

# SPI Slave with Single Port RAM

Project Description can be found in this [link](#).

- Create a constraint file where the rst\_n , SS\_n & MOSI are connected to 3 switches, and the MISO to a led.
- The SPI slave implementation is done using FSM, we shall try out three different encoding (gray, one\_hot or seq)
- We wish to operate at the highest frequency possible and so you shall choose the encoding based on the best timing report that gives the highest setup/hold slack after implementation.
- After choosing the best encoding, add a debug core such that all internals (MISO, MOSI, SS\_n, rst\_n & clk) can be analyzed and then generate a bitstream file.

**Note:** The FSM and RAM in the project description is using synchronous active low reset.

## Deliverables:

- 1) I am expecting a **rar** file with this format <your\_name>\_Project2 for example Kareem\_Waseem\_Project2 having the design files, testbench file, do file, constraint file, netlist generated, bitstream file and a PDF file
  - Testbench can be directed to verify the different state transitions through writing and reading from the RAM through the SPI interface.
- 2) Write in the PDF file the group member names. **Please use the same certificate names [here](#).**

**Note:** I will use the do file written from your side to run the simulation on QuestaSim so make sure that they are correct and functionable.

The PDF file will have snippets of the following:

- 1) Snippets from the waveforms captured from QuestaSim for the design with inputs assigned values and output values visible.
- 2) Synthesis snippets for each encoding
  - Schematic after the elaboration & synthesis
  - Synthesis report showing the encoding used
  - Timing report snippet
  - Snippet of the critical path highlighted in the schematic
- 3) Implementation snippets for each encoding
  - Utilization report
  - Timing report snippet
  - FPGA device snippet
- 4) Snippet of the “Messages” tab showing no critical warnings or errors after running elaboration, synthesis, implementation and a successful bitstream generation.