Digital Hardware Design

Verilog in 10 Minutes – Verilog Coding Styles

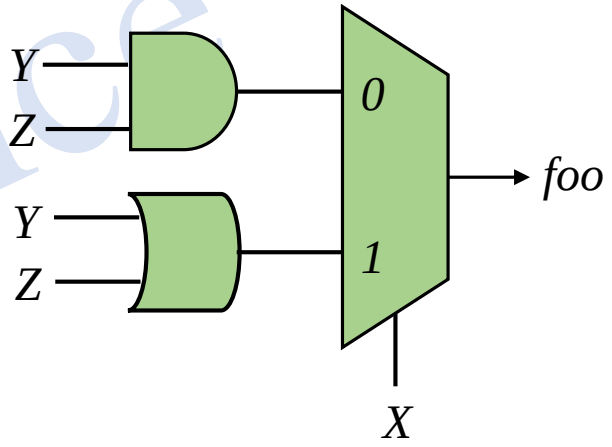
Video Lecture Link

Digital Hardware Design

Verilog Combinational Logic (CL)

How would you describe the behavior of this function in words ?

If X = 1, then foo = Y or Z,
else foo = Y and Z



How would you describe the behavior of this function in Verilog Code ?

```
always @ (X or Y or Z)
 if(X) foo = Y | Z;
 else foo = Y & Z;
```

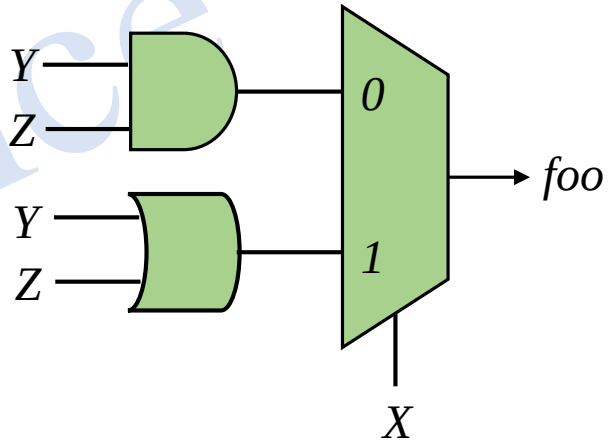
foo can change when any of X, Y or Z changes
So, code must be rerun whenever any of these changes

Digital Hardware Design

Alternative Coding Style for CL

Verilog has a short hand way to capture Combinational Logic
Called “Continuous Assignment”

```
assign foo = X ? Y | Z : Y & Z;
```



LHS Re-evaluated whenever anything in RSH changes

```
assign f = a ? b : c
```

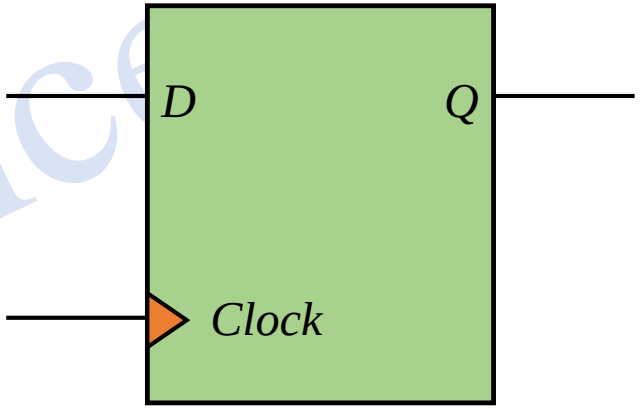
Is same as
if(a) f = b;
else f = c;

Digital Hardware Design

Flip-Flop

Flip Flop Behavior -

For every positive edge of the clock Q changes to become equal to D



Flip Flop Behavior as Verilog Code -

```
always @ (posedge clock)
  Q <= D;
```

always @ ()
- Triggers execution of the following code block
- () is called 'Sensitivity List'
  -- Describe when execution triggered

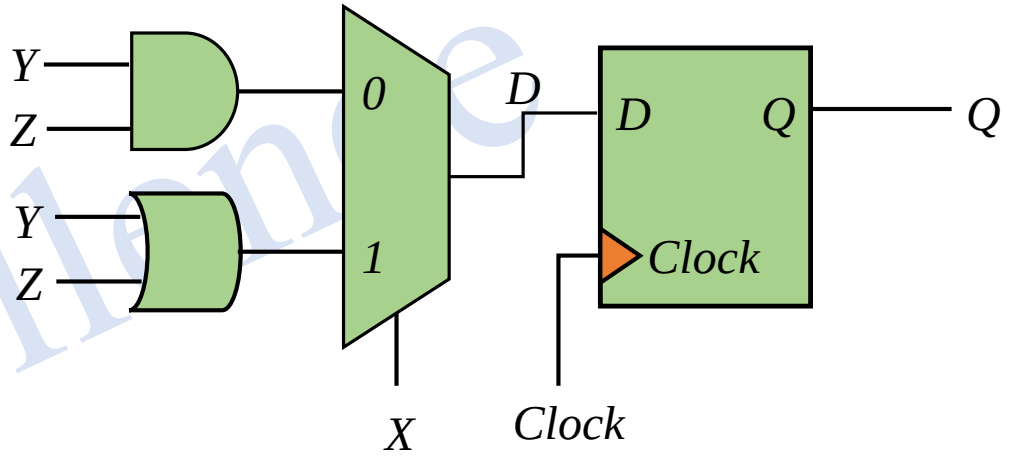
Digital Hardware Design

Verilog Module for Flip Flop

```
module flipflop (Clock, X, Y, Z, Q);

    input Clock, X, Y, Z;
    output Q;
    reg Q;
    wire D;
assign D = X ? Y | Z : Y & Z;

always @ (posedge Clock)
   begin
       Q <= D;
    end
endmodule
```



Digital Hardware Design

Verilog Module for Flip Flop

```
module flipflop (Clock, X, Y, Z, Q);

    input Clock, X, Y, Z;
    output Q;
    reg Q;
    wire D;
assign D = X ? Y | Z : Y & Z;

always @ (posedge Clock)
   begin
       Q <= D;
    end
endmodule
```

Module Name

Connected Ports

Port Declarations

Local Variables Declarations

Code Segments

Digital Hardware Design

RTL Coding Styles

Three Coding Styles -

- always @(???edge Clock) -> Flip Flops and Input Logic
- always @(*) -> Combinational Logic (CL) specified by its behavior
- assign a = . . . -> Combinational Logic (CL) specified as structure

Best Free VLSI Content

1. Verilog HDL Crash Course – Link
2. Static Timing Analysis (STA) – Theory Concepts – Link
3. Static Timing Analysis (STA) – Practice/Interview Questions – Link
4. Low Power VLSI Design – Theory Concepts – Link
5. Low Power VLSI Design (LPVLSI) – Practice/Interview Questions – Link
6. Digital ASIC Design Verilog Projects – Link

Please Like, Comment, Share & Subscribe My Channel in Order to Reach Out the Content to a Larger Audience.

Thanks !!