TMDSEVM6474L Known Issues

1. **Potential SERDES Clocking issue in TMS320C6474**

   - **TI Document Reference**: SPRZ283B–October 2008–Revised October 2009
   - **DSP Silicon Revision(s) Affected**: 2.1, 1.3, 1.2

   Details: A bug has been found in the SerDes interfaces of TMS320C6474 that causes a SerDes clocking problem in normal functional operation. This problem will not occur when external pull-down is applied on the TCK pin (JTAG controller clock). SerDes are used in the Ethernet interface (EMAC), Serial RapidIO® interface (SRIO) and the Antenna Interface (AIF).

   The TCK pin (JTAG controller clock) is internally assigned to an internal signal that is used by the SerDes macro. For the SerDes macro to get proper clocking in the normal functional operation, it needs the internal signal to be held low. However, there is an internal pull-up on the TCK, creating problems for SerDes operation. This problem exists on all SerDes interfaces.

   In TMDSEVM6474L this issue may cause EMAC loop back test to fail randomly during Power On Self Test (POST).

   <u>Workaround in TMDSEVM6474L Rev 2.0 and 1.0:</u>

   A pull down is needed on TCK, to do that:

   1. Un-mount pull up resistor R94 (4.7K) from net TG_TCK. Refer below figure for the location of R94.

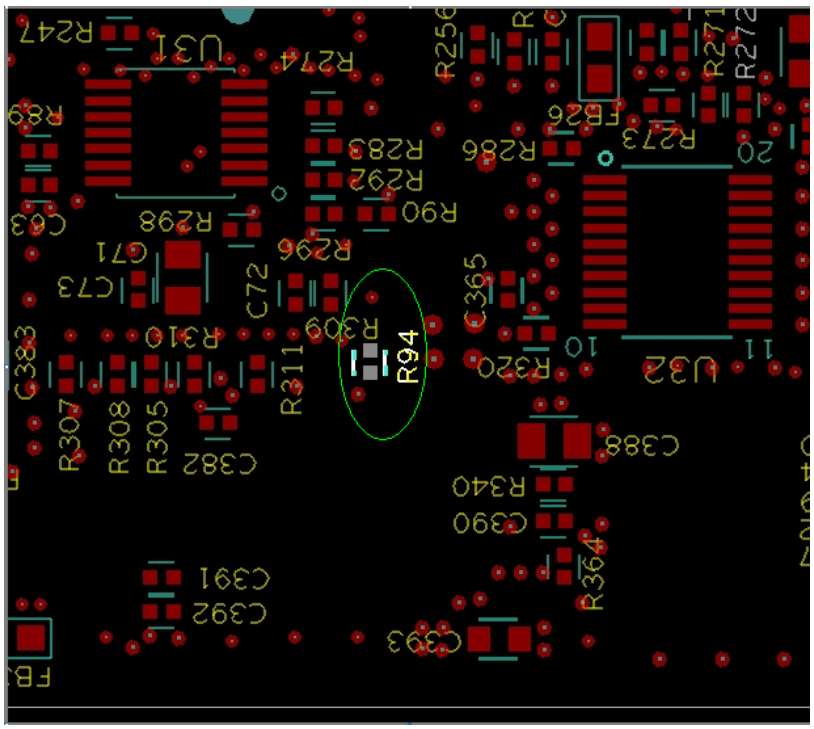

   Figure 1. Location of R94 -- Bottom side of PCB near U32

2. Add 1K pull down resistor on net TCK (U32.20). Refer below figure for the location of TCK net and nearest GND node for the addition of pull down resistor between them.

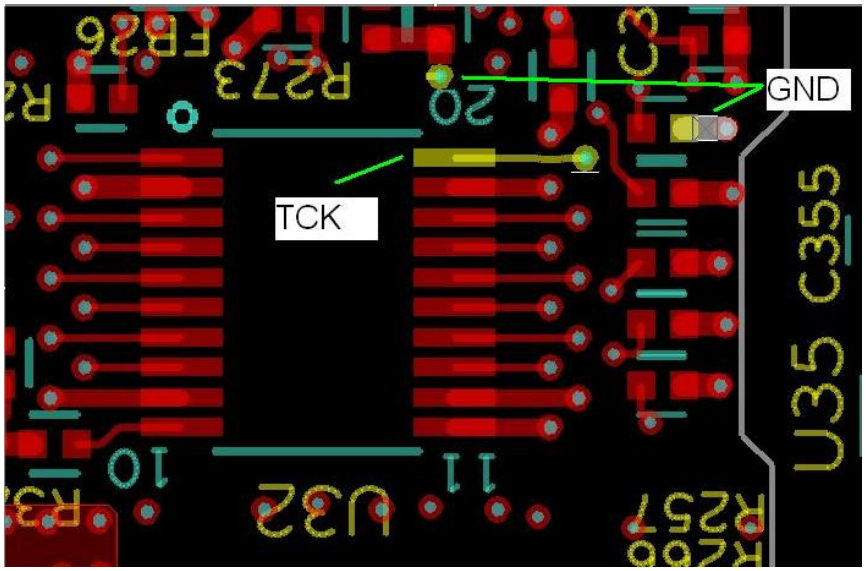

Figure 2. Location of TCK (U32.20) net and nearest GND nodes -- Bottom side of PCB

3. Add 4.7K pull-down resistor on net EM_TCK. Refer below figure for the location of EM_TCK net and nearest GND node for the addition of pull down resistor between them

