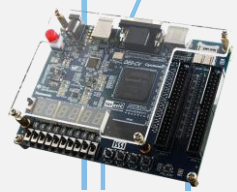
FSM PRIMER

FPGA/VERILOG FROM SCRATCH

DAY 3

L3.1: IMPLEMENTATION OF FSM DESIGN

- **Creating a new project via System Builder Tool**
- **Asynchronous D Flip-Flop and Counter Implementation**
- **An Example: FSM Problem Solving**
  - Understanding problem and solving on paper
  - Implementation on FPGA
  - Simulation via University Vector Waveform
  - Running on DE0-CV Boad



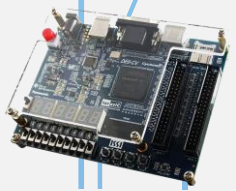
EXPERIMENTS

PART 1

1. Draw the FSM, and make sure you included all possible actions
2. Create architecture
3. Encode the states
4. Create the state table
5. Write boolean expressions and simplify them

PART 2

1. Implement a custom clock by writing a 32-bit counter and using one of its bits
2. Implement a state register
3. Implement combinational part of your design
4. Assign the state bits to the LEDs as well as the outputs of the designed circuit.
5. Show the results on FPGA



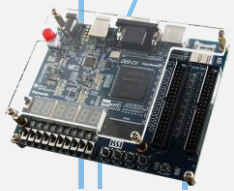
2-BIT FORWARD-BACKWARD COUNTER CIRCUIT

- The designed circuit will have two inputs and three outputs.
- One input will increase the current number value while the other will decrease the number.
- Two output bits will represent the current number, and the other output will be ON if the number is odd.
- Use the buttons as inputs of the circuit, and the LEDs as the outputs of the circuit.

Note: Don't forget the buttons on DE0 board are ON when you don't press.

4-BIT PRIME NUMBER COUNTER CIRCUIT

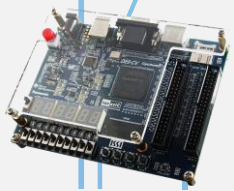
- The designed circuit will have no input and four outputs.
- Think which prime numbers can be represent with four bits.
- In every clock cycle, the output will show next prime number in an ascending order.
- Use the LEDs as the outputs of the circuit. Also, assign the state bits to some LEDs to track which state the circuit in.



3-BIT BACKWARD ODD NUMBER COUNTER

- The designed circuit will have one input and three outputs.
- Use a button as the input, and the LEDs as the outputs of the circuit.
- The initial output value is the biggest odd number that can be represented with 3 bits.
- The circuit will count odd numbers in descending order. When the button is pressed, the LEDs will show the next odd number.

Note: Don't forget the buttons on DE0 board are ON when you don't press.



3-BIT SEQUENCE DISPLAYER

- The designed circuit will have two inputs and three outputs. Use the buttons as the inputs, and the LEDs as the outputs of the circuit.
- Our sequence is 001,110,101,010. One button is the next button, and the other one is the reset button.
- When the next button is pressed, the LEDs will show the next element of the sequence. When the reset button is pressed, the circuit will turn to the initial state.
- If the current output is the last element, the next element will be the first element of the sequence.

Note: Don't forget the buttons on DE0 board are ON when you don't press.