**Bug-6 Report**

**Group-5**

1. **Failing Test name**

Data Cache MESI test

Sequence : M\_S\_I\_S

1. **Test description (Describe the planned scenario and the expected result)**

Main Processor (mp) Writes specified data to specified address, thus getting that block in M state.

Side Processor-1 (sp1) issues a read request, to the same address, thus getting it in S state.

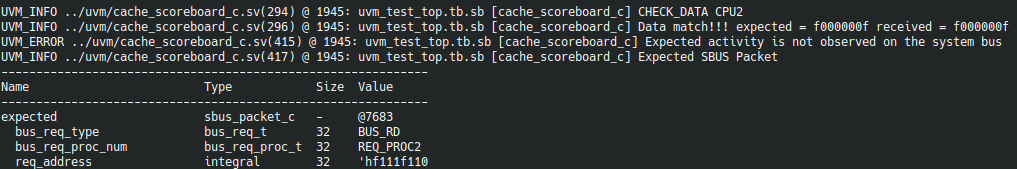
Sp1 writes to this block, which should become M for sp1 and I for mp

Sp1 reads the block again to make sure write occurred properly

Mp reads the block to check if Invalidation occurred properly and make it shared again.

1. **Failing assertion that helped you identify the bug**

No assertion failed, but the second read by Sp1 returned the same value as the first read, indicating that the write was NOT successful.

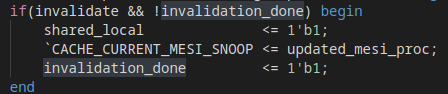


1. **Debugging:**

Waveform was observed, and it was found that cpu\_wr\_done was never being asserted, thus affirming that the write was indeed not getting completed. Since invalidation is the major component of this process, it was examined and found that even though invalidate was being asserted, the all\_invalidation\_done was never given a value.



This was happening because invalidation\_done was always high impedance for all 4 caches. The code was checked and the following condition was found for assigning this signal (at two places):

****

Thus if invalidation\_done was initialized to Z, it would never change its value. It was found that this was indeed the case.

1. **Erroneous RTL file name**

main\_func\_lv1\_dl.sv

1. **Lines of RTL file responsible for the bug**

138. invalidation\_done <= 1'bz;

1. **Corrected RTL code**

138. invalidation\_done <= 1'b0;

Waveform after Implementing Fix:

