Lecture - 8

✱ Verilog Description Styles

Two different styles of description:

1) Dataflow
   - continuous assignment

2) Behavioral
   - procedural statements
     - → blocking
     - → non-blocking

✱ Dataflow Style

- "assign" is used
- assign net_type_variable = net_type_variable;
  OR
  assign net_type_variable = register_type_variable;
- The assignment is continuously active so it is used to model combinational circuits.
- A verilog code can contain any number of "assign" statements

- module
  - signal declaration
  - assign statements
  - other statements (will be discussed later)

endmodule

Typical structure of a Verilog code

* Example:

```
module generate_MUX (data, select, out);
  input [15:0] data;
  input [3:0] select;
  output out;
  assign out = data [select];
endmodule
```

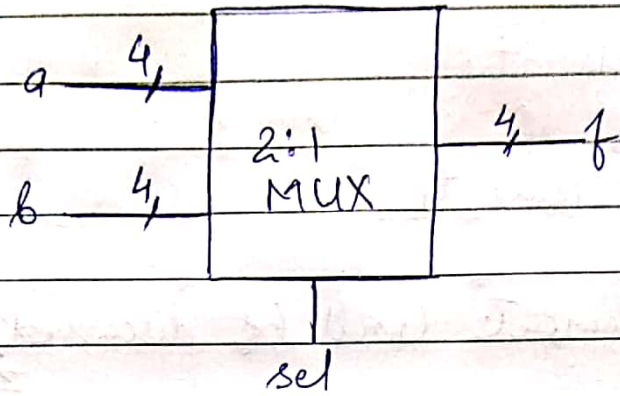
Q] What will be the corresponding hardware synthesized?

→ • Non-constant index in an expression on R.H.S. generates a MUX.
• If a constant index is present (Eg: assign out = data[2];) just a wire will be generated.

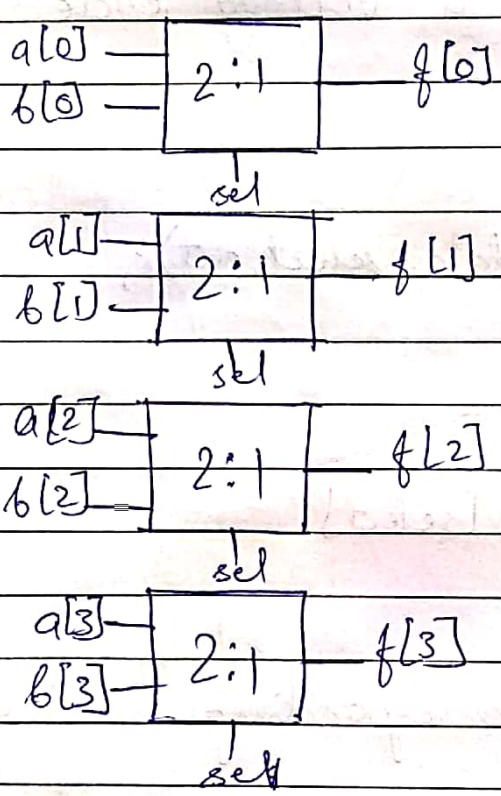
* Example:

```
module generate_set_of_MUX(a, b, f, sel);
  input [0:3] a, b;
  input sel;
  output [0:3] f;
  assign f = sel ? a : b;
endmodule
```

→ {Conditional operator generates a MUX

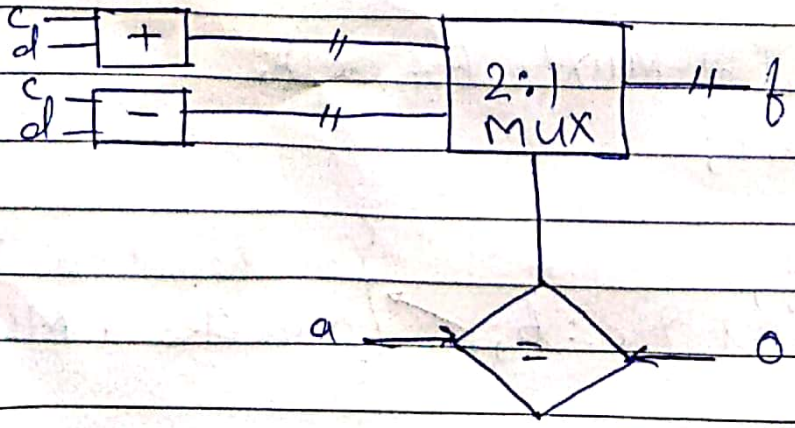


OR



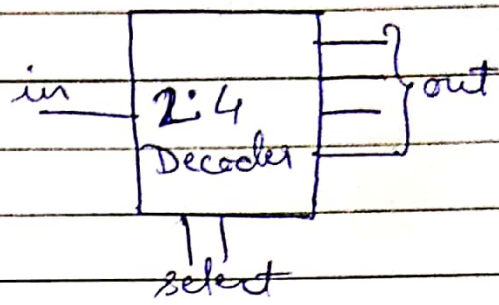
Note: A conditional assignment always generates a 2:1 MUX

- assign f = (a == 0) ? (c+d) : (c-d);



*Example:

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



Note: • Non-constant index in an expression on L.H.S. generates a decoder.
• Constant index in the expression on the LHS will simply generate a wire connection.

*Example:

```
module level_sensitive_latch (D, Q, En);
    input D, En;
    output Q;
    assign Q = en ? D : Q;
endmodule
```

En	D	Qn
0	X	Q_{n-1}
1	0	0
1	1	1

Note: A latch is generated if previous value of a variable is assigned to itself.