorrectly test for the setting of the Parity Enable (PEN) bit when in MP mode. The UART is not supposed to use parity when the MP bit is enabled. **Workaround** When operating in MP mode, the user code should disable parity by clearing the PEN bit in the UART Control 0 register.
15	Data written to the I^2C Data register cannot be read back.	Data written to the I^2C Data register for transmission cannot be read back. This is unlikely to affect user operation at all. **Workaround** None. Generally, the user code does not need to read back the data that was written to the I^2C Data register for transmission.
16	Execution of a software TRAP instruction may erroneously clear pending interrupts.	If an interrupt is pending and a software TRAP or an illegal instruction TRAP is executed, the highest priority pending interrupt will be erroneously cleared. This causes interrupts to be lost. **Workaround** Do not execute a software TRAP instruction.
17	Driving the GPIO port pins with a high-impedance source may result in logic errors.	When configured as inputs, the GPIO pins source high (50+ µA) current when the input voltage on the pin is near mid-range. If a high impedance device is used to drive the input, this can result in logic errors due to the resistive divider effect of the current source and the external impedance. **Workaround** Do not drive the GPIO port input pins with high-impedance drivers (greater than approximately 20 kΩ).

Table 1. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
18	When receiving data, the IrDA endec may have bit errors if the external transmitter's baud rate is greater than endec baud rate.	The IrDA endec is sensitive to external transmitters that have baud rates somewhat higher than the baud rate of the Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x's endec. This can cause bit errors in the data transmission. **Workaround** When receiving, increase the baud rate of the UART and IrDA endec by a few percent. The endec can handle minor baud rate discrepancies to an external transmitter that is slower, but is sensitive to an external transmitter that is faster. When the endec is transmitting, the baud rate should be set as close as possible to the desired baud rate.
19	When the CPU exits from HALT mode, it fails to reset the master Interrupt Request Enable (IRQE) bit.	When the CPU exits from HALT mode, it fails to reset the master Interrupt Request Enable (IRQE) bit (bit 7 of the Interrupt Control Register). Watchdog Timer (WDT) interrupts will cause the Program Counter (PC) and Flags to be pushed twice on the stack. The first push will be the PC and Flags from where the interrupt occurred. The second push will be the starting address and Flags of the ISR. This problem also affects exits from HALT mode caused by other interrupt sources if more than one interrupt is pending. If only a single interrupt is pending then the routine is executed normally except that interrupts are not disabled. **Workaround** To mimic standard interrupt operation, the ISR should execute a Disable Interrupts (DI) instruction to reset the Master Interrupt Request Enable (IRQE) bit to 0. Further, on WDT interrupts before exiting, the ISR should add three to the Stack Pointer (SP). On Normal interrupts the ISR should check the PC on the stack. If the PC on the stack contains the starting address of the ISR, then the ISR should add three to the SP. This problem only affects exits from HALT mode.
20	The DMA does not support the ADC in CONTINUOUS mode.	The DMA does not support the ADC in CONTINUOUS mode. **Workaround** None.

Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x with QUAL Topmark

The errata listed in Table 2 are found in the Z8F640x products with a QUAL topmark and date codes 0239 and later, where the date code is YYWW (year and week of assembly).When reviewing the following errata, refer to the most recent version of the product specification. Data contained in this document is Preliminary only.

Table 2. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later

Sl No	Summary	Description
1	The OCD's Step, Stuff, and Execute instructions do not work properly if an interrupt is pending.	The OCD's Step, Stuff, and Execute instructions do not work if an interrupt is pending. When in DEBUG mode, the eZ8 CPU will not acknowledge interrupts or DMA requests. However, if an interrupt or DMA request is pending, the eZ8 CPU will not acknowledge an instruction. If an interrupt is pending and an OCD Step, Stuff, or Execute instruction is executed, the Debugger will wait forever for the eZ8 CPU to acknowledge the opcode because of the pending interrupt. **Workaround** The OCD must look at the next instruction before single stepping and take appropriate measures. Instead of executing the EI instruction, rewrite the PC to the instruction following the EI and then enable interrupts through a register write to the interrupt control register.
2	Extraneous register reads by the eZ8 CPU.	There are several instructions during which the CPU performs extra register reads. Most are addresses the CPU was trying to read, the CPU reads the same register twice. There are a couple instructions where the CPU reads from random addresses. This is typically not a problem, unless the register being read is affected by a read operation. The registers affected by read operations include the WDTCTL and DMAA_STAT registers and the UART, SPI, and I^2C Receive Data registers. If a read occurs on these registers, receive characters may be lost or the WDT status lost. **Workaround** Do not set RP to %XF. Also, only use the LDX instruction on peripheral registers affected by read operations.
3	SPI does not support single bit data transfers.	The SPI does not function properly when configured for single-bit data transfers. This is not a typical SPI data format. **Workaround** None.

