Digital IC Design

CHENCHETY ABHISHEK

EE20BTECH11012

Assignment-1

Question-4

- Implement f= x0h0 + x1h1 + x2h2 +x3h3 + x4h4 + x5h5 + x6h6 + x7h7 + x8h8 + x9h9. Compute f assuming you have **TWO multipliers** and **ONE adder**.

Verilog Code for Module

```
module Q3
(out,clk,x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7,h
_8,h_9);

  input
[3:0]x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8,h
_9;

  input clk;

  output reg [10:0] out;

  reg [2:0] counter = 3'b000;

  // 4 muxes are used
  reg [3:0] mux_1, mux_2, mux_3, mux_4;

  // 2 multipliers
  reg [7:0] mul_1, mul_2;

  reg [10:0] sum, temp;

  always
@(x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8,h_9)
begin
  counter = 3'b000;
    // out <= 11'b00000000000;
   sum  <= 11'b00000000000;
   temp <= 11'b00000000000;
  end

  always @(posedge clk) begin
  if (counter < 6) counter <= counter + 1;
  else counter <= 3'b000;
  end
```

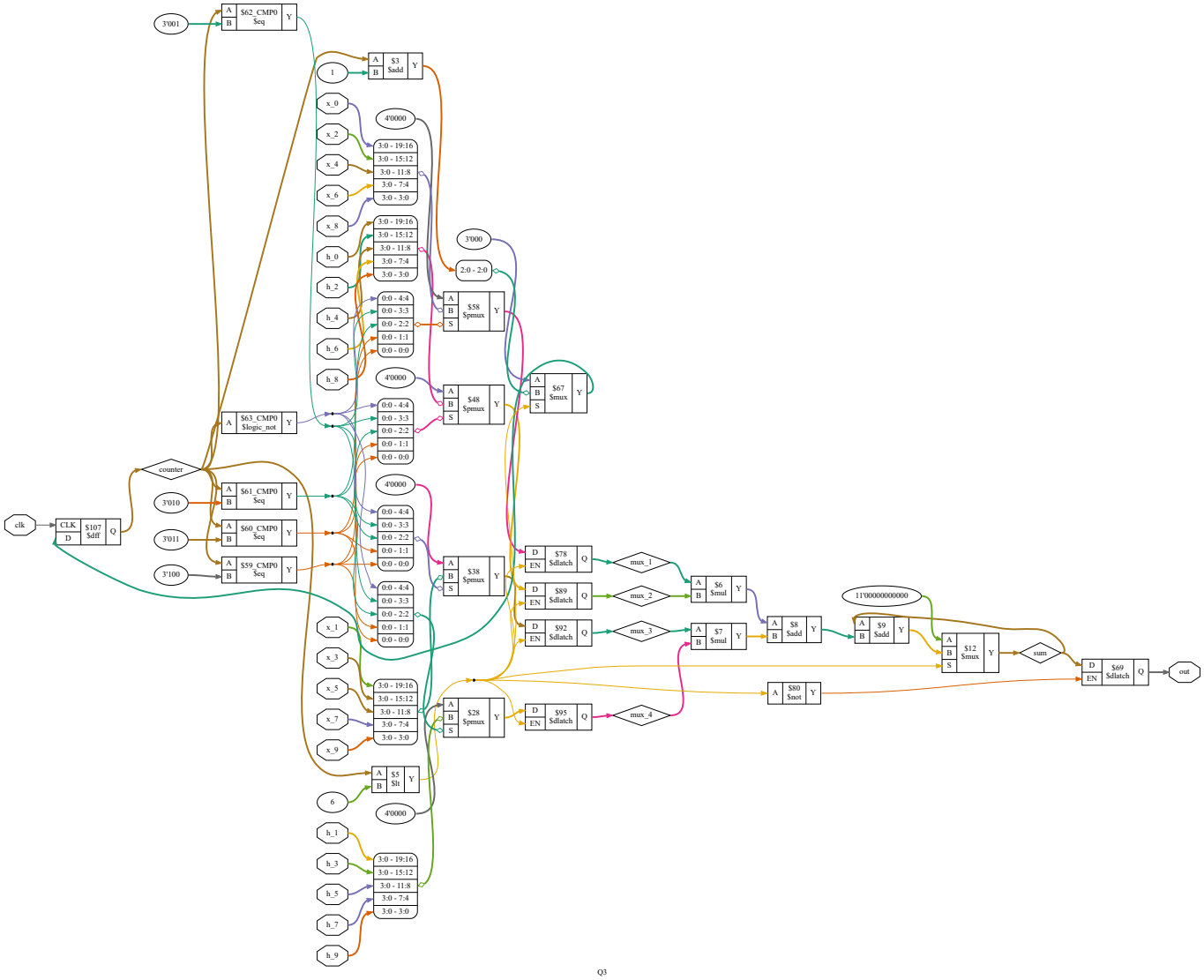
```
  always @(counter) begin
  if (counter < 6) begin
  case (counter)
  3'b000: begin mux_1 <= x_0; mux_2 <= h_0; mux_3 <= x_1; mux_4 <= h_1; end
  3'b001: begin mux_1 <= x_2;mux_2 <= h_2;mux_3 <= x_3;mux_4 <= h_3;end
  3'b010: begin mux_1 <= x_4;mux_2 <= h_4;mux_3 <= x_5;mux_4 <= h_5;end
  3'b011: begin mux_1 <= x_6;mux_2 <= h_6;mux_3 <= x_7;mux_4 <= h_7;end
  3'b100: begin mux_1 <= x_8;mux_2 <= h_8;mux_3 <= x_9;mux_4 <= h_9;end
  default: begin mux_1 <= 0;mux_2 <= 0;mux_3 <= 0;mux_4 <= 0;end
  endcase
    mul_1 = mux_1 * mux_2;
    mul_2 = mux_3 * mux_4;
    temp  = mul_1 + mul_2;
    sum   = sum + temp;
  end
  else begin out = sum;sum = 0;end
  end

endmodule
```

