Lecture - 6

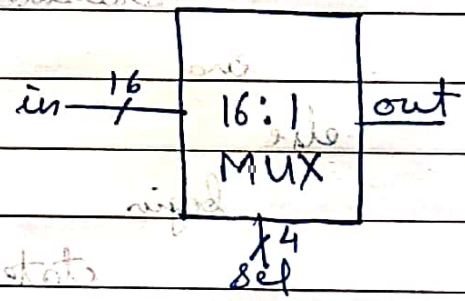
☆ Verilog Modeling

- **Behavioral** → How a system is working?
- **Structural** → How a sys is constructed?

☆ Behavioral Modeling

* Example: 16:1 Multiplexer

```
module mux16to1 (in, sel, out);
    input [15:0] in;
    input [3:0] sel;
    output out;
    wire [3:0] t;
    assign out = in[sel]; // Equivalent to accessing
endmodule                 // element of an array
```



Note: In above example, we have only specified the behavior of a 16:1 multiplexer.

* Write similare codes for the following:

1) 8:1 MUX, 4:1 MUX
2) 4:2 Encoder, 8:3 Encoder
3) 2:4 Decoder, 3:8 Decoder

Scanned with CamScanner

☆ If-Else statement

- if-else should be used only inside "@always@ block

- Syntax:

1) if (condition)
   begin
      statements
   end

2) if (condition)
   begin
      statements
   end
   else
   begin
      statements
   end

3) if (condition)
   begin
      statements
   end
   else if (condition)
   begin
      statements
   end
   else
   begin
      statements
   end

Scanned with CamScanner

☆ always @

* syntax:

```
always @(sensitivity_list)
begin
        statements;
end
```

* Non - Blocking Assignment (<=)

```
always @(a,b,c,d)
begin
    b <= a;
    c <= b;      } Concurrent execution
    d <= c;
end
```

* Blocking Assignment (=)

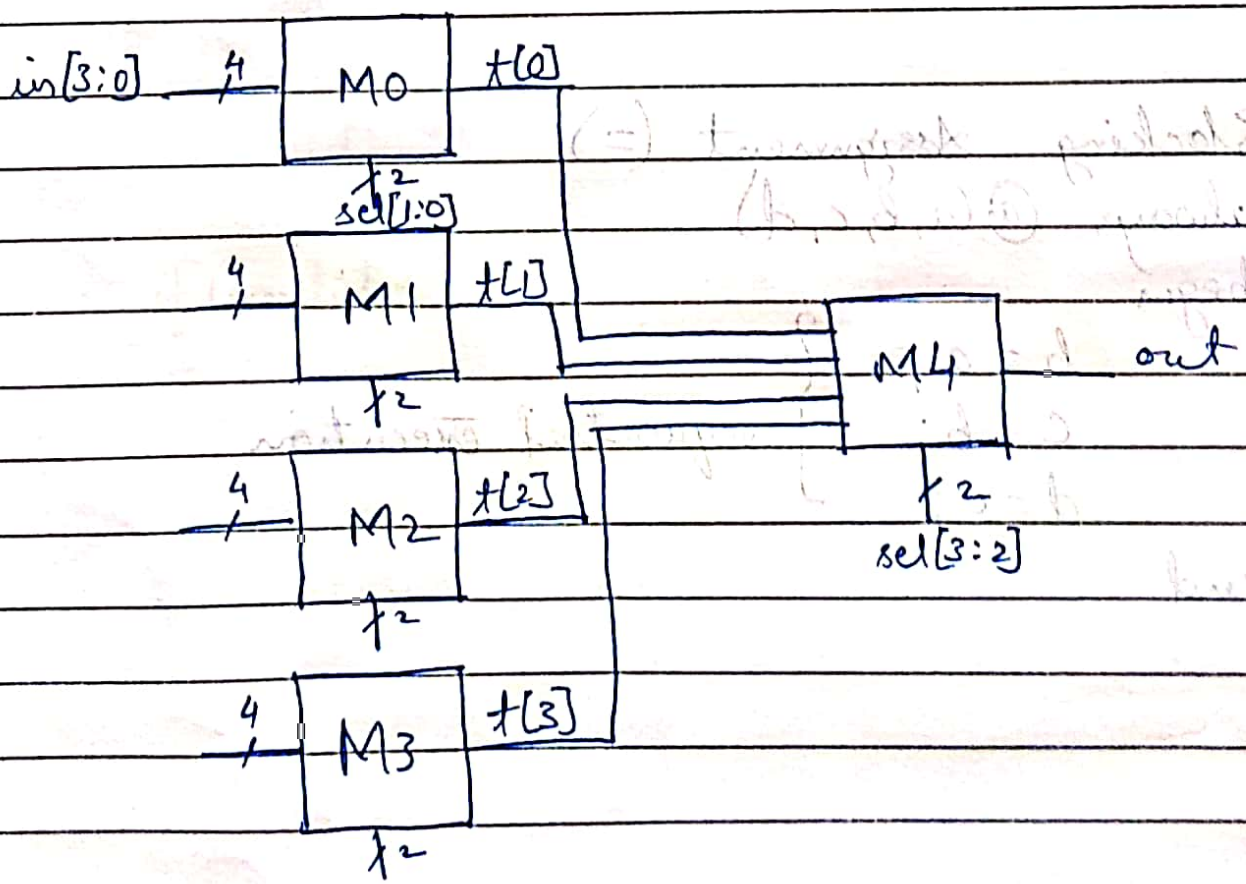
```
always @(a,b,c,d)
begin
    b = a;
    c = b;       } Sequential execution
    d = c;
end
```

Scanned with CamScanner

☆ Structural Modeling

*Example: 16:1 MUX using only 4:1 MUX

```
module mux16to1 (in, sel, out);
    input [15:0] in;
    input [3:0] sel;
    output out;
    wire [3:0] t;
    mux4to1 M0 (in[3:0], sel[1:0], t[0]);
    mux4to1 M1 (in[7:4], sel[1:0], t[1]);
    mux4to1 M2 (in[11:8], sel[1:0], t[2]);
    mux4to1 M3 (in[15:12], sel[1:0], t[3]);
    mux4to1 M4 (t, sel[3:2], out);
endmodule
```



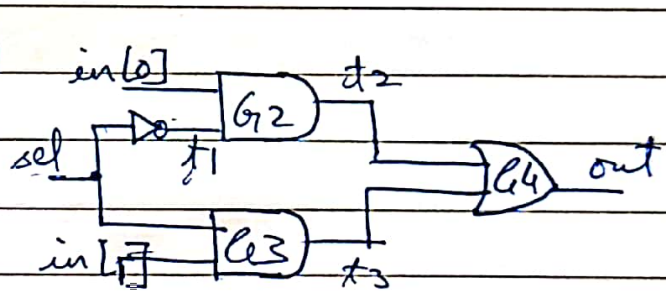
Scanned with CamScanner

* 4:1 MUX using behavioural style

```
module mux4to1 (in, sel, out);
  input [3:1] in;
  input [1:0] sel;
  output out;
  assign out = in[sel];
endmodule
```

* 2:1 MUX using structural modeling

```
module mux2to1 (in, sel, out)
  input [1:0] in;
  input sel;
  output out;
  wire t1, t2, t3;
  NOT G1 (t1, sel)
  AND G2 (t2, in[0], t1);
  AND G3 (t3, in[1], sel);
  OR G4 (out, t2, t3);
endmodule
```



Scanned with CamScanner