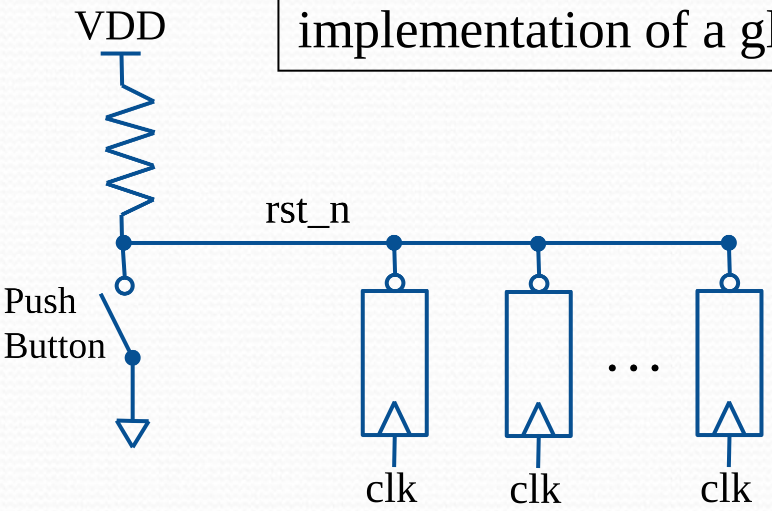
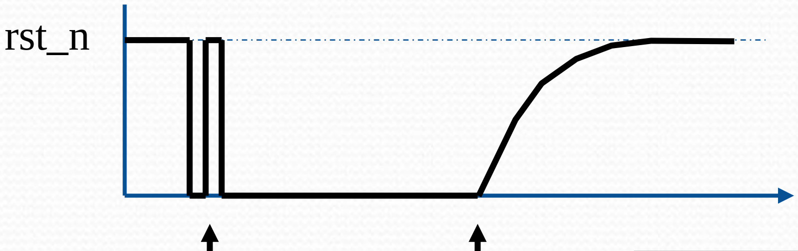
Exercise 20: Testing inert_intf on DE0 & Other

- On the FPGA board we have a couple of push buttons. We will use one as the source for our asynchronous reset.

- It is simply a momentary push button switch to ground with a pull-up resistor.

Imagine what a disaster this implementation of a global reset would be!!





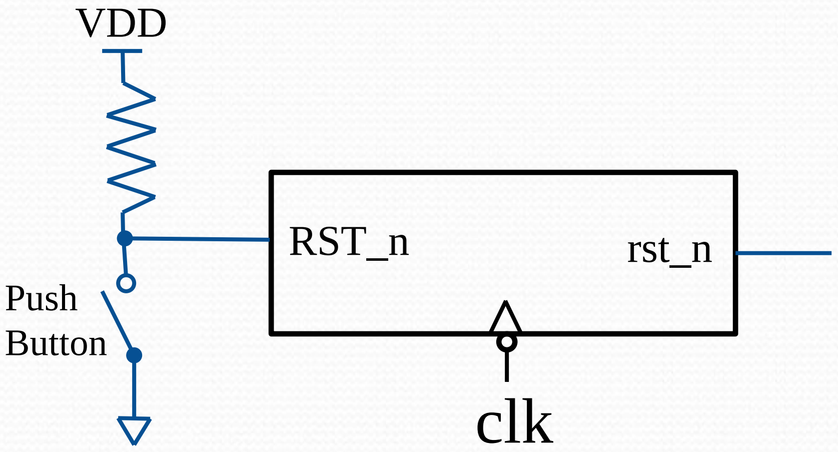
Button pushed (has bounce)

Button released (RC approach)

When does reset deassert relative to clock?

