ECO and Workarounds for Bugs in ESP32

Version 1.0

About This Guide

This document details the silicon errata for the ESP32. The structure is as below:

Chapter	Title	Content
Chapter 1	Chip Revision	Introduction to how to identify the chip revision.
Chapter 2	Errata Summary	Summary of all the errata.
Chapter 3	Errata Descriptions and Workarounds	Detailed description of each erratum and possible workarounds.

Release Notes

Date	Version	Release notes
2016.11	V1.0	Initial release.

Table of Contents

1. Chip Revision

The chip revision is identified by the eFuse bit. Details can be found in Chapter eFuse in *ESP32 Technical Reference Manual*.

Table 1-1. Chip Revision

Chip revision	Release date
0	2016.09
1	2017.02

2. Errata Summary

Table 2-1 provides a summary of the errata.

Table 2-1. Errata Summary

Section	Title	Affected revisions
Section 3.1	When ESP32 is powered up or wakes up from Deep-sleep, a spurious watchdog reset occurs.	0
Section 3.2	When the CPU accesses external SRAM through cache, sometimes random read and write errors occurs.	0
Section 3.3	When the CPU accesses peripherals and writes one address repeatedly, random data loss occurs.	0
Section 3.4	The Brown-out Reset (BOR) function is not functional. The system fails to boot up after BOR.	0
Section 3.5	The CPU crashes when the clock frequency switches from 240 MHz to 80/160 MHz.	0
Section 3.6	The pull-ups and pull-downs for the pads with both GPIO and RTC_GPIO functions can only be controlled by the RTC_GPIO registers. For these pads, the GPIO pull-up and pull-down configuration fields are non-functional.	0/1
Section 3.7	There is a limit on the frequency range for the audio PLL.	0

3. Errata Descriptions and Workarounds

3.1. When ESP32 is powered up or wakes up from Deep-sleep, a spurious watchdog reset occurs.

Description:

The spurious power-up watchdog reset cannot be bypassed with software.

However, when ESP32 wakes up from Deep-sleep, the watchdog reset can be bypassed with software.

Workarounds:

After waking up from Deep-sleep, CPU will read an instruction from the RTC fast memory, and will then execute boot program from this memory. This program needs to clear the illegal access flag in the cache MMU by:

- setting the PRO_CACHE_MMU_IA_CLR bit in DPORT_PRO_CACHE_CTRL1_REG to 1, and then,
- clearing the bit.

3.2. When the CPU accesses external SRAM through cache, sometimes random read and write errors occurs.

Description:

The error cannot be bypassed with software.

For the current version of ESP32, the access to external SRAM by CPU through cache is limited to a one-way operation, that is, write operation only or read operation only. Alternative operations are not available.

Workarounds:

- Clear the pipeline after read operation and then initiate write operation.
- Use MEMW command. add `__asm__( “MEMW” )` command after read operation, and then initiate write operation.

3.3. When the CPU accesses peripherals and writes one address repeatedly, random data loss occurs.

Workarounds:

The FIFO-related addresses and some of the GPIO-related addresses need to be changed as follows:

Registers	Original addresses	Changed addresses
UART_FIFO	0x3ff40000	0x60000000
UART1_FIFO	0x3ff50000	0x60010000
UART2_FIFO	0x3ff6E000	0x6002E000
I2S0_FIFO	0x3ff4F004	0x6000F004
I2S1_FIFO	0x3ff6D004	0x6002D004
GPIO_OUT_REG	0x3ff44004	0x60004004
GPIO_OUT_W1TC_REG	0x3ff4400c	0x6000400c
GPIO_OUT1_REG	0x3ff44010	0x60004010
GPIO_OUT1_W1TS_REG	0x3ff44014	0x60004014
GPIO_OUT1_W1TC_REG	0x3ff44018	0x60004018
GPIO_ENABLE_REG	0x3ff44020	0x60004020
GPIO_ENABLE_W1TS_REG	0x3ff44024	0x60004024
GPIO_ENABLE_W1TC_REG	0x3ff44028	0x60004028
GPIO_ENABLE1_REG	0x3ff4402c	0x6000402c
GPIO_ENABLE1_W1TS_REG	0x3ff44030	0x60004030
GPIO_ENABLE1_W1TC_REG	0x3ff44034	0x60004034

3.4. The Brown-out Reset (BOR) function is not functional. The system fails to boot up after BOR.

Workarounds:

There is no workaround for this issue.

3.5. The CPU crashes when the clock frequency switches from 240 MHz to 80/160 MHz.

Workarounds:

We recommend the following two switching modes:

(1) 2 MHz <-> 40 MHz <-> 80 MHz <-> 160 MHz

(2) 2 MHz <->40 MHz <->240 MHz

3.6. The pull-ups and pull-downs for the pads with both GPIO and RTC_GPIO functions can only be controlled by the RTC_GPIO registers. For these pads, the GPIO pull-up and pull-down configuration fields are non-functional.

Workarounds:

Use RTC_GPIO registers for both GPIO and RTC_GPIO functions.

3.7. There is a limit on the frequency range for the audio PLL.

Description:

For the current version of chip, the frequency formula is as follows:

$$f_{\text{out}} = \frac{f_{\text{xtal}}(sdm2+4)}{2(odiv+2)}$$

For the later versions of the chips with the bug fixed, the frequency formula is:

$$f_{\text{out}} = \frac{f_{\text{xtal}}(sdm2+\frac{sdm1}{2^8}+\frac{sdm0}{2^{16}}+4)}{2(odiv+2)}$$

Espressif IOT Team

www.espressif.com