Verilog in Cadence
What is Verilog?

• Hardware Description Language for describing electronic circuits and systems
Why??

• **Verification** through simulation, timing analysis, test analysis, and logic synthesis

• Will/should my design work?

• Simpler to test/simulate multiple cases
Implementation

• **Goal:** Verilog match Schematic

• However, in Cadence:
  – Checks Verilog (behavioral) against symbol (inputs/outputs)
  – Does NOT check Verilog against actual schematic design
Therefore...

• It is the designer’s responsibility to implement Verilog modules exactly how designed in schematic

• Best way to make sure of this consistency during implementation??
How to Implement Modules

• Implement Verilog modules at the lowest level possible
  – Even down to transistor level!!
  – Avoid using higher level modules (not, and, nor, etc.)

• Can call other Verilog modules from current module
  – “Library” of modules
  – Example: Calling inverter module in Ring Oscillator
NOR2 Gate Example Module

- Implementing at lowest level possible

```
module nor2 (out, A, B);
input A;
input B;
output out;
wire node0;
assign out = node0;
supply1 vdd;
supply0 vss;
nmos n0 (node0,vss,A);
nmos n1 (node0,vss,B);
pmos p0 (node0,node1,B);
pmos p1 (node1,vdd,A);
endmodule
```
Ring Oscillator Example

• Calling one module inside another

```verilog
module inv(input in, output out);
    wire node0;
    assign out = node0;

    supply1 vdd;
    supply0 vss;

    pmos p0 (node0,vdd,in);
    nmos n0 (node0,vss,in);
endmodule

module ring_osc (out, out1, out2, out3, out4, control);

    input control;
    output out;
    output out1;
    output out2;
    output out3;
    output out4;

    and a1 (start, control, out);
    inv n1 (out1,start);
    inv n2 (out2, out1);
    inv n3 (out3, out2);
    inv n4 (out4, out3);
    inv n5 (out, out4);
endmodule
```