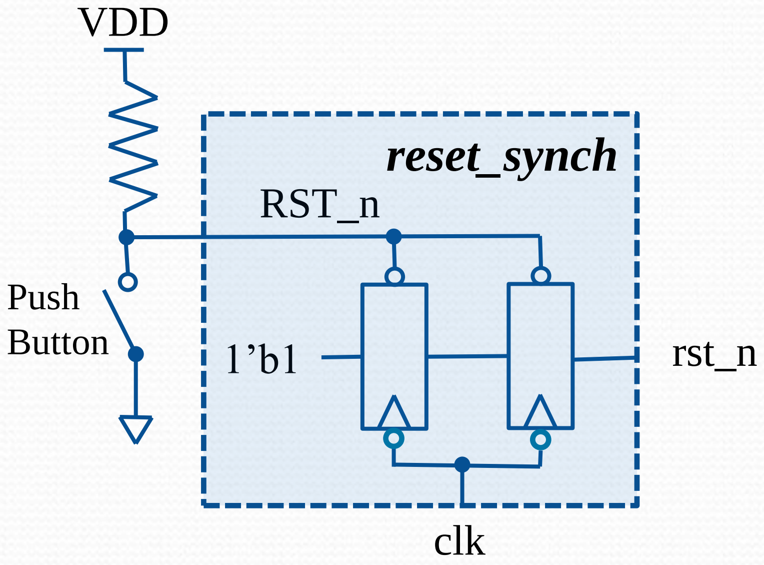
Remember…you want your reset de-asserted on the opposite edge of clock that your other flops are active on. This means we want our reset to de-assert (rise) on **negative** edge of clock.

Exercise 20: Testing Inert_intf on DE0 & Other



We want to build a reset synchronizer that takes in the raw push button signal and creates a signal that is deasserted at the negative edge of clock.

It will have an interface of:
RST_n = raw input from push button
clk = clock, and we use negative edge
rst_n = our syncrhronized output which will form the global reset to the rest of our chip.

