- System specifications are behavioral in nature.

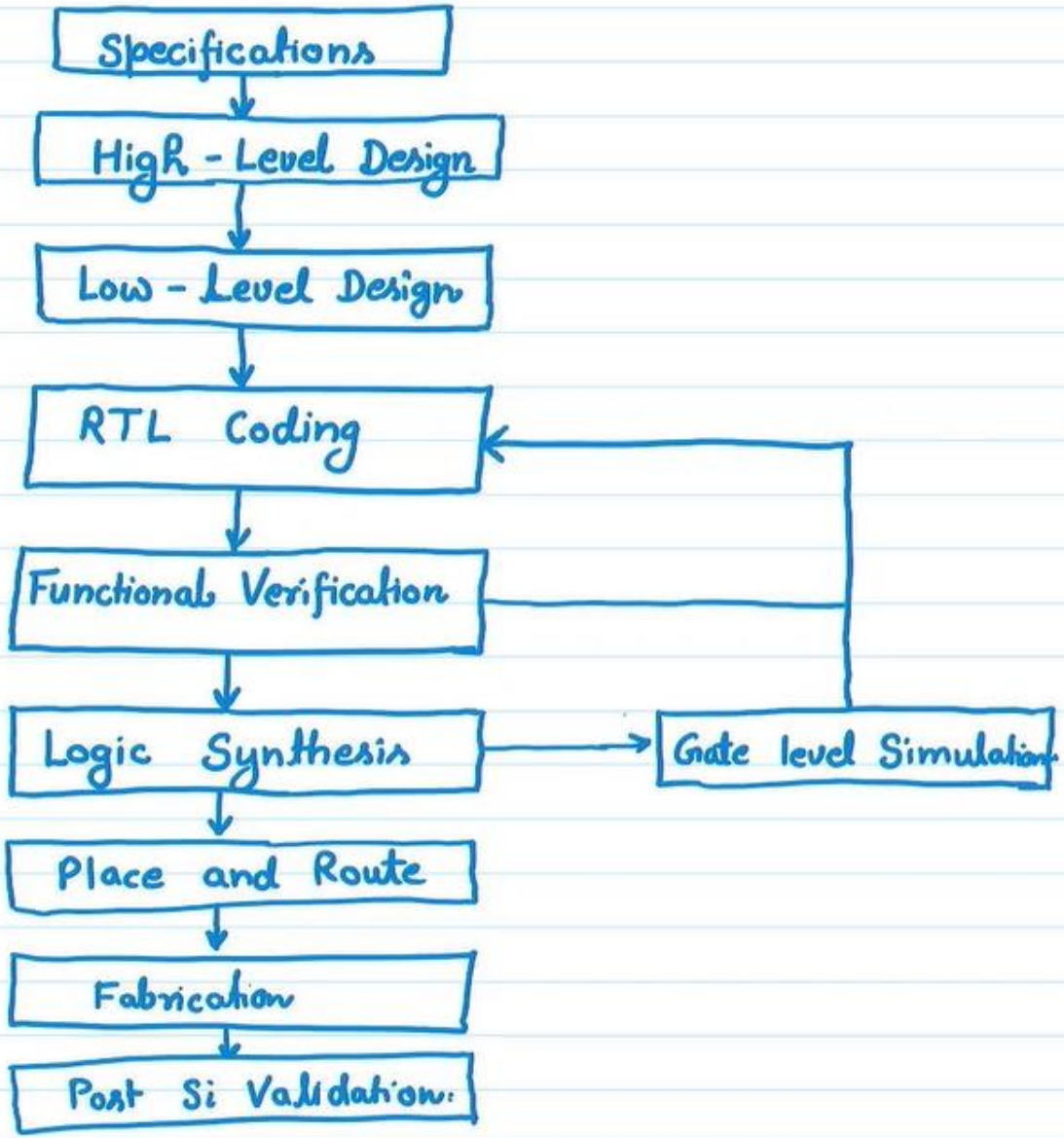
- Handling of large complex designs is possible through CAD softwares.

- Impractical to use SPICE for simulating large digital circuits
  - Time complexity.

- Manual optimization of design Boolean expressions are difficult.

Verilog	VHDL
- Has good acceptance in ASIC.	- Relatively weak in low-level design.
- Good for low-level designs - RTL and below	- Superior in system level design.
- Fast simulations	- Slower simulations.
- Relatively simple (similar to C lang.)	- Relatively complex.
- Popular in N. America, Asia and Japan.	- Popular in Europe.

