UBC CHESS-2 UPDATE

OUR PROBLEM:

• 3 weeks ago, we were having trouble getting test-mode to work when we added a second chipscope at the next point in the pipeline.

• After rebuilding the project, this problem appeared to be fixed, which led us to hypothesize that this bug was due to timing violations which were randomly fixed/broken from build to build.

FORCE IOB

- Initially we thought that the location of the SACI output registers were causing the timing issue
- We attempted to fix this by forcing them to IOB registers
  - Something about our current I/O configuration causes ISE to reject this
    - We are using 'inout' I/O busses (ITSDAQ practice due to multitude of supported HW)
  - This issue is still a work in progress – the next thing to try specify parts of the busses as 'in' xor 'out'
- It turns out that this is not the source of our problem, however we will go ahead with this since it is best practice to have your outputs in the IOB's

The True Source of the Problem

- Upon investigating SACI with a chipscope, we noticed that some commands weren't receiving responses, which was causing SACI 'op code block' to timeout.
- Even on the 2-chipscope-setup where test mode was working, we discovered that some configuration commands were timing out (this was semi-random between different builds)

NO TIMEOUT   TIMEOUT
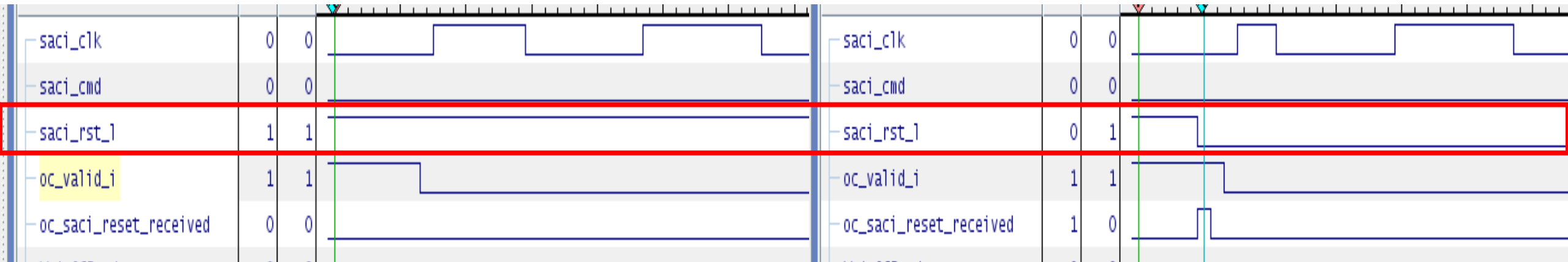
- It turns out that for some commands, SACI op code block believed it was receiving a reset and setting the reset bit to it's active state (low)
- The first error was that SACI should not have thought it was receiving a reset bit, and the second error was that when it did think it received that, it should have been going to the Reset state which would eventually set reset to high, and CHESS-2 would respond.
- However since saci_rst_l was stuck low, SACI would continuously be receiving a reset signal, never respond, and op code block state machine would eventually timeout

Fix

- We are not 100% clear why SACI ocb thought it was receiving a reset signal, however we managed to fix the bug by changing a bit of the code in ocbsaci state machine preamble (defaults):
  - Originally, the default value for the reset line in the state machine was

    saci_rst_l <= saci_rst_l

    which we changed to

    saci_rst_l<='1'

- We believe that the original line was causing the interpreter to infer a latch, and that was throwing off the synchronicity of the reset line

- We are no longer having problems with commands timing out

Question for JJJ

- You are presumably using this same code for analog board work – have you had any problems with SACI timing out (or did you change this at some point as well)?