CECS 201
State Machine
Project

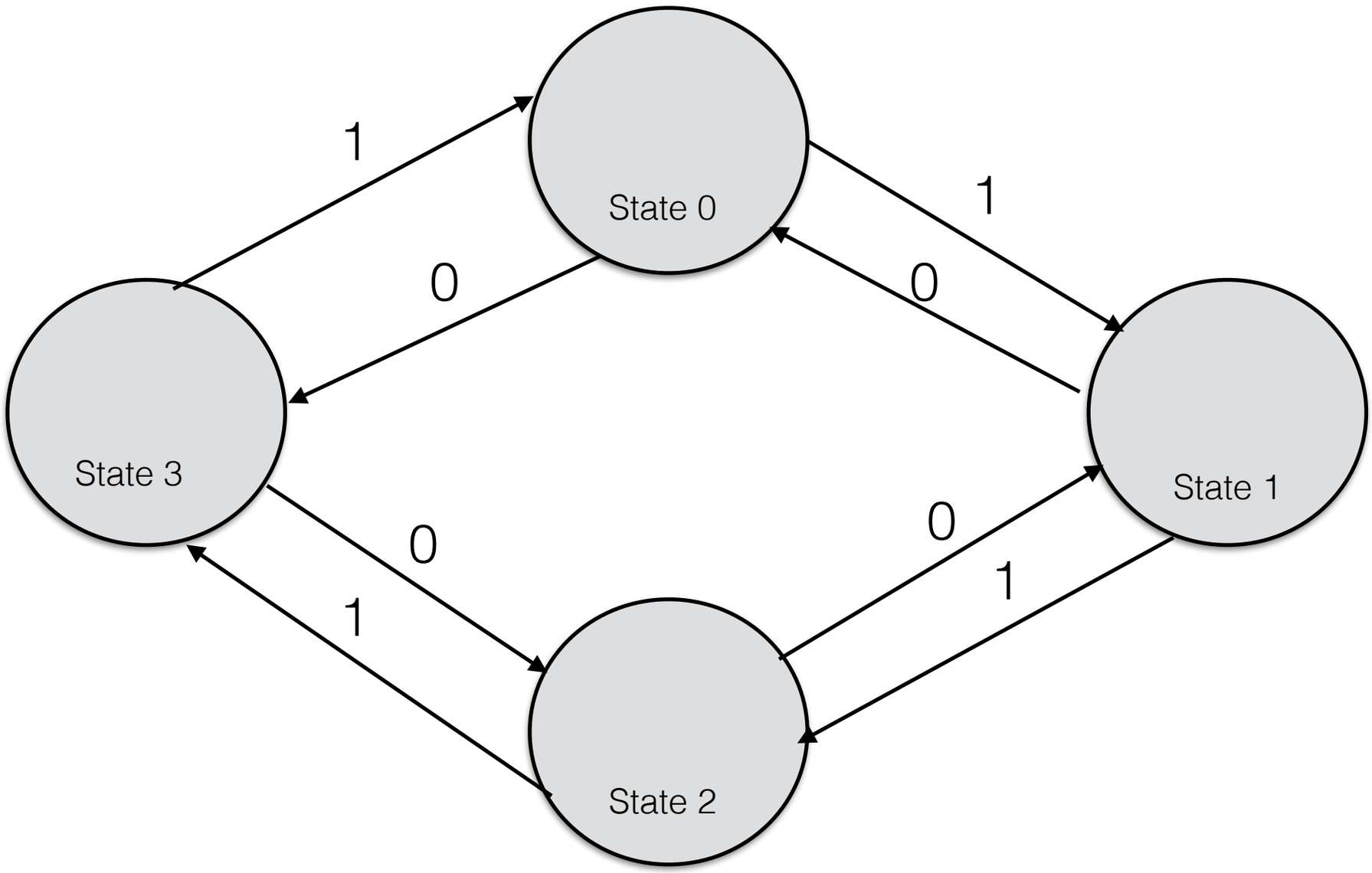
Introduction

- This project will implement the simple state machine presented in class
- A state machine is comprised of inputs, outputs and a state variable (register) to keep track of the steps the state machine traces through
- In our case the output will be the state variable itself
- There will be a single input