Testbench Code

```
`timescale 1ns/1ns
`include "Q3.v"

module Q3_tb;

// Declaring Port parameters for testbench
reg
[3:0]x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8,h
_9;
reg clk=1'b0;
wire [10:0]Output;


// Clock pulse generation
always #1 clk = ~clk;


Q3
uut(Output,clk,x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,h_0,h_1,h_2,h_3,h_4,h_5,h_6
,h_7,h_8,h_9);

// Declaring variables for testbench
integer testcases,expected_out;
// Monitoring the input values & Output
initial begin
    $monitor($time,"x_0=%d x_1=%d x_2=%d x_3=%d x_4=%d x_5=%d x_6=%d x_7=%d x_8=%d
x_9=%d h_0=%d h_1=%d h_2=%d h_3=%d h_4=%d h_5=%d h_6=%d h_7=%d h_8=%d h_9=%d
Output=%d
```

```
expected=%d",x_0,x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,h_0,h_1,h_2,h_3,h_4,h_5,h_6,h
_7,h_8,h_9,Output,expected_out);

    $dumpfile("Q3.vcd");
    $dumpvars;

   #1100 $finish;
end

// Running testcases
initial begin

        expected_out =  x_0*h_0 + x_1*h_1 + x_2*h_2 +x_3*h_3 + x_4*h_4 + x_5*h_5 +
x_6*h_6 + x_7*h_7 + x_8*h_8 + x_9*h_9;

        for (testcases=0; testcases<10; testcases++) begin
            #100
x_0=$random;x_1=$random;x_2=$random;x_3=$random;x_4=$random;x_5=$random;x_6=$rando
m;x_7=$random;x_8=$random;x_9=$random;h_0=$random;h_1=$random;h_2=$random;h_3=$ran
dom;h_4=$random;h_5=$random;h_6=$random;h_7=$random;h_8=$random;h_9=$random;

            expected_out =  x_0*h_0 + x_1*h_1 + x_2*h_2 +x_3*h_3 + x_4*h_4 +
x_5*h_5 + x_6*h_6 + x_7*h_7 + x_8*h_8 + x_9*h_9;
        end
end
endmodule
```

Verification through Synthesis



GTKWave waveforms plot