Verilog - Fundamental Concepts

Levels of abstraction

- Verilog is able to model a design at different levels of abstraction
- High levels express little detail, low levels express much
- Boundaries between levels often not well defined
- Other equally valid and slightly different descriptions exist

Verilog - Fundamental Concepts

Levels of abstraction (Behavioral)

- Could be C, Perl, or Verilog
- C or Perl considered to be High-level behavioral
- Models behavior without respect to any implementation or structure
- No timing or sequence of operations is implied
- No signals or devices are defined
- Typically non-synthesible
- For example: addout = operanda + operandb
- Not specified: bus width, timing, clock
- Often used in testbenches

Verilog - Fundamental Concepts

Levels of abstraction (RTL)

- Uses Verilog at RTL level (Register Transfer Level)
- Timing accurate to clock period level (thus "register" moniker)
- No explicit gate descriptions (important)
- Typical example is a state machine

```
//RTL level 2:1 mux
module mux2to1 (input  a,b,  //data inputs
                input  sel,  //select line
                output z     //output
                );
always_comb begin
  case (sel)
    1'b0    : z = a;
    1'b1    : z = b;
    default : z = 1'bx;
  endcase
end
endmodule
```

Verilog - Fundamental Concepts

Levels of abstraction (Gate)

- Models behavior at the gate level
- Gates, FFs, other cells exist in the design
- Description is a Verilog netlist (ASCII schematic)
- Can be used to model more complex real gates, AOI, OAI
- Seldom used for design - level of abstraction is too low

```
//Gate level 2:1 mux
module mux2to1 (input  a,b,  //data inputs
                input  sel,  //select line
                output z     //output
                );
  wire not_sel;
  not not0 (sel_not, sel);

  wire and0_out, and1_out;
  or or0(          z ,  and0_out,   and1_out);
  and and0( and0_out ,         a,        sel);
  and and1( and1_out ,         b,    not_sel);
endmodule
```

Verilog - Fundamental Concepts

Levels of abstraction (Switch)

- Primitive transistor (as a switch) behavior
- Propagation delay, rise and fall time modeling possible
- Can be used to model CMOS circuits (not gates)
- We won't use in our class

Verilog - Fundamental Concepts

Advantages of multiple levels of abstraction

- Multiple designers may be at different levels of implementation
- Verilog can work with a design using all levels simultaneously
- IP may be offered at different levels of abstraction than you are at