Exercise 2: Structural Verilog *(Implement a ring osc)*

- You should have some familiarity with structural verilog. You used it to perform the AHW's in ECE 352.
- Structural Verilog is simply instantiation and wiring of cells/blocks whose functionality is known.
- The cells/blocks might be built in primitives (AND, OR, NOT, …) or might be more complex cells that you defined using dataflow or behavioral verilog.

Exercise 2: Structural Verilog *(built in primitives)*

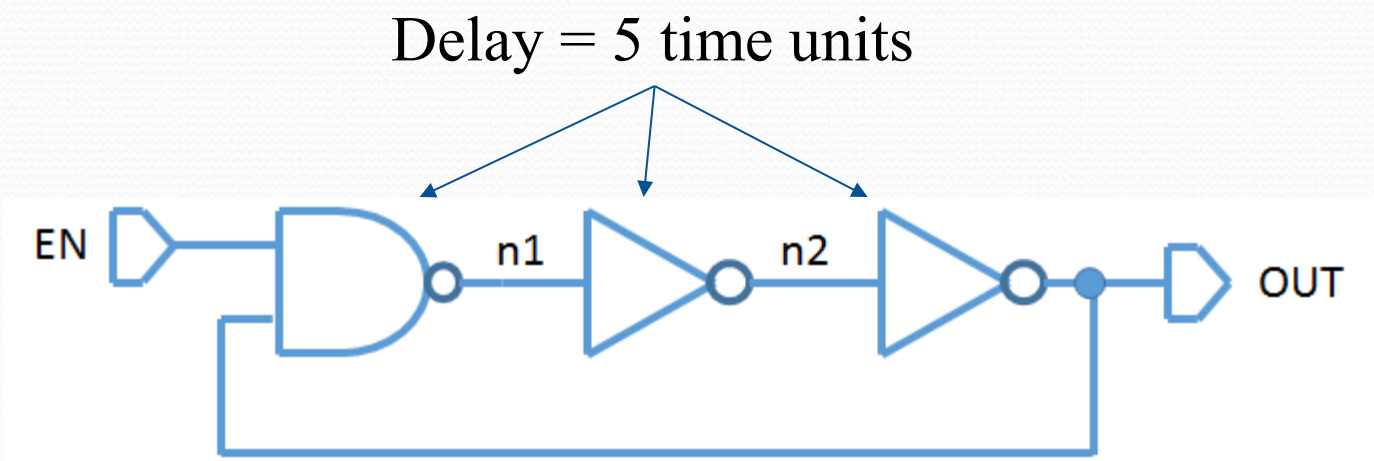
- Built in primitives: **and, or, not, nand, nor, xor, xnr, bufif1, notif1, bufif0, notif0**
- Output port appears before input ports
- Optionally specify: instance name and/or delay

```
and N25 (Z, A, B, C);      // name specified

and #10 (Z, A, B, X),
        (X, C, D, E);      // delay specified, 2 gates

and #10 N30 (Z, A, B); // name and delay specified
```

Exercise 2: Structural Verilog *(implement ring oscillator)*



- What does this circuit do when you raise EN?
- Assume **EN** has been low for a while. Then simulate *(in your head)* what happens when **EN** is raised.
- Create a test bench *(see next slide)* to instantiate and test it.
  - **EN** should be low for 15 time units then raised

Exercise 2: Testbench Basics

- At a minimum a testbench has to do 3 things:
  1. Instantiate the DUT
  2. Hook up the DUT
  3. Apply stimulus to DUT inputs

```
module basic_test_bench();                  // a testbench is self-contained (has no inputs/outputs)

 reg stim1, stim2;                          // stiumulus to DUT declared as type reg or logic

 // Instantiate DUT = Device Under Test //
 DUT_module  iNAME(.clk(clk), .rst_n(rst_n), .in1(stim1), .in2(stim2), .out1());

 initial begin                              // stimulus typically provided in an initial block
   stim1 = 0;
   stim2 = 1;
   #5;                                      // wait 5 time units then change stimulus
   stim1 = 1;
   #5 $stop();                              // wait 5 more time units then stop the simulation
 end

endmodule
```

module name of DUT
Does not have to match file name

Instance name can be anything, but is required

Input of DUT named in1 is hooked to a signal called stim1 at this level of hierarchy

Outputs of DUT can be hooked to nothing, however, more likely you would hook it to a signal so you can monitor it

Exercise 2: Testbench Basics

- Create the DUT using structural verilog (**ring_osc.sv**)
- Create a test bench (**ring_osc_tb.sv**) to instantiate and test it.
  - **EN** should be low for 15 time units then raised
- Capture the waveforms (**waves.jpg**) proving it works.
- You are allowed to use Copilot / LLM for this exercise, but you must still understand what you built.
- There is Canvas quiz associated with this exercise. Complete it.