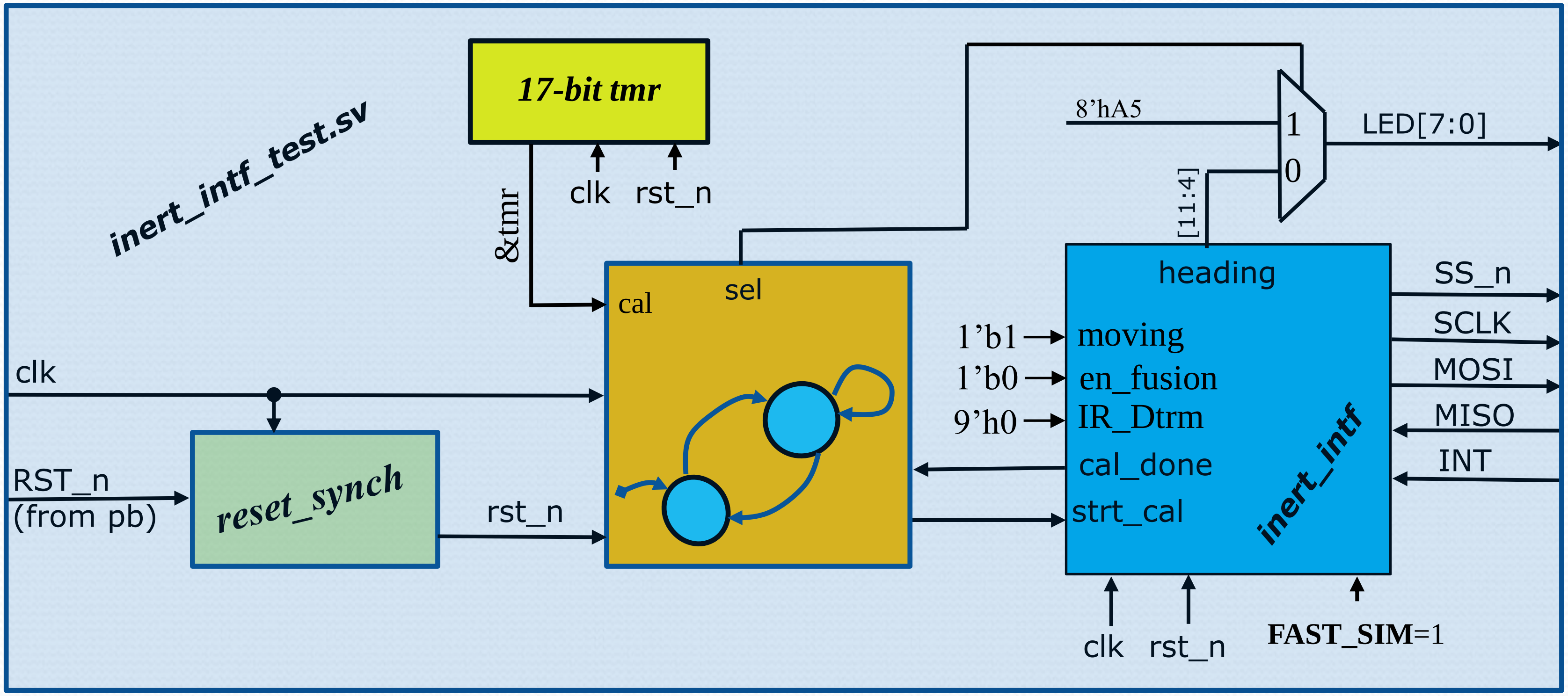


Push of the button will asynch reset the two flops. When button is released we have a double flopping (meta-stability reasons) to produce our global **rst_n**. The flops are negative edge triggered so our global reset will deassert on the opposite edge of all our other flops.

Code this reset synch unit (**reset_synch.sv**)

Exercise 20: Mapping inert_intf to DE0 & Other

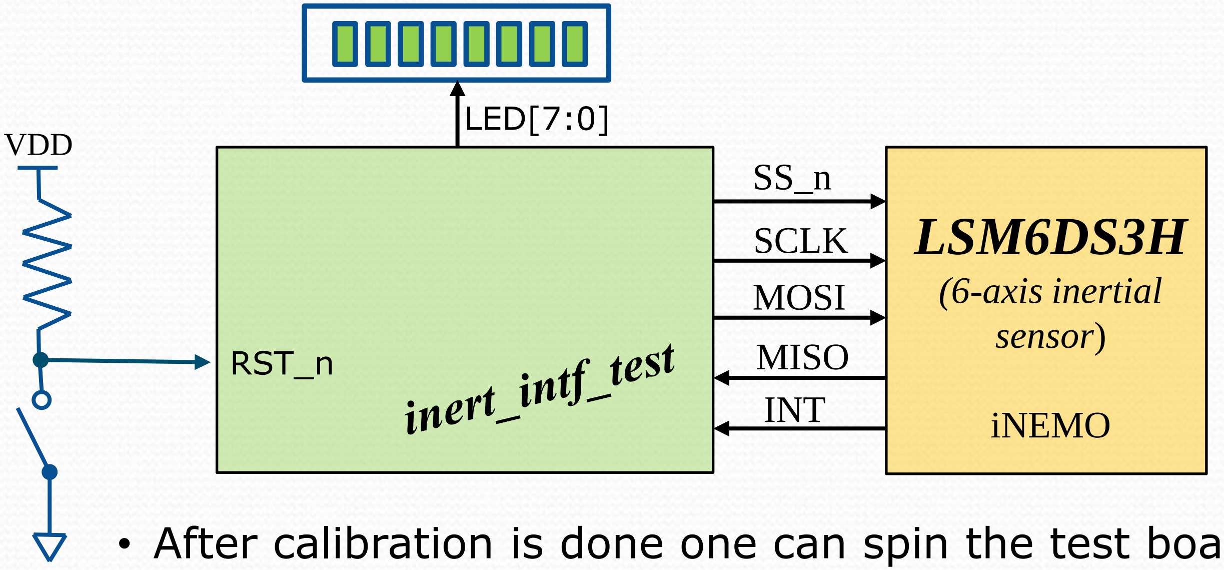
- In exercise 19 you finished **inert_intf.sv** and tested it. Now you will map it to the DE0-Nano board so it can be tested "for real".



