Introduction

This document describes known errata in ESP32-C6 series of SoCs.
Contents

Chip Identification
1 Chip Revision 3
2 Additional Methods 5

Errata Description
3 RISC-V CPU 7
  3.1 Possible deadlock due to out-of-order execution of instructions when writing to LP SRAM is involved 7
4 Clock 8
  4.1 Inaccurate Calibration of RC_FAST_CLK Clock 8
5 Reset 8
  5.1 System Reset Triggered by RTC Watchdog Timer Cannot be Correctly Reported 8
6 RMT 8
  6.1 The idle state signal level might run into error in RMT continuous TX mode 8
7 Wi-Fi 9
  7.1 ESP32-C6 Cannot be 802.11mc FTM Initiator 9

Related Documentation and Resources 10

Revision History 11
1 Chip Revision

Espressif is introducing vM.X numbering scheme to indicate chip revisions.

M – Major number, indicating the major revision of the chip product. If this number changes, it means the software used for the previous version of the product is incompatible with the new product, and the software version shall be upgraded for the use of the new product.

X – Minor number, indicating the minor revision of the chip product. If this number changes, it means the software used for the previous version of the product is compatible with the new product, and there is no need to upgrade the software.

The vM.X scheme replaces previously used chip revision schemes, including ECOx numbers, Vxxx, and other formats if any.

The chip revision is identified by:

- eFuse field EFUSE_RD_MAC_SPI_SYS_3_REG[23:22] and EFUSE_RD_MAC_SPI_SYS_3_REG[21:18]

<table>
<thead>
<tr>
<th>eFuse Bit</th>
<th>v0.0</th>
<th>v0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFUSE_RD_MAC_SPI_SYS_3_REG[23]</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EFUSE_RD_MAC_SPI_SYS_3_REG[22]</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Minor Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFUSE_RD_MAC_SPI_SYS_3_REG[21]</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EFUSE_RD_MAC_SPI_SYS_3_REG[20]</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EFUSE_RD_MAC_SPI_SYS_3_REG[19]</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EFUSE_RD_MAC_SPI_SYS_3_REG[18]</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Espressif Tracking Information line in chip marking
Table 2: Chip Revision Identification by Chip Marking

<table>
<thead>
<tr>
<th>Chip Revision</th>
<th>Espressif Tracking Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.0</td>
<td>XAxxxxxxxxxx</td>
</tr>
<tr>
<td>v0.1</td>
<td>XBxxxxxxxxxx</td>
</tr>
</tbody>
</table>

- **Specification Identifier** line in module marking

---

Figure 1: Chip Marking Diagram

Figure 2: Module Marking Diagram
Table 3: Chip Revision Identification by Module Marking

<table>
<thead>
<tr>
<th>Chip Revision</th>
<th>Specification Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.0</td>
<td>XAXXXX</td>
</tr>
<tr>
<td>v0.1</td>
<td>MBXXXX</td>
</tr>
</tbody>
</table>

Note:
- Information about ESP-IDF release that supports a specific chip revision is provided in Compatibility Between ESP-IDF Releases and Revisions of Espressif SoCs.
- For more information about the chip revision upgrade and their identification of ESP32-C6 series products, please refer to ESP32-C6 Product/Process Change Notifications (PCN).
- For more information about the chip revision numbering scheme, see Compatibility Advisory for Chip Revision Numbering Scheme.

2 Additional Methods

Some errors in the chip product don’t need to be fixed at the silicon level, or in other words in a new chip revision.

In this case, the chip may be identified by Date Code in chip marking (see Figure 1). For more information, please refer to Espressif Chip Packaging Information.

Modules built around the chip may be identified by PW Number in product label (see Figure 3). For more information, please refer to Espressif Module Packaging Information.
### Module Product Label

- **PW Number**: PW-2020-11-0001
- **Product Name**: ESP32-WROOM-32D
- **Product Number**: M21EH3264PH3Q0
- **Quantity**: 650 pcs
- **Country of Origin**: MADE IN CHINA
- **Seal Date**: 2020-11-30
- **Lot Numbers**:
  - 202048-000001
  - 202048-000002
  - 202048-000003
  - 202048-000004
  - 202048-000005

---

**Note:**

Please note that **PW Number** is only provided for reels packaged in aluminum moisture barrier bags (MBB).
### RISC-V CPU

#### 3.1 Possible deadlock due to out-of-order execution of instructions when writing to LP SRAM is involved

**Description**

When HP CPU executes instructions (instruction A and instruction B successively) in LP SRAM, and instruction A and instruction B happen to follow the following patterns:

- Instruction A involves writing to memory. Examples: `sw/sh/sb`
- Instruction B involves only accessing the instruction bus. Examples: `nop/jal/jalr/lui/auipc`
- The address of instruction B is not 4-byte aligned

The data written by instruction A to memory is only committed after instruction B has completed execution. This introduces a risk where, after instruction A writing to memory, if an infinite loop is executed in instruction B, the writing of instruction A will never complete.

