



UP005104-0404

Product Update

Errata for eZ80F91 Engineering Sample AA Silicon

eZ80F91 MCU with Date Codes 0247 to 0316

The seven errata listed in Table 1 are found in the Revision AA silicon for the eZ80F91 product with package date codes of 0247 to 0316, indicating assembly no earlier than the 47th week of the year 2002, and no later than the 16th week of the year 2003.

Table 1. Errata to the eZ80F91 Device

No.	Issue	Detailed Description
1	The infrared endec receiver misses bits when configured for low data rates and the incoming signals are only 1.6µs.	The infrared encoder/decoder (endec) samples the incoming IR pulses using the baud rate clock divided by 16. This sampling rate can be insufficient to capture the incoming pulses when they use a short-pulse format and low data rates. This short pulse, 1.6 µs, is within IrDA specifications; however, not all transmitters use this particular signalling format. When the external transmitter is sending $3/_{16}$ IR pulses, the endec on the eZ80F91 device receive the data properly.
2	The Real-Time Clock consumes excess current when the eZ80F91 is not in SLEEP mode.	When the eZ80F91 device is not in SLEEP mode, the system clock drives the Real-Time Clock's control and data registers. As a result, excess current is consumed through the RTC_V _{DD} pin, which supplies power to the portion of the clock tree that drives the RTC registers. This current consumption is a function of operating frequency.
3	GPIO Port B and C pins draw current when input voltage exceeds one diode drop above the supply voltage.	Error: For the 5V input-tolerant GPIO pins, when the input voltage exceeds approximately 4.0V (for a 3.3V supply voltage), current is drawn by the input pin. Work-Arounds: For GPIO pins that toggle at low frequencies, a 10KΩ resistor can be placed between the GPIO pin and the external driver. This placement limits the current into the pin to about 150µA. For higher frequencies, a 1KΩ resistor should be used; the resulting input current to the pin is limited to about 1.5mA.
4	WP pin functionality.	Assertion of the $\overline{\text{WP}}$ pin does <u>not</u> protect Flash memory from a Mass Erase operation. Assertion of the WP pin protects the Flash memory boot block from memory Writes, and also protects all blocks (except the Information Page) from Page Erase and I/O Writes.
5	RTC does not run on RTC_V _{DD} only	The RTC does not operate correctly if system power is removed.
6	Disable of PLL when selected as system clock source.	If the PLL is selected as the system clock source, the PLL can be disabled by the eZ80 [®] CPU. As a result, the system clock is caused to stop. The eZ80F91 will not respond to RESET pin assertion without a system clock. If the system clock source is disabled, the eZ80F91 can only be reset by a WDT reset, a POR reset, or a RTC SLP_WAKE.

Table 1. Errata to the eZ80F91 Device (Continued)

No.	Issue	Detailed Description
7	A pulse on the SCL line while the I ² C bus is idle and the SDA line is held High causes the I ² C to lock up.	<p>Error: A pulse on the SCL line prior to a START condition or after a STOP condition causes the I²C to lock up. This situation occurs irrespective of the I²C Control Register ENAB settings (I2C_CTL). When this situation occurs, an I²C Software Reset does not unlock the I²C bus.</p> <p>Work-Arounds: To prevent lock-up, it is possible to completely disable the I²C block prior to any bus activity using the Clock Peripheral Power-Down Register 1 (CLK_PPD1) before setting ENAB in the I²C Control Register.</p> <p>In the event that lock-up of the I²C block occurs, then after the SCL line is released, another device on the bus can issue a STOP condition by pulsing the SDA line. As a result, the I²C should unlock.</p> <p>Another option is to initiate an overall SYSTEM RESET of the eZ80F91 device to reset the I²C block.</p>
8	GPIO Edge Trigger Interrupt Mapping Error	<p>Error: GPIO Mode 6, dual edge-triggered interrupt, and GPIO Mode 9, falling/rising edge-triggered interrupt, are mapped backwards. For example, an interrupt on PB0 is decoded as an interrupt on PB7, PB1 as an interrupt on PB6, etc. This issue applies to all four GPIO ports.</p> <p>Workarounds: To maintain software compatibility with future silicon updates and to reduce programming confusion when using edge-triggered interrupts, ZiLOG recommends the following:</p> <ol style="list-style-type: none"> 1. Program the interrupts in pairs, e.g.: Px0 with Px7, Px1 with Px6, Px2 with Px5, and Px4 with Px3. 2. If the application does not allow programming interrupts in pairs, extreme caution will be required when writing Interrupt Service Routines and modifying the Interrupt Priority Register INT_Px Reset.



This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact: **3**

ZiLOG Worldwide Headquarters

532 Race Street
San Jose, CA 95126
Telephone: 408.558.8500
Fax: 408.558.8300
www.ZiLOG.com

Document Disclaimer

ZiLOG is a registered trademark of ZiLOG Inc. in the United States and in other countries. All other products and/or service names mentioned herein may be trademarks of the companies with which they are associated.

©2004 by ZiLOG, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval of ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses are conveyed, implicitly or otherwise, by this document under any intellectual property rights.