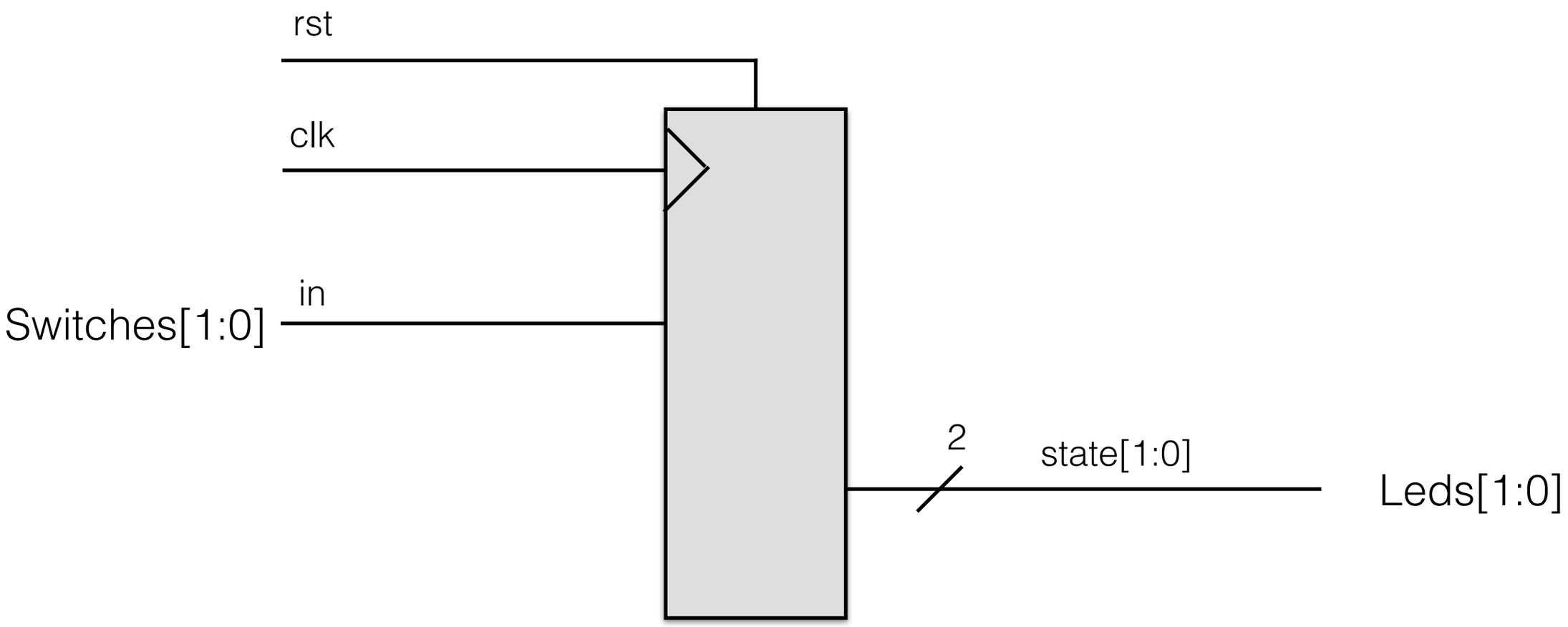
Description

- The behavior of a state machine is captured in a State Transition Diagram
- The diagram identifies the states as circles (or bubbles), transitions between states are indicated by an arc leaving one state and pointing to another, the inputs are the decision makers and are indicated by a value above the arc, and the outputs are identified with the state and are documented within the circle

State Transition Diagram



Design



Coding the Machine

```
`timescale 1ns/1ns

// state machine design

module machine (clk, reset, in, state);
  input         clk, reset, in;
  output [1:0] state;
  reg    [1:0] state, nstate;

always @(posedge clk, posedge reset)
   if (reset) state <= 2'b0;
   else       state <= nstate;

always @(*)
   case(state)
     2'b00: nstate = (in) ? 2'b01: 2'b11;
     2'b01: nstate = (in) ? 2'b10: 2'b00;
     2'b10: nstate = (in) ? 2'b11: 2'b01;
     2'b11: nstate = (in) ? 2'b00: 2'b10;
   endcase

endmodule
```

