AUBURN

UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

ELEC 4200 Lab#2

Sequential Design Using Logic Equations

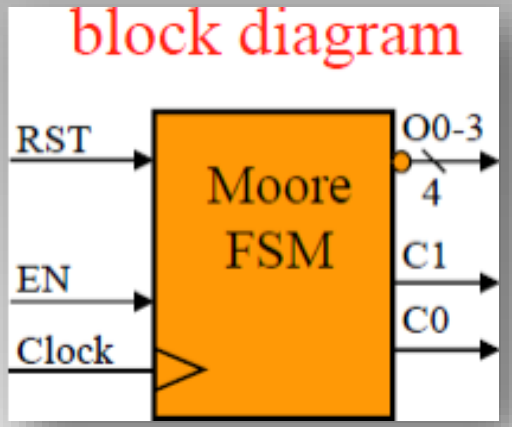
- References you may need:
  - **Nexys4-DDR_rm.pdf**
  - **Lab #0 Tutorial**

AUBURN
UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

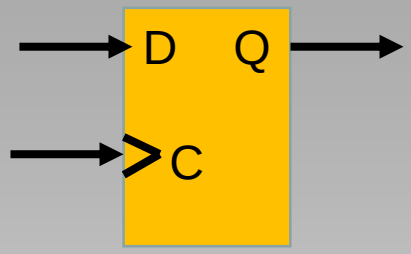
Overview

- Design a Moore finite state machine (FSM) with:
  - 4 states (S0-S3)
  - Active high synchronous reset (RST)
    - » Resets the FSM to state S0
    - » Takes precedence over all other operations
  - Active high synchronous enable (EN)
    - » EN=0 → Hold state (unless reset)
    - » EN=1 → Cycle to next state on active clock edge
      - S0 → S1 → S2 → S3 → S0
  - 4 active LOW outputs O0-O3.
    - » One output for each state
    - » i.e: Oi = 0 when FSM is in state Si
  - 2 outputs C1-C0 giving the binary value of the current state
    - » C1 is the most significant bit (MSB)



Pre-lab Assignment

- Derive the following
  - State Diagram
  - State Table



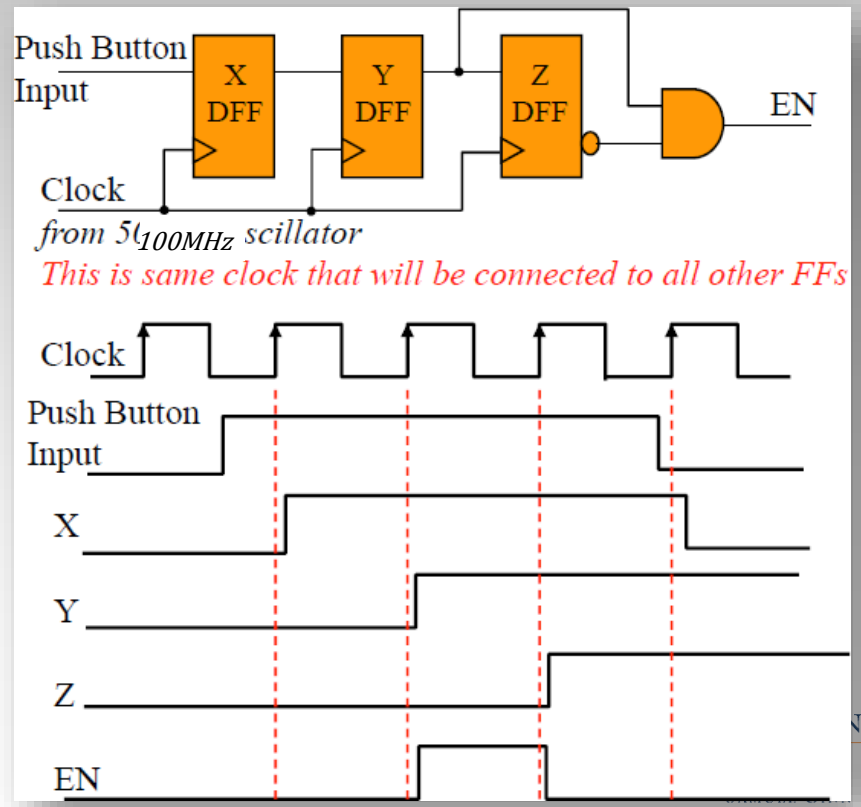
- Design the circuit using D flip-flops
  - The UNISIM library contains a D flip-flop component “FD”
- Use K-maps to obtain minimized SOP expressions
- Draw a logic diagram
  - Share common product terms and gates if possible
  - Label all inputs and outputs
- Review the following in the Nexys4 reference manual
  - Chapter 6 (Clock sources)
    - » You will be using the 100MHz clock source.
  - Chapter 10

Lab Exercise

- Capture your design as a VHDL model in *Vivado*.
  - Try to be neat and compact as we will be adding additional circuitry to the model.
  - Note that the UNISIM library must be included, so that FD components can be instantiated in the model.
- Simulate your circuit in *Active-HDL*
  - Your simulation should test every state of the FSM and each feature (next state, reset)
  - Fix any incorrect outputs
- Once your circuit works, add the digital one-shot discussed on the following page to the Enable(EN) input

Digital One-Shot

- Why a digital one-shot?
  - The clock on the board is pulsing at a rate of 100Mhz and any time the enable button is pressed over a clock edge, the FSM will move to the next state. Unless you can press and release the button in 1/100000000th of a second (not to mention switch bouncing) you will fly through all of your 4 states. (If all you want is flickering LEDS and to get a "0"—leave it off)
  - The digital one-shot will provide a single enable pulse for each push of the push-button as shown in the timing diagram.
  - Study the circuit and the diagram to be sure you understand its function.



Synthesize

- Synthesize and implement your design for the Artix-7 FPGA on the Nexys4 board.
  - RST → Switch or Push-button
  - EN → Connected to push-button via the one-shot
    - » With the one-shot, the Enable signal is now an internal net and should be connected to the one-shot output. The input of the one-shot should be connected to the push-button
  - O0-O3 → LED's
  - C1-C0 → LED's
- Download and verify your design on the Nexys4 board.
  - Fix any errors
  - Demonstrate to the GTA

Report Guidelines

- Be sure to include all sections required by the lab manual guidelines. In addition be sure your report includes the following:
  - Screenshot of you verified schematic (Be sure labels are legible)
  - Annotated screenshot of your Aldec Active-HDL simulation results
    - Be sure to **describe** your testing method
  - Design work (truth-tables, k-maps, equations, etc)
  - Answers to the following questions…

1. What method did you use to encode the states in the state machine? Some of the possible methods you could have used were binary, gray code, one hot, or one cold.

2. How did this affect the resource usage on the FPGA? What effect would using an alternate encoding method have had on the resource usage of the FPGA?