Table 2. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
4	UART Overrun errors may be missed.	Framing Error, Parity Error, Break Detect, and Rx Overrun Error conditions are cleared up on reading the UART Receive Data register. During the time between reading the UART Status register and the UART Receive Data register, it is possible for another character to be received. This causes all UART error flags and the UART Receive Data register to be updated with the new character. Thus making it is possible to miss the Overrun Error. The window for this error to occur is very small. It can only occur if a UART Overrun Error occurs between the time the UART Status register is read and the UART Receive Data register is read. If vectored interrupts are used, the UART should be serviced in a timely fashion and Receiver Overrun conditions should not occur. If you have long ISR (bad coding style) or are polling the UART instead of using vectored interrupts, Overrun errors become more likely. The window for this problem to occur is still small, yet becomes more probable if UART Receiver Overrun conditions occur frequently. **Workaround** When the user code employs vectored interrupts for the UART and does not have long ISR, this is not a problem. Even for long ISR, the problem can be avoided by, • Nesting the ISR • Adjusting the interrupt masks and re-enabling interrupts
5	Interrupts can be lost if received by the interrupt controller at the same time as a write to the corresponding IRQ register.	Incoming interrupts can be lost if received by the interrupt controller at the same time as a write to the corresponding IRQ register. **Workaround** Clear the Continuous Assertion interrupts using a two-step interrupt service process. In the ISR, first check if the interrupt source (for example, the UART) really has a pending interrupt. If yes: • Process the interrupt as usual. • Clear the interrupt at the source (for example, at the UART). • Do not clear the IRQ register bit (this would make it possible to miss another incoming interrupt). • Execute a return from the ISR. After this first pass through the ISR, the IRQ register bit will still be set to 1. This will cause the interrupt to occur again. When the Encore vectors to the interrupt, check if the interrupt source (for example, the UART) really has a pending interrupt. If there are no pending interrupts, immediately execute a return from the ISR.

Table 2. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
6	Reset is not generated when power supply voltage (VCC) drops below the VBO threshold.	Reset is not generated when power supply voltage (VCC) drops below the VBO threshold. **Workaround** An external VBO circuit is used in the user application to drive the external RESET pin on the Z8 Encore! when the system supply voltage drops below acceptable operating levels.
7	The TXST bit in the SPISTAT register does not assert until the transmission actually starts.	When data is written to the SPIDATA register to be transmitted, the TXST bit in the SPISTAT register does not assert until the transmission actually starts which results in a short delay (delay is dependent on the baud rate). It is possible for software to poll the TXST bit and see a 0 before the transmission has started. Software may erroneously conclude that the data has already been transmitted. **Workaround** User code can poll the IRQ bit in the SPISTAT register. Even when the SPI interrupt is disabled, the IRQ bit will assert at the end of the data transaction and remain asserted until cleared by software.
8	Reading the UART Status 1 register through the On-Chip Debugger always returns the value 00H.	The UART Status 1 register is cleared when read. When the OCD reads the register it holds the read for multiple system clock cycles, thereby clearing the value before completing the read. Thus, the value returned through the OCD is always 00H. This issue only affects OCD operation and does not affect normal operation. **Workaround** Issue a CPU command through the OCD to transfer the UART Status 1 register data to a Register File location. Then the desired UART Status 1 register data can be read from the Register File.
9	When used as simple timers, the Baud Rate Generators in the UARTs, I^2C, and SPI generate a spurious interrupt at the beginning of the count.	When the Baud Rate Counters for UARTs, I^2C and SPI are placed in timer mode they immediately generate an interrupt. This is because the counters are incorrectly reset to 0001H rather than the reload value. Since 0001H is the reload state, it initiates an interrupt request. **Workaround** • Use one of the four other Timers rather than the Baud Rate Generators in TIMER mode. • Delay enabling the interrupt for these counters until the count value has progressed beyond 0001H. • Write the ISR so that is disregards the first interrupt. • Clear the associated interrupt request in the Interrupt Control shortly after starting the timer.