Top-down Design Approach :



Definition of Module in Verilog :

module module_name (port_list)

port declaration

`include directives

variable declarations
assignments
lower-level module instantiation.
initial and always blocks.
tasks and functions.

} body

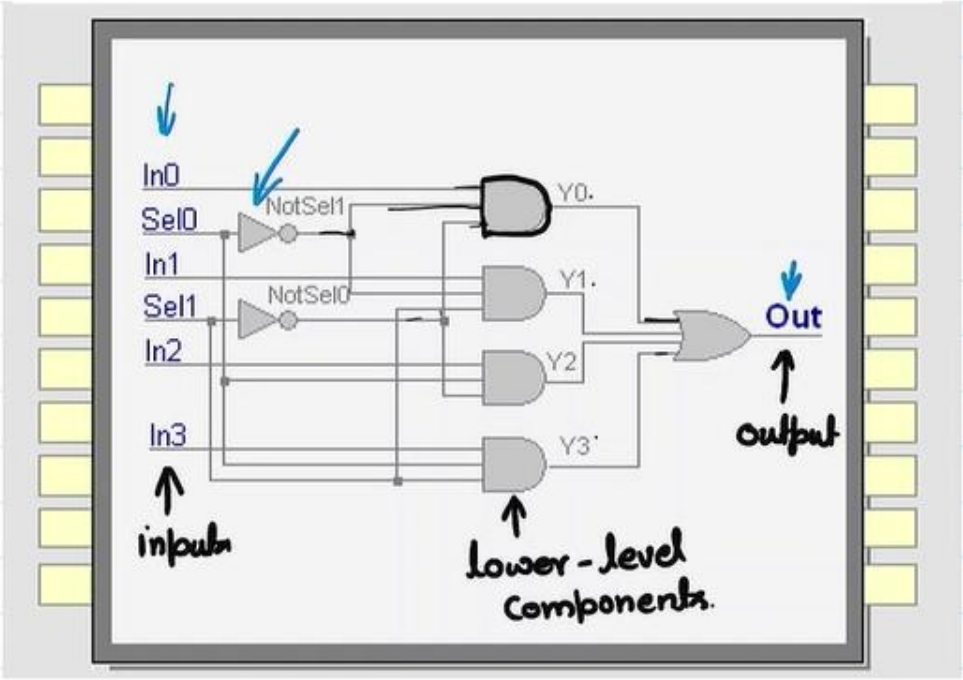
endmodule

Comments in Verilog

- One line comment
  // ....

- Block comments
  /* .... */

Coding Style : Structural

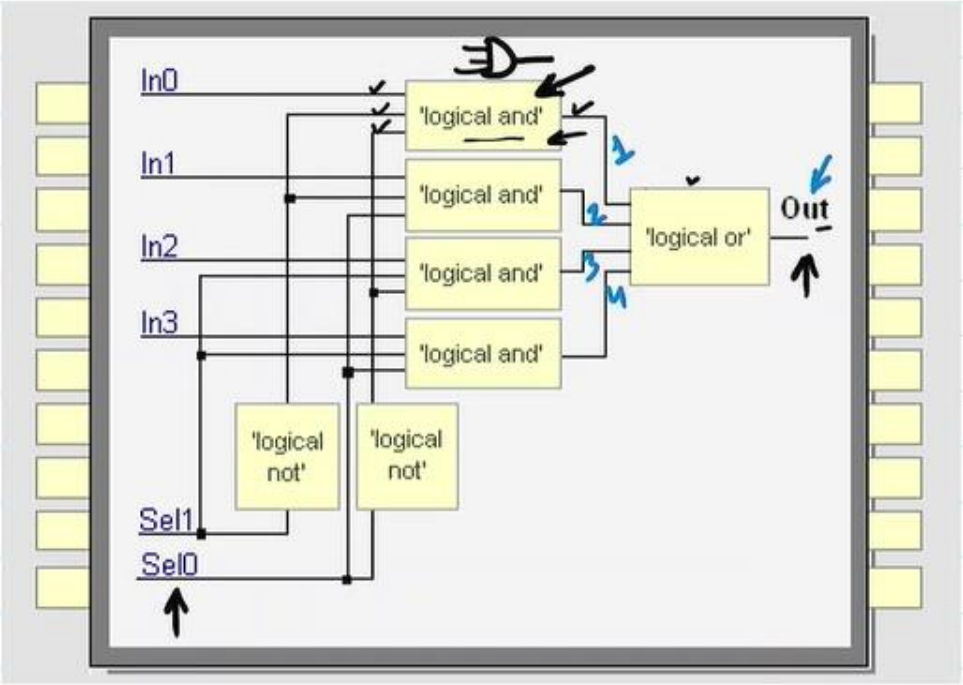



```
module mux_4_to_1 (Out,In0,In1,In2,In3,Sel1,Sel0);
output Out;
input In0, In1, In2, In3, Sel0, Sel1;

wire NotSel0, NotSel1;
wire Y0, Y1, Y2, Y3;

not (NotSel0, Sel0);
not (NotSel1, Sel1);
and (Y0, In0, NotSel1, NotSel0);
and (Y1, In1, NotSel1, Sel0);
and (Y2, In2, Sel1, NotSel0);
and (Y3, In3, Sel1, Sel0);
or (Out, Y0, Y1, Y2, Y3);

endmodule
```