**Workarounds**

When you experience this problem, or when you check the assembly code and see the above mentioned pattern,

- Add a `fence` instruction between instruction A and the infinite loop. This can be achieved by using the `rv_utils_memory_barrier` interface in ESP-IDF.
- Replace the infinite loop with instruction `wfi`. This can be achieved by using the `rv_utils_wait_for_intr` interface in ESP-IDF.
- Disable the RV32C (compressed) extension when compiling code that to be executed in LP SRAM to avoid instructions with not 4-byte aligned addresses.
4 Clock

4.1 Inaccurate Calibration of RC_FAST_CLK Clock

Description
In the ESP32-C6 chip, the frequency of the RC_FAST_CLK clock source is too close to the reference clock (40 MHz XTAL_CLK) frequency, making it impossible to calibrate accurately. This may affect peripherals that use RC_FAST_CLK and have stringent requirements for its accurate clock frequency.

For peripherals using RC_FAST_CLK, please refer to [ESP32-C6 Technical Reference Manual > Chapter Reset and Clock].

Workarounds
Use other clock sources instead of RC_FAST_CLK.

Solution
Fixed in chip revision v0.1.

5 Reset

5.1 System Reset Triggered by RTC Watchdog Timer Cannot be Correctly Reported

Description
When the RTC watchdog timer (RWDT) triggers a system reset, the reset source code can not be latched correctly. As a result, the reset cause reported is indeterminate and might be wrong.

Workarounds
No workaround.

Solution
Fixed in chip revision v0.1.

6 RMT

6.1 The idle state signal level might run into error in RMT continuous TX mode

Description
In ESP32-C6's RMT module, if the continuous TX mode is enabled, it is expected that the data transmission stops after the data is sent for RMT_TX_LOOP_NUM_CHANNELS rounds, and after that, the signal level in idle state should be controlled by the “level” field of the end-marker.
However, in real situation, after the data transmission stops, the channel's idle state signal level is not controlled by the “level” field of the end-marker, but by the level in the data wrapped back, which is indeterminate.

**Workarounds**

Users are suggested to set `RMT_IDLE_OUT_EN_CHn` to 1 to only use registers to control the idle level.

This issue has been bypassed since the first ESP-IDF version that supports continuous TX mode (v5.1). In these versions of ESP-IDF, it is configured that the idle level can only be controlled by registers.

**Solution**

No fix scheduled.

---

**7 Wi-Fi**

**7.1 ESP32-C6 Cannot be 802.11mc FTM Initiator**

**Description**

The time of T3 (i.e. time of departure of ACK from Initiator) used in 802.11mc Fine Time Measurement (FTM) cannot be acquired correctly, and as a result ESP32-C6 cannot be the FTM Initiator.

**Workarounds**

No workaround.

**Solution**

To be fixed in the future chip revisions.
Related Documentation and Resources

Related Documentation

- **ESP32-C6 Series Datasheet** – Specifications of the ESP32-C6 hardware.
- **ESP32-C6 Hardware Design Guidelines** – Guidelines on how to integrate the ESP32-C6 into your hardware product.
- **Certificates**
- **ESP32-C6 Product/Process Change Notifications (PCN)**
- **Documentation Updates and Update Notification Subscription**

Developer Zone

- **ESP-IDF** and other development frameworks on GitHub.
  - [https://github.com/espressif](https://github.com/espressif)
- **ESP32 BBS Forum** – Engineer-to-Engineer (E2E) Community for Espressif products where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
  - [https://esp32.com/](https://esp32.com/)
- **The ESP Journal** – Best Practices, Articles, and Notes from Espressif folks.
  - [https://blog.espressif.com/](https://blog.espressif.com/)
- **See the tabs** SDKs and Demos, Apps, Tools, AT Firmware.

Products

- **ESP32-C6 Series SoCs** – Browse through all ESP32-C6 SoCs.
- **ESP32-C6 Series Modules** – Browse through all ESP32-C6-based modules.
- **ESP32-C6 Series DevKits** – Browse through all ESP32-C6-based devkits.
- **ESP Product Selector** – Find an Espressif hardware product suitable for your needs by comparing or applying filters.

Contact Us

- **See the tabs** Sales Questions, Technical Enquiries, Circuit Schematic & PCB Design Review, Get Samples (Online stores), Become Our Supplier, Comments & Suggestions.
# Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Release Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023-11-14</td>
<td>v1.0</td>
<td>First release</td>
</tr>
</tbody>
</table>