- The SM simply waits for a 17-bit free running timer to fill then asserts **strt_cal**, and waits for **cal_done**. While in calibration is asserts **sel** so the constant 0xA5 is displayed on the LEDs. Once calibration is over the LEDs will display the upper 8-bits of heading.

Exercise 20: Mapping inert_intf_test to DE0 Nano

Signal:	Dir:	Description:
clk	in	50MHz clock
RST_n	in	Unsynchronized input from push button
LED[7:0]	out	Flopped upper 8-bits of heading
SPI Intf	out/ in	The SS_n, SCLK, MOSI, MISO and INT of SPI interface to 6-axis inertial sensor

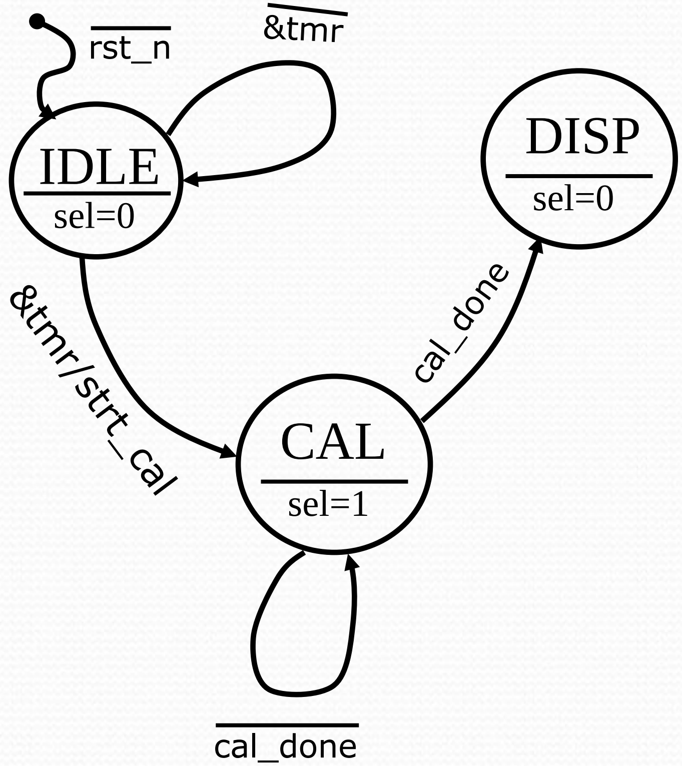


- After calibration is done one can spin the test board and check that the heading LEDs change as expected.

Exercise 20: Mapping inert_intf_test to DE0 Nano

- The state machine is simple. After reset it waits for a 17-bit free running timer to fill then asserts **strt_cal** and waits for **cal_done**. While in calibration it selects 0xA5 to be displayed on the LEDs.

- Once calibration is completed it simply displays bits [11:4] of heading on the LEDs. Only a reset can get it out of DISP state.


$\overline{\text{rst\_n}}$
$\overline{\text{\&tmr}}$
IDLE
sel=0
DISP
sel=0
&tmr/strt_cal
cal_done
CAL
sel=1
$\overline{\text{cal\_done}}$

- Create **inert_intf_test.sv**

- Test it in ModelSim

- Ensure it compiles in Quartus

Exercise 20: Mapping inert_intf_test to DE0 Nano

- There are Quartus project file and settings file available for download: (**inert_intf_test.qpf, inert_intf_test.qsf**).

- Open the .qpf and **ensure you add** all necessary files to the project.

- Ensure the project builds with no errors

- Once your project build in Quartus call Eric, Harish, or Khailini over to demo on a DE0 test platform they will have.