- Circuit is specified in terms of lower-level components (NAND , NOR etc.). These are connected with internal signals.
- Synthesis is easy.

NPTEL

Coding Style : Data Flow

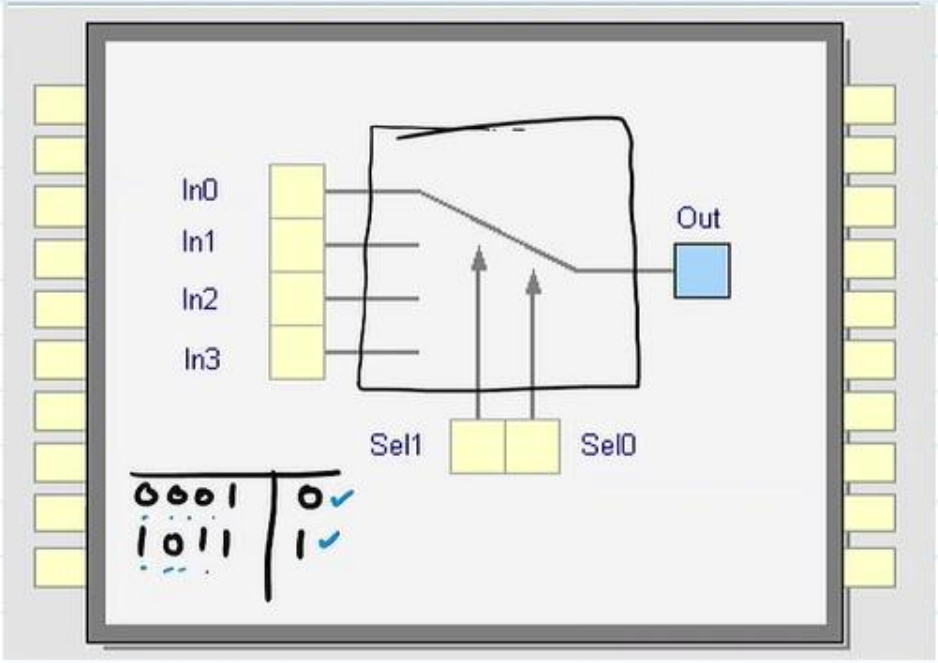



```
module mux_4_to_1 (Out, In0, In1, In2, In3, Sel1, Sel0);

output Out;
input In0, In1, In2, In3, Sel0, Sel1;

assign Out = (~Sel1 & ~Sel0 & In0) | (~Sel1 & Sel0 & In1)
           | (Sel1 & ~Sel0 & In2) | (Sel1 & Sel0 & In3);

endmodule
```

- Similar to logical equations.
- Expressions are assigned to output.
- Synthesis is okay.

NPTEL

Coding Style : Behavioral




```
module mux_4_to_1 (Out,In0,In1,In2,In3,Sel1,Sel0);
output Out;
input In0, In1, In2, In3, Sel0, Sel1;
reg Out;

always @(Sel1 or Sel0 or In0 or In1 or In2 or In3)
  begin
    case ({Sel1, Sel0})
    2'b00 : Out = In0;
    2'b01 : Out = In1;
    2'b10 : Out = In2;
    2'b11 : Out = In3;
    default : Out = 1'bx;
    endcase
  end

endmodule
```

- Specify the circuit in terms of its expected behavior.
- close to natural language description.
- Most difficult to synthesize.

NPTEL