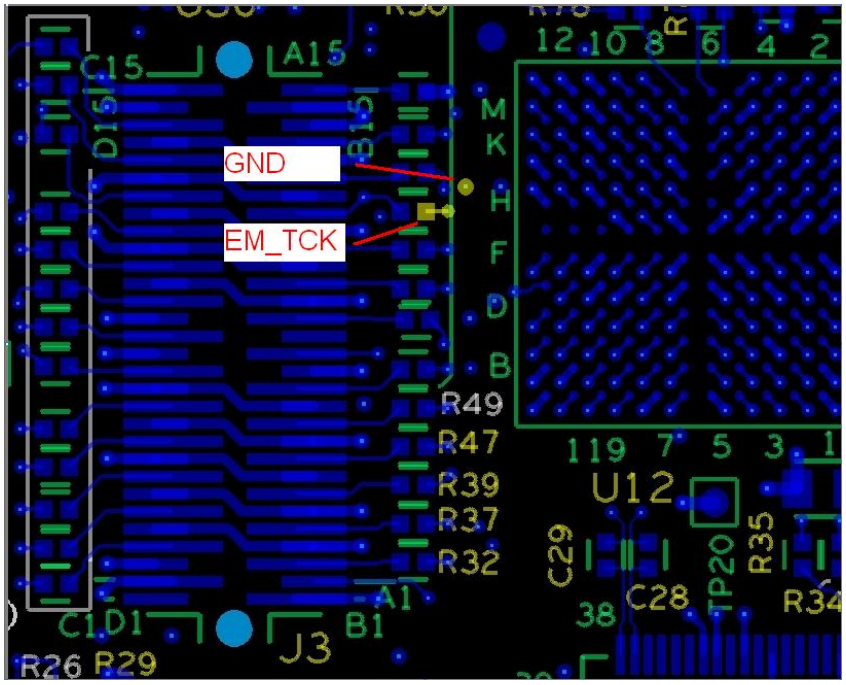

Figure 3. EM_TCK net and nearest GND node -- Top side of PCB near 60 pin connector J3

2. **Version number displayed on UART console during POST execution**

Details: During POST execution the board version displayed on the UART console does not match the PCA/PCB major revision indicated on the board silk.

The version displayed by the POST is read from the FPGA, which reflects the board version set using the assembly of resistors shown below:

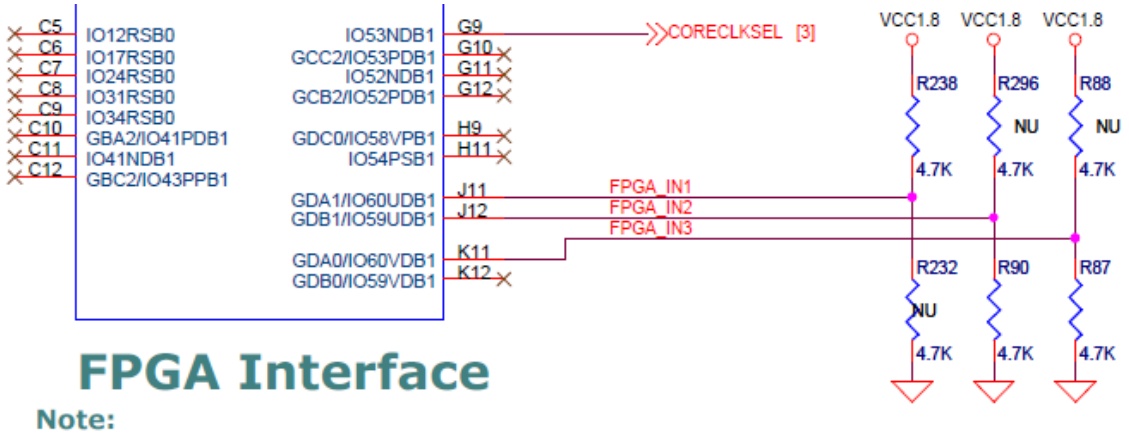

Figure 1: Board version resistor assembly

The following table shows the mapping of PCB/PCA major version and the board build Identification version:

[K11:J12:J11]	Description
000	Proto Batch, PCB Rev 01
001	Production Batch, PCB Rev 02
010 - 111	Reserved for future use

Figure 2: Board build identification table

3. **Intermittent DDR2 data corruption**

Details: If the board is booted in 'no boot' mode and JTAG is used to connect only core 0 of the DSP then the DDR2 data may get corrupted during code execution.

The same behavior is observed when only core 0 of DSP is booted using any other boot mode.

This happens as core 1 and core 2 are not in IDLE state or halted.

Solution:

Connect all the three cores of the DSP by JTAG when using 'no boot' mode.

When booting only core 0 of DSP through other bootmodes, load IDLE instruction at L2 base location of core 1 and core 2. To do this, add the following lines in the C code:

```
#pragma DATA_SECTION(idle_c1, ".idle_c1")
const unsigned int idle_c1 = 0x0001e000;   /* This is an idle instruction */
#pragma DATA_SECTION(idle_c2, "idle_c2")
const unsigned int idle_c2 = 0x0001e000;   /* This is an idle instruction */
```

Then in the linker command file add the following lines to the MEMORY area:

```
CORE_1   :  origin = 0x11800000,  length = 4
CORE_2   :  origin = 0x12800000,  length = 4
```

And then in the SECTIONS area add this:

```
.idle_c1 > CORE_1
.idle_c2 > CORE_2
```

This will make an idle instruction appear at the base of L2 for cores 1 and core 2.