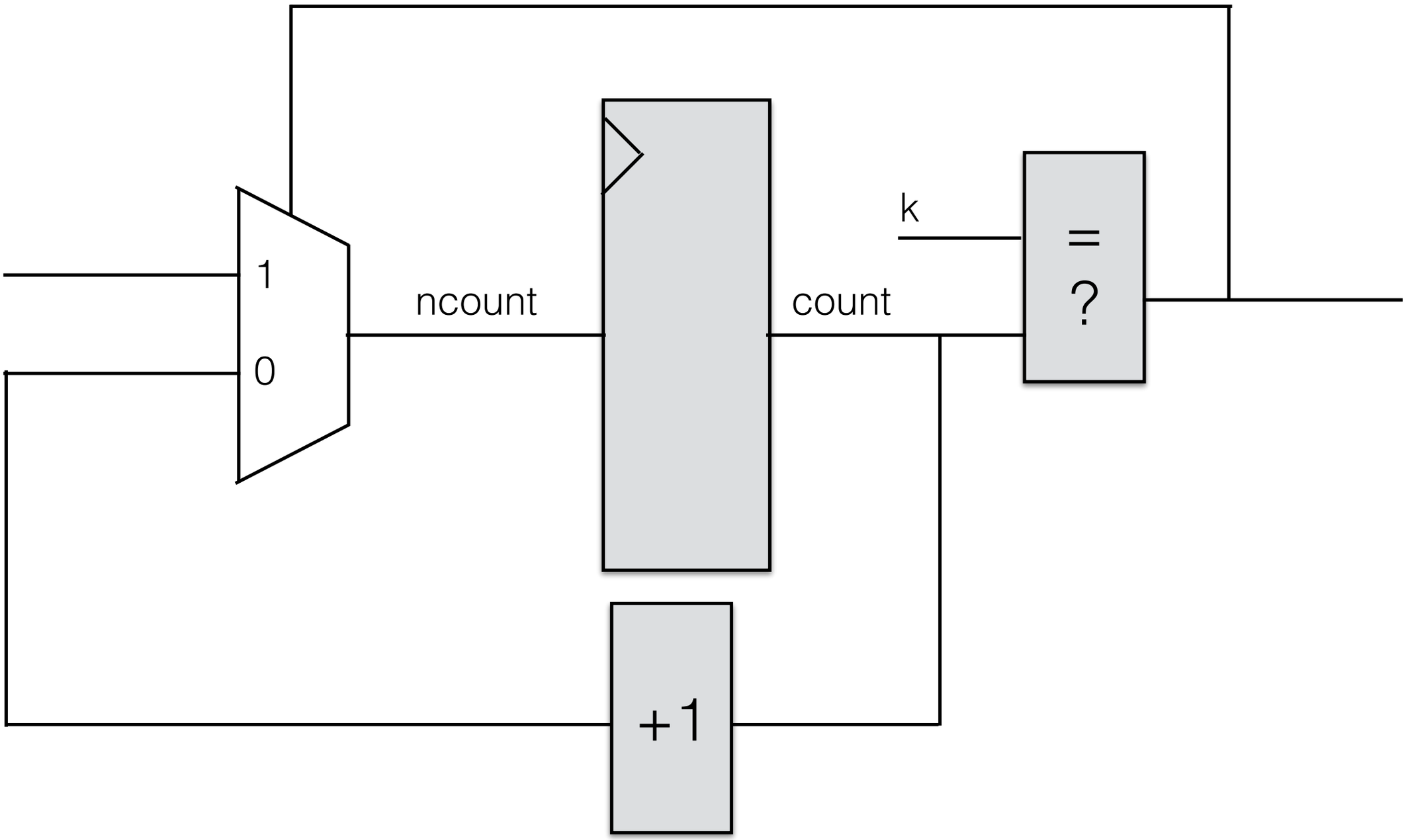
Problem

- The problem that we uncovered with the previous project was that since the board is clocked at 50MHz we can not see the LEDS switching at such a high speed

- We need a way to slow down the behavior of the state machine

- We will design a circuit that will generate a pulse every 1/2 second and then use that pulse to control the movement through the state machine

Pulse Generator



Pulse Generator

- We want to generate a pulse every 1/2 second (500ms).
- To generate a timed pulse we must be aware of the period of our clock (10ns).
- The question becomes: how many clock pulses are there in the interval between pulses?

$$\frac{500\ E\text{-}3}{10\ E\text{-}9} = 50E6 = 50{,}000{,}000 \text{ so count } 0..49{,}999{,}999$$

- So the k in the previous diagram is the value 49,999,999. (remember there are no “,” in Verilog numbers but there are ‘_’ so the value is expressed as “49_999_999”)

Pulse Generator (cont)

- The last value to compute is the number of bits required to hold a value up to 50,000,000.

- Remember:
  - 2^{10} = 1024 (a thousand)
  - 2^{20} = 2^{10} * 2^{10} = 1,048,576 (a million)
  - 2^{26} = 2^{20} * 2^{6} = 64 million

- From the math we see we need 26 bits to hold a value of 50 million

Coding the Pulse Generator

```
`timescale 1ns/1ns

// create a pulse that will slow down the state machine

module pulseit (clk, reset, pulse);
  input          clk, reset;
  output         pulse;
  reg    [25:0] count;
  wire           pulse;


assign pulse = (count == 49_999_999);

always @(posedge clk, posedge reset)
   if (reset) count <= 26'b0; else
   if (pulse) count <= 26'b0; else
              count <= count + 26'b1;



endmodule
```

Modification to the Machine

```
module machine (clk, reset, in, state, pulse);
  input         clk, reset, in, pulse;
  output [1:0] state;
  reg    [1:0] state, nstate;

always @(posedge clk, posedge reset)
   if (reset) state <= 2'b0;
   else       state <= nstate;

always @(*)
   begin
   nstate = state;
   if (pulse)
     case(state)
       2'b00: nstate = (in) ? 2'b01: 2'b11;
       2'b01: nstate = (in) ? 2'b10: 2'b00;
       2'b10: nstate = (in) ? 2'b11: 2'b01;
       2'b11: nstate = (in) ? 2'b00: 2'b10;
       endcase
   end

endmodule
```

Assignment

- The basic code for the project has been given to you. It is your responsibility to integrate these blocks into a top level block that can be used to program the board

- You also need to create a test fixture for the top level design to be able to simulate the design before programming the board

- You also need to create the ucf file for the top level I/O before programming the board

- You will need to write a report and at the time you demonstrate hand to the instructor: 1) report, 2) all the .v files, 3) your .ucf file, and 4) evidence of simulating the design

- Place all files on BeachBoard - Project due May 2