Table 2. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
10	SPI operating as a Slave in a multi-Slave system can lose transmit data.	If the SPI devices on the Z8 Encore! is configured as a Slave device in a multi-Slave system, the SPI data can be corrupted by transfers to and from the Master to other Slaves sharing the same SPI pins. Even though the $\overline{SS}$ input pin is High (that is, not selecting the Z8 Encore!'s SPI device), the data will be shifted into the SPI's receive buffer. This can overwrite any data that has been placed in the SPI's transmit buffer by the eZ8 CPU in preparation for transmit out from the SPI Slave. **Workaround** If it is desired to use the SPI device as a Slave in multi-Slave systems, the SCK input signal to the Z8 Encore! should be disabled externally when the $\overline{SS}$ signal is High. This will prevent shifting in of data by the SPI Slave receiver when not selected.
11	ADC output is inaccurate for input values below approximately 20 mV.	The output from the ADC can vary widely when the input signal drops below about 20 mV. **Workaround** Measure analog inputs only above 20 mV.
12	eZ8 CPU opcode timing is incorrect for three Load instructions.	The following instructions have timing errors in which an extra (unused) clock cycle is inserted during instruction execution: Instruction / Opcode / Spec Cycles / Actual Cycles LD / E4 / 2 / 3 LD / E7 / 3 / 4 LD / E5 / 3 / 4 **Workaround** None. This issue is not likely to affect the user code. If the user code has software timing loops, it is possible that future Z8 Encore! products will have different loop timing using the same code. Due the pipelined nature of the eZ8 CPU, timing loops are less likely to be employed than on older Z8® products.
13	GPIO Port pins draw current when input voltage exceeds one diode drop above the supply voltage.	For the 5 V-input tolerant GPIO pins, when the input voltage exceeds approximately 4.0 V (for a 3.3 V supply voltage), current will be drawn by the input pin. **Workaround** For GPIO pins that toggle at low frequencies, a 10 kΩ resistor can be placed between the GPIO pin and the external driver. This will limit the current into the pin to about 150 µA. For higher frequencies, a 1 kΩ resistor should be used. This will limit the input current to the pin to about 1.5 mA.

Table 2. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
14	With the SPI configured for multi-master operation, an occurrence of multi-master collision will not be detected.	With the SPI configured for multi-master operation, a multi-master collision, should it occur, will not be detected by the Z8 Encore!'s SPI device. **Workaround** A GPIO pin, configured as an input with interrupt on a falling edge, could potentially be used to detect multi-master collisions. If the interrupt occurs with the SPI configured as a Master, the user software could determine that a multi-master collision has likely occurred.
15	The UART Receiver and Transmitter incorrectly test for the setting of the Parity Enable bit when in MULTIPROCESSOR mode.	The UART Receiver and Transmitter incorrectly test for the setting of the Parity Enable (PEN) bit when in MULTIPROCESSOR mode. The UART is not supposed to use parity when the multiprocessor bit is enabled. **Workaround** When operating in MULTIPROCESSOR mode, the user code should disable parity by clearing the PEN bit in the UART Control 0 register.
16	Data written to the I^2C Data register cannot be read back.	Data written to the I^2C Data register for transmission cannot be read back. This is unlikely to affect user operation at all. **Workaround** None. Generally, the user code does not need to read back the data that was written to the I^2C Data register for transmission.
17	Execution of a software TRAP instruction may erroneously clear pending interrupts.	If an interrupt is pending and a software TRAP or an illegal instruction TRAP is executed, the highest priority pending interrupt will be erroneously cleared. This causes interrupts to be lost. **Workaround** Do not execute a software TRAP instruction.
18	Driving the GPIO port pins with a high-impedance source may result in logic errors.	When configured as inputs, the GPIO pins source high (50+ µA) current when the input voltage on the pin is near mid-range. If a high impedance device is used to drive the input, this can result in logic errors due to the resistive divider effect of the current source and the external impedance. **Workaround** Do not drive the GPIO port input pins with high-impedance drivers (greater than approximately 20 kΩ).

Table 2. Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x Errata for Devices with Date Codes 0239 and Later (Continued)

Sl No	Summary	Description
19	When receiving data, the IrDA endec may have bit errors if the external transmitter's baud rate is greater than endec baud rate.	The IrDA endec is sensitive to external transmitters that have baud rates somewhat higher than the baud rate of the Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x's endec. This can cause bit errors in the data transmission. **Workaround** When receiving, increase the baud rate of the UART and IrDA endec by a few percent. The endec can handle minor baud rate discrepancies to an external transmitter that is slower, but is sensitive to an external transmitter that is faster. When the endec is transmitting, the baud rate should be set as close as possible to the desired baud rate.
20	The DMA does not support the ADC in CONTINUOUS Mode.	The DMA does not support the ADC in CONTINUOUS mode. **Workaround** None.

Z8F640x, Z8F480x, Z8F320x, Z8F240x, Z8F160x

The errata listed in Table 3 are found in the Z8F6403 devices with date codes 0226 and earlier, where the date code is YYWW (year and week of assembly). If you have devices with these date codes, you must contact Zilog® to obtain replacements. When reviewing the following errata, refer to the most recent version of the product specification. Data contained in this document is Preliminary only.

