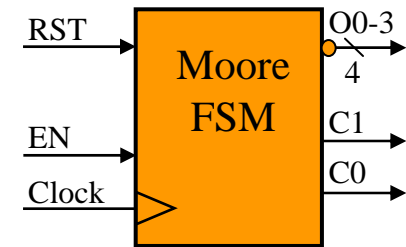


Sequential Logic Design

- Design a Moore FSM with
 - 4 states (S0-S3)
 - Active high synchronous reset (RST)
 - Resets to the S0
 - Active RST takes precedence over other operations
 - Active high synchronous enable (EN)
 - When EN=0, FSM holds its state (unless reset)
 - When EN=1, FSM cycles through states on active edge of clock
 - ✓ State sequence: $S0 \Rightarrow S1 \Rightarrow S2 \Rightarrow S3 \Rightarrow S0$ and so on
 - 4 active low outputs O0-3, one output for each state
 - The output $O_i = 0$ while the FSM is in the corresponding state S_i
 - 2 outputs (C1 and C0) giving the binary value of the current state
 - S0=00, S1=01, S2=10, S3=11 (C1 is MSB)
- Capture schematic, simulate and verify (debugging where needed) design using ISE and ModelSim
- Synthesize, download, and verify design on Spartan 3 FPGA

block diagram



Sequential Logic Design

- Pre-lab assignment:
 - Drive State Diagram & State Table
 - You may use any combination of D and JK flip-flops you like
 - ✓ These FFs are denoted “fd” and “fjkc” in ISE schematic library
 - » Note: tie the clear input of the JK FF to a logic 0 (since it is an asynchronous clear)
 - Use K-maps to obtain minimized SOP expressions for flip-flop inputs
 - You may share common gates where possible
 - ✓ If you are “at-one” with your Boolean Algebra
 - » Otherwise stick with SOP
 - Draw logic diagram
 - Sharing common gates when possible
 - ✓ Again, only if you are “at-one” with your BA
 - Label all I/O according to system specifications
 - Read tutorials (see next page)
- Show pre-lab work to GTA at beginning of lab session

Tutorials

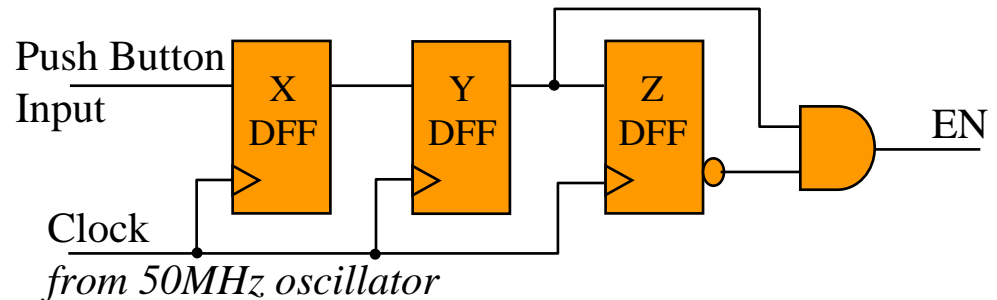
- Read the following:
 - Spartan 3 PCB reference manual
 - Chapter 4 (2 pages) push buttons
 - ✓ You will need to select a push button for EN so you will need to know the pin numbers
 - Chapter 8 (1 page) clock sources
 - ✓ You will need to connect to the 50MHz oscillator for your clock input so, again, you will need to know the pin number for the oscillator
- You may also need to review FSM design in your ELEC 2200 textbook
 - If you are not “at-one” with FSMs

Sequential Logic Design

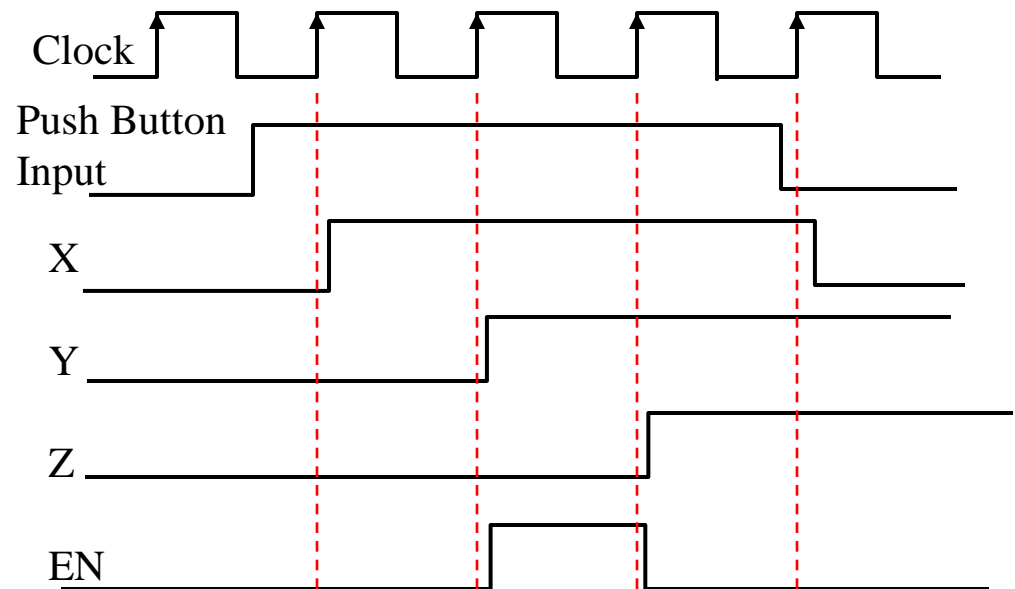
- Lab exercise:
 - Capture your design using schematic entry
 - Simulate your circuit for design verification
 - Simulate & verify various possible input sequences
 - ✓ Including a complete counting sequence with roll-over
 - Debug & fix problems when output is incorrect
 - ✓ Check state diagram and state table
 - ✓ Check state table against K-map population
 - ✓ Check K-map groups against logic equation product terms
 - ✓ Check logic equations against schematic
 - Once you have verified your circuit via simulation add the digital one-shot on the following page for the Enable input (EN) and resimulate
 - This will help to reduce (but not eliminate) multiple clock cycles due to switch bounce

Digital One-Shot

- Push button and dip switches “bounce” and cannot be used for generating a clock
 - Multiple clock pulses will be produced with each push of button due to the high electrostatic field as the switch contacts are near each other when both making and breaking contact
- A digital one-shot should provide a single enable pulse for each push of the button
 - See timing diagram



This is same clock that will be connected to all other FFs



Sequential Logic Design

- Lab exercise continued:
 - Synthesize your design for the Spartan 3S200 FPGA using ISE
 - Connect inputs to switches and push button (for EN)
 - ✓ Connect the Clock input to the 50 MHz oscillator
 - Connect outputs to LEDs
 - Download and verify your design
 - Debug, re-synthesize, and re-download as needed
 - Demonstrate your working circuit to the GTA
- Post-lab: Turn in your lab report at beginning of next lab session
 - Include: verified schematic, simulation results, and all pre-lab work