Lecture 14: Finite State Machines and Parameterized Modules

Matthew Shuman
November 4th, 2011

1 Combinational Logic in Verilog Review — Not in text

```verilog
module MysteryBlock(input [15:0] A, B, input [2:0] F,
                     output reg [15:0] Y);

wire [15:0] S, Bout;

assign S = A + Bout + F[2];

always @ (*)
    case (F[1:0])
        2'b00: Y <= A & Bout;
        2'b01: Y <= A | Bout;
        2'b10: Y <= S;
        2'b11: Y <= S[15];
    endcase
endmodule
```

Figure 1: Initial example to begin lecture.

Here are the key details to learn from this example:

1. Line 2: The 'reg' keyword has to be added to the output [15:0] Y declaration, because Y is assigned in an always block. See Example 4.18 in the text for more information about this.

2. Line 6: The '?' describes a 2:1 multiplexer. See Example 4.5 in the text for more information about this.

3. Line 9: The 'always @ (*)' indicates what the always block will listen to in order to determine if it should process inputs into output. This is called a sensitivity or stimulus list.

4. Line 9 - 15: This always block describes a 4:1 multiplexer that switches 16 bit wide inputs.

5. Line 6 & Line 11: Note that the assign statement uses an '=' on line 6, but Y is also assigned using an '|=' on line 11. Look at Example 4.23 in the text for more information about this.
2 Finite State Machines in Verilog — Section 4.6 in text

```verilog
module TrafficLight(
    input Clock,
    input Reset,
    output reg RedLight,
    output reg YellowLight,
    output reg GreenLight
);

reg [1:0] State, NextState;

//State Assignment
parameter Red = 2'b00;
parameter Yellow = 2'b01;
parameter Green = 2'b11;

//State Register
always @(posedge Clock, negedge Reset)
    if(!Reset) State <= Red;
    else State <= NextState;

//Next State Logic
always @(*)
    case(state)
        Red:     NextState = Green;
        Green:   NextState = Yellow;
        Yellow:  NextState = Red;
        default: NextState = Red;
    endcase

//Output Logic
assign RedLight = (State == Red);
assign YellowLight = (State == Yellow);
assign GreenLight = (State == Green);
endmodule
```

Figure 2: Traffic Light Controller example done in Verilog.

Here are the key details to learn from this example:

1. Line 11: The state assignment section allows for the remainder of the description to be much more intuitive.
2. Line 16 - 19: This section describes a register that is made from D flip-flops.

3. Line 22 - 28: This section describes a combinational logic block to decide what the next state should be.

3 Parameterization of Modules — Section 4.7 in the text

Look at Example 4.34 of the text.