Table 3. Z8F6403 Errata for Devices with Date Codes 0226 and Earlier

Sl No	Summary	Description
1	OCD incorrectly single steps through an EI instruction if an interrupt is pending.	If the OCD is single stepping through an EI instruction and there are interrupts pending, the eZ8 CPU does not pause prior to executing the interrupt and vectors to an invalid interrupt vector address located at Program Memory addresses 000H and 001H (the User Option Bits information). **Workaround** The OCD must look at the next instruction before single stepping and take appropriate measures. Instead of executing the EI instruction, rewrite the PC to the instruction following the EI and then enable interrupts through a register write to the interrupt control register.

Table 3. Z8F6403 Errata for Devices with Date Codes 0226 and Earlier (Continued)

Sl No	Summary	Description
2	First cycle of Pulse-Width Modulator (PWM) output is incorrect, but all subsequent cycles operate properly.	The TIOI bit, which sets the initial state of the timer output, also sets the transition when the counter reaches the PWM value. Thus, no transition occurs the first time the counter reaches the PWM value. **Workaround** During timer initialization, the 16-bit Reload value must be written into the Timer High and Low byte registers. This write causes a reload to happen as soon as the timer is enabled, forcing timer out to its proper state.
3	Timer interrupts in CAPTURE or CAPTURE/COMPARE mode may not clear properly.	When the counter is in capture or capture/compare mode the interrupt to the IRQ controller is held from the time the count matches until the pre-scaler times out. Thus the interrupt can be held for as much as 1-128 (Depending upon the pre-scale value) cycles. If CPU services the interrupt before the pre-scaler has timed out, the interrupt is not cleared automatically. Also just clearing the bit does not work as well since the interrupt is not de-asserted until the pre-scaler decrements to 0. The ISR must continually clear the bit until it stays cleared. **Workaround** The ISR must clear the IRQ bit and then verify that it stayed cleared. If it is not cleared then it repeat the process until the IRQ bit stays cleared.
4	ADC output values are one-half the measured value.	The ADC exhibits a gain error that results in the output value being one-half the actual measured value. **Workaround** Left shifting the 10-bit value converts (equivalent to a multiply by 2) scales the ADC output properly. Alternatively, the 10-bit ADC data can be read as (ADC_DATAH[6:0], ADC_DATAL[7:5]). ADC_DATAL bit 5 is not indicated in the Product Specification, but does function as an extra bit in the current silicon for test purposes.
5	ADC output is inaccurate for input values below approximately 50 mV.	The output from the ADC can vary widely when the input signal drops below about 30 mV to 50 mV. **Workaround** Measure analog inputs only above 50 mV.
6	Power-On Reset (POR) voltage threshold does not meet specification.	The POR voltage threshold is approximately 3.1 V which can prevent proper operation with power supplies below 3.1 V. **Workaround** On power-up, insure that the power supply is kept between 3.2 V and 3.6 V during operation.

Table 3. Z8F6403 Errata for Devices with Date Codes 0226 and Earlier (Continued)

Sl No	Summary	Description
7	Port H alternate function forces pins to output.	When the alternate function for Port H is enabled, the Port H pins are configured as outputs. **Workaround** Do not enable the Port H alternate function when using the Port H ADC inputs. These pins continue to function properly as ADC inputs even though not configured for alternate function. For ADC operation configure the Port H pins as inputs.
8	Extraneous register reads by the eZ8 CPU.	There are several instructions during which the CPU performs extra register reads. Most are addresses the CPU was trying to read, the CPU reads the same register twice. There are a couple instructions where the CPU reads from random addresses. This is typically not a problem, unless the register being read is affected by a read operation. The registers affected by read operations including the WDTCTL register are the UART, SPI, and I^2C Receive Data registers. If a read occurs on these registers, receive characters may be lost or the WDT status lost. **Workaround** Do not set RP to %XF. Also, only use the LDX instruction on peripheral registers affected by read operations.
9	UART Overrun errors may be missed.	Framing Error, Parity Error, Break Detect, and Rx Overrun Error conditions are cleared up on reading the UART Receive Data register. During the time between reading the UART Status register and the UART Receive Data register, it is possible for another character to be received. This causes all UART error flags and the UART Receive Data register to be updated with the new character, making it is possible to miss the Overrun Error. The window for this error only occurs if a UART Overrun Error occurs between the time the UART Status register is read and the UART Receive Data register is read. If vectored interrupts are used, service the UART. Receiver Overrun conditions should not occur thereafter. If you have long ISR or are polling the UART instead of using vectored interrupts, Overrun errors become more likely. The window for this problem to occur is small, yet becomes more probable if UART Receiver Overrun conditions occur frequently. **Workaround** When the user code employs vectored interrupts for the UART and does not have long ISR, this is not a problem. Even for long ISR, the problem can be avoided by, • Nesting the ISR • Adjusting the interrupt masks and re-enabling interrupts

Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer