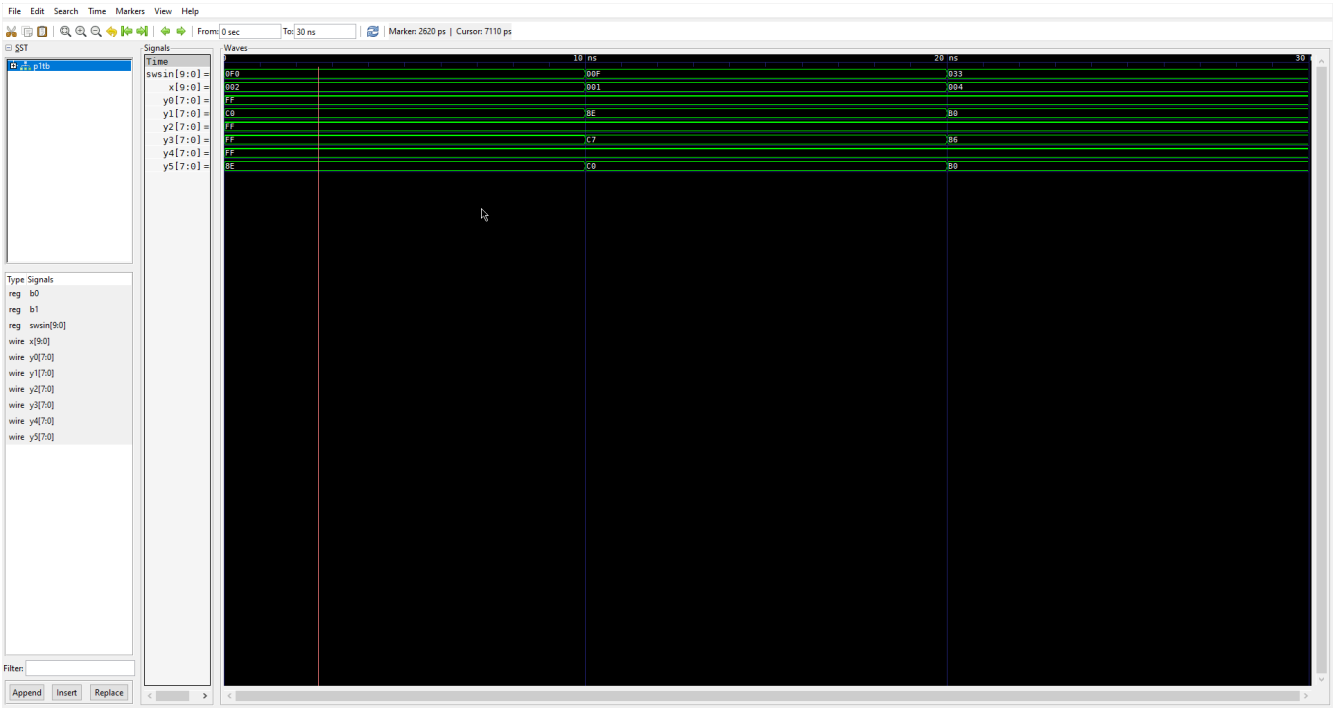
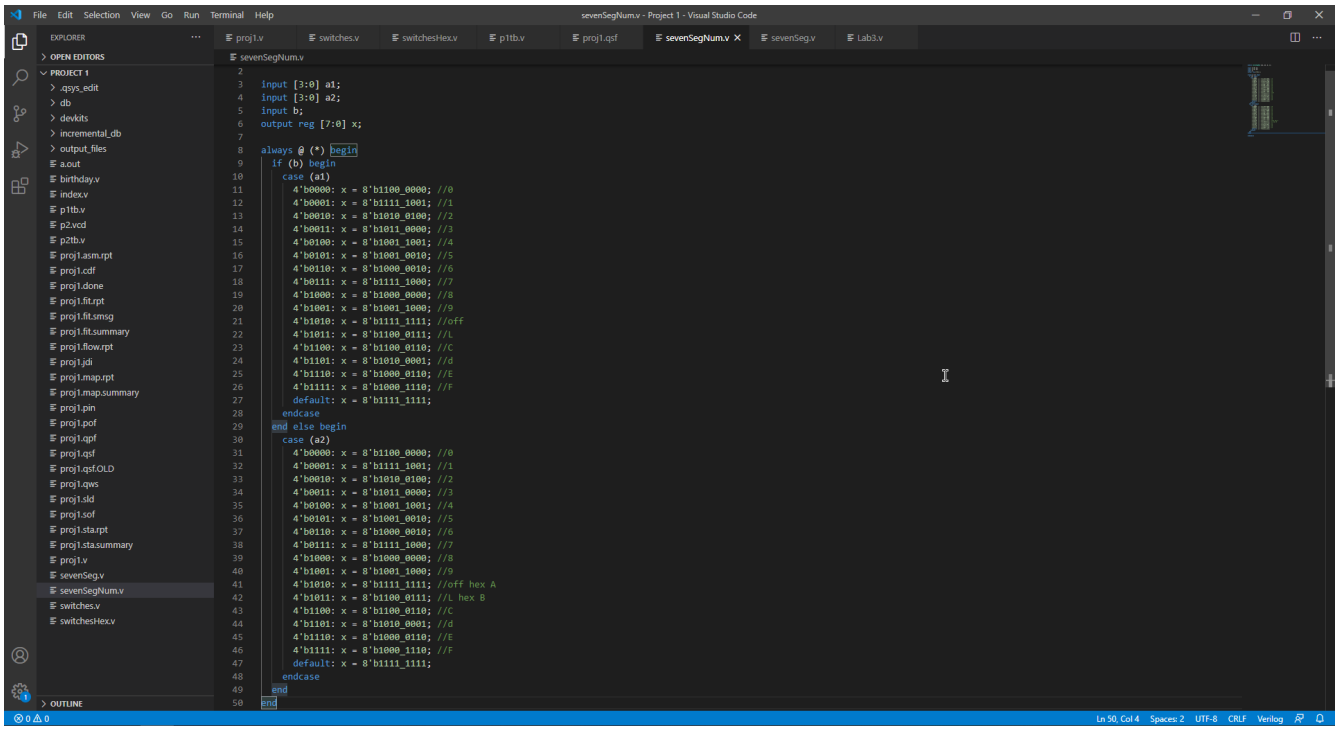
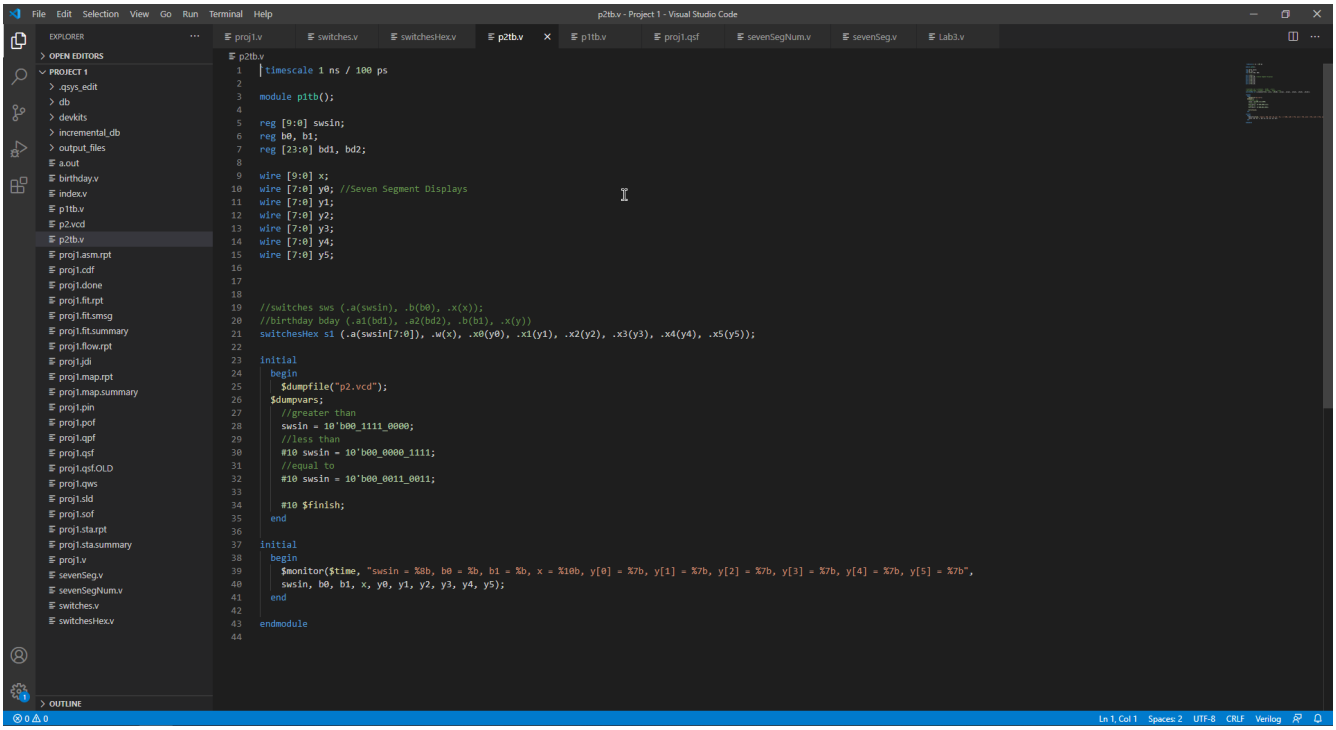
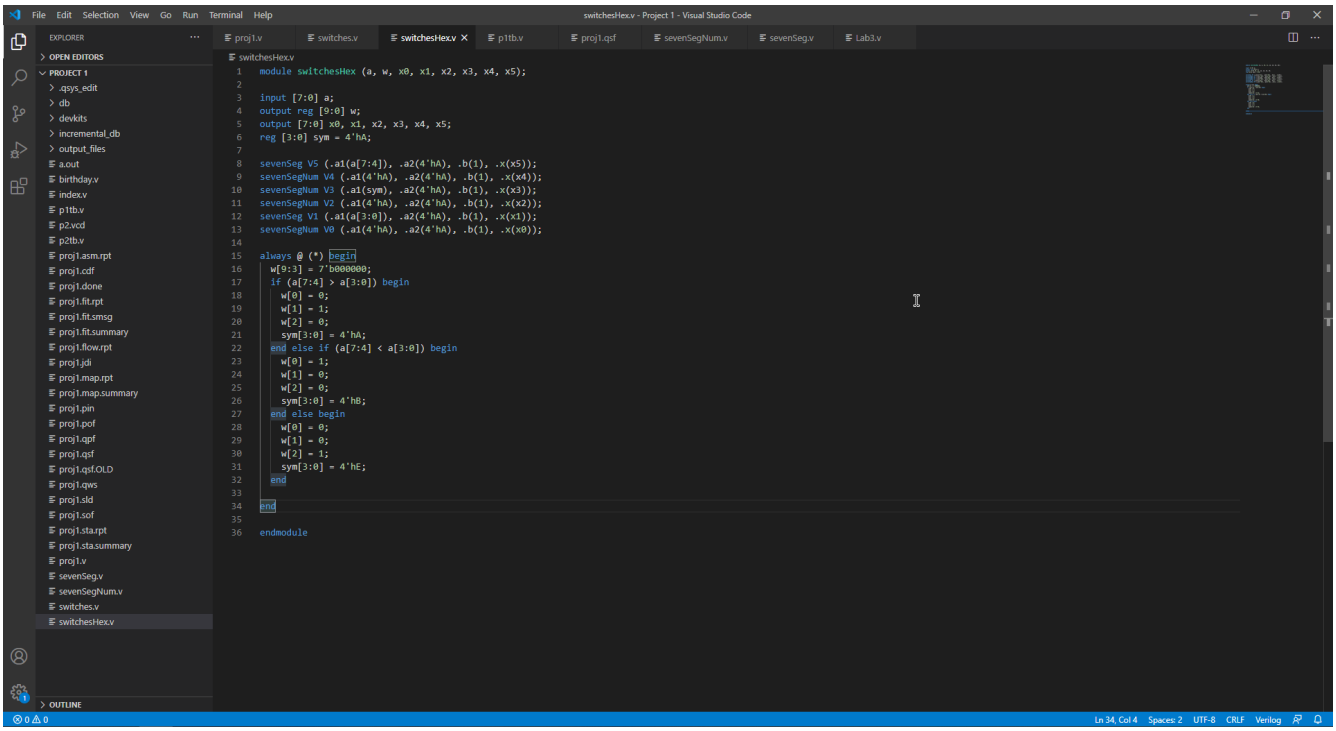
Felix Pawlowski and Thomas Hart

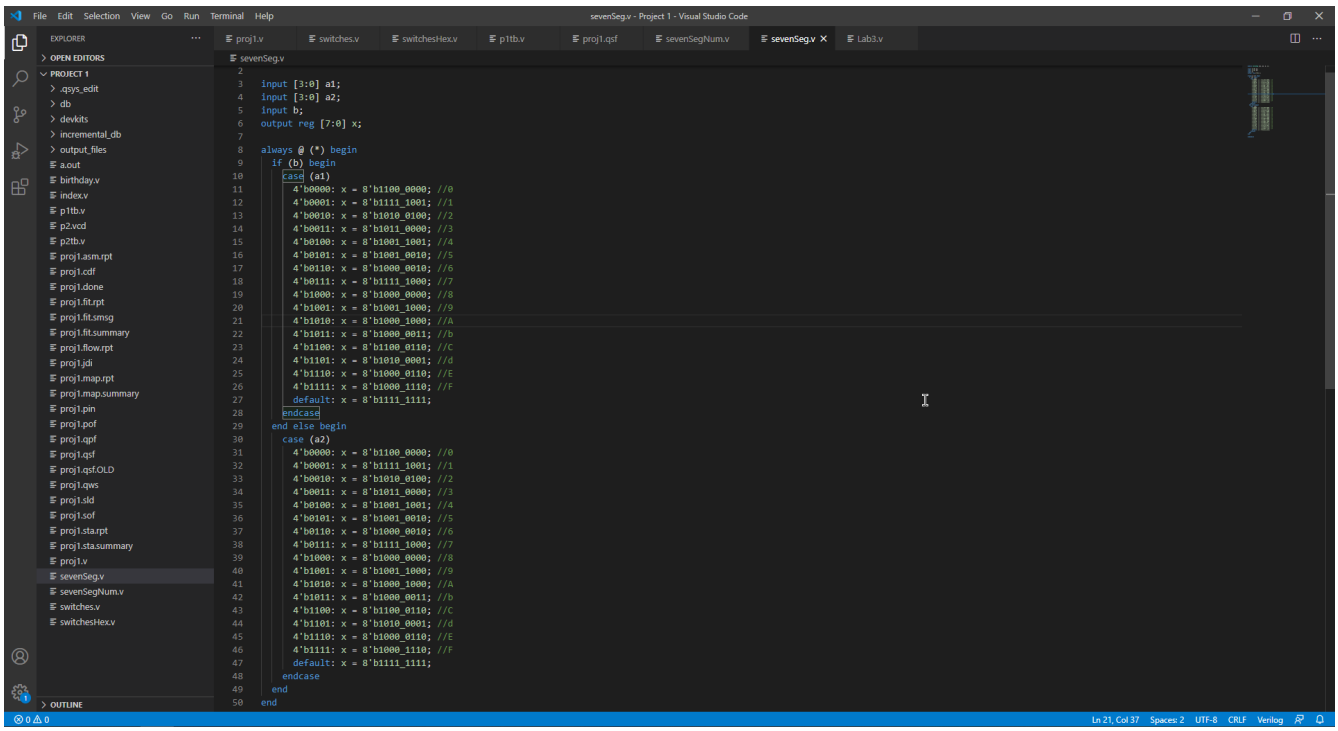
```
PS E:\SpookyHax\Digital Logic Class\Project 1> vvp a.out
VCD info: dumpfile p2.vcd opened for output.
                    0swsin = 0011110000, b0 = x, b1 = x, x = 0000000010, y[0] = 11111111, y[1] = 11000000, y[2] = 1111111
1, y[3] = 11111111, y[4] = 11111111, y[5] = 10001110
                   10swsin = 0000001111, b0 = x, b1 = x, x = 0000000001, y[0] = 11111111, y[1] = 10001110, y[2] = 1111111
1, y[3] = 11000111, y[4] = 11111111, y[5] = 11000000
                   20swsin = 0000110011, b0 = x, b1 = x, x = 0000000100, y[0] = 11111111, y[1] = 10110000, y[2] = 1111111
1, y[3] = 10000110, y[4] = 11111111, y[5] = 10110000
```




sevenSegNum.v - Project 1 - Visual Studio Code
sevenSegNum.v
input [3:0] a1;
input [3:0] a2;
input b;
output reg [7:0] x;
always @ (*) begin
if (b) begin
case (a1)
4'b0000: x = 8'b1100_0000; //0
4'b0001: x = 8'b1111_1001; //1
4'b0010: x = 8'b1010_0100; //2
4'b0011: x = 8'b1011_0000; //3
4'b0100: x = 8'b1001_1001; //4
4'b0101: x = 8'b1001_0010; //5
4'b0110: x = 8'b1000_0010; //6
4'b0111: x = 8'b1111_1000; //7
4'b1000: x = 8'b1000_0000; //8
4'b1001: x = 8'b1001_1000; //9
4'b1010: x = 8'b1111_1111; //off
4'b1011: x = 8'b1100_0111; //L
4'b1100: x = 8'b1100_0110; //C
4'b1101: x = 8'b1010_0001; //d
4'b1110: x = 8'b1000_0110; //E
4'b1111: x = 8'b1000_1110; //F
default: x = 8'b1111_1111;
endcase
end else begin
case (a2)
4'b0000: x = 8'b1100_0000; //0
4'b0001: x = 8'b1111_1001; //1
4'b0010: x = 8'b1010_0100; //2
4'b0011: x = 8'b1011_0000; //3
4'b0100: x = 8'b1001_1001; //4
4'b0101: x = 8'b1001_0010; //5
4'b0110: x = 8'b1000_0010; //6
4'b0111: x = 8'b1111_1000; //7
4'b1000: x = 8'b1000_0000; //8
4'b1001: x = 8'b1001_1000; //9
4'b1010: x = 8'b1111_1111; //off hex A
4'b1011: x = 8'b1100_0111; //L hex B
4'b1100: x = 8'b1100_0110; //C
4'b1101: x = 8'b1010_0001; //d
4'b1110: x = 8'b1000_0110; //E
4'b1111: x = 8'b1000_1110; //F
default: x = 8'b1111_1111;
endcase
end
end


p2tb.v - Project 1 - Visual Studio Code
timescale 1 ns / 100 ps
module p1tb();
reg [9:0] swsin;
reg b0, b1;
reg [23:0] bd1, bd2;
wire [9:0] x;
wire [7:0] y0; //Seven Segment Displays
wire [7:0] y1;
wire [7:0] y2;
wire [7:0] y3;
wire [7:0] y4;
wire [7:0] y5;
//switches sws (.a(swsin), .b(b0), .x(x));
//birthday bday (.a1(bd1), .a2(bd2), .b(b1), .x(y))
switchesHex s1 (.a(swsin[7:0]), .w(x), .x0(y0), .x1(y1), .x2(y2), .x3(y3), .x4(y4), .x5(y5));
initial
begin
$dumpfile("p2.vcd");
$dumpvars;
//greater than
swsin = 10'b00_1111_0000;
//less than
#10 swsin = 10'b00_0000_1111;
//equal to
#10 swsin = 10'b00_0011_0011;
#10 $finish;
end
initial
begin
$monitor($time, "swsin = %8b, b0 = %b, b1 = %b, x = %10b, y[0] = %7b, y[1] = %7b, y[2] = %7b, y[3] = %7b, y[4] = %7b, y[5] = %7b",
swsin, b0, b1, x, y0, y1, y2, y3, y4, y5);
end
endmodule


File Edit Selection View Go Run Terminal Help
switchesHex.v - Project 1 - Visual Studio Code
EXPLORER
OPEN EDITORS
PROJECT 1
.qsys_edit
db
devkits
incremental_db
output_files
a.out
birthday.v
index.v
p1tb.v
p2.vcd
p2tb.v
proj1.asm.rpt
proj1.cdf
proj1.done
proj1.fit.rpt
proj1.fit.smsg
proj1.fit.summary
proj1.flow.rpt
proj1.jdi
proj1.map.rpt
proj1.map.summary
proj1.pin
proj1.pof
proj1.qpf
proj1.qsf
proj1.qsf.OLD
proj1.qws
proj1.sld
proj1.sof
proj1.sta.rpt
proj1.sta.summary
proj1.v
sevenSeg.v
sevenSegNum.v
switches.v
switchesHex.v
OUTLINE
proj1.v
switches.v
switchesHex.v
p1tb.v
proj1.qsf
sevenSegNum.v
sevenSeg.v
Lab3.v
switchesHex.v
1 module switchesHex (a, w, x0, x1, x2, x3, x4, x5);
2
3 input [7:0] a;
4 output reg [9:0] w;
5 output [7:0] x0, x1, x2, x3, x4, x5;
6 reg [3:0] sym = 4'hA;
7
8 sevenSeg V5 (.a1(a[7:4]), .a2(4'hA), .b(1), .x(x5));
9 sevenSegNum V4 (.a1(4'hA), .a2(4'hA), .b(1), .x(x4));
10 sevenSegNum V3 (.a1(sym), .a2(4'hA), .b(1), .x(x3));
11 sevenSegNum V2 (.a1(4'hA), .a2(4'hA), .b(1), .x(x2));
12 sevenSeg V1 (.a1(a[3:0]), .a2(4'hA), .b(1), .x(x1));
13 sevenSegNum V0 (.a1(4'hA), .a2(4'hA), .b(1), .x(x0));
14
15 always @ (*) begin
16 w[9:3] = 7'b000000;
17 if (a[7:4] > a[3:0]) begin
18 w[0] = 0;
19 w[1] = 1;
20 w[2] = 0;
21 sym[3:0] = 4'hA;
22 end else if (a[7:4] < a[3:0]) begin
23 w[0] = 1;
24 w[1] = 0;
25 w[2] = 0;
26 sym[3:0] = 4'hB;
27 end else begin
28 w[0] = 0;
29 w[1] = 0;
30 w[2] = 1;
31 sym[3:0] = 4'hE;
32 end
33
34 end
35
36 endmodule
Ln 34, Col 4 Spaces: 2 UTF-8 CRLF Verilog


File Edit Selection View Go Run Terminal Help
sevenSeg.v - Project 1 - Visual Studio Code
EXPLORER
OPEN EDITORS
PROJECT 1
.qsys_edit
db
devkits
incremental_db
output_files
a.out
birthday.v
index.v
p1tb.v
p2.vcd
p2tb.v
proj1.asm.rpt
proj1.cdf
proj1.done
proj1.fit.rpt
proj1.fit.smsg
proj1.fit.summary
proj1.flow.rpt
proj1.jdi
proj1.map.rpt
proj1.map.summary
proj1.pin
proj1.pof
proj1.qpf
proj1.qsf
proj1.qsf.OLD
proj1.qws
proj1.sld
proj1.sof
proj1.sta.rpt
proj1.sta.summary
proj1.v
sevenSeg.v
sevenSegNum.v
switches.v
switchesHex.v
OUTLINE
proj1.v
switches.v
switchesHex.v
p1tb.v
proj1.qsf
sevenSegNum.v
sevenSeg.v
Lab3.v
sevenSeg.v
2
3 input [3:0] a1;
4 input [3:0] a2;
5 input b;
6 output reg [7:0] x;
7
8 always @ (*) begin
9 if (b) begin
10 case (a1)
11 4'b0000: x = 8'b1100_0000; //0
12 4'b0001: x = 8'b1111_1001; //1
13 4'b0010: x = 8'b1010_0100; //2
14 4'b0011: x = 8'b1011_0000; //3
15 4'b0100: x = 8'b1001_1001; //4
16 4'b0101: x = 8'b1001_0010; //5
17 4'b0110: x = 8'b1000_0010; //6
18 4'b0111: x = 8'b1111_1000; //7
19 4'b1000: x = 8'b1000_0000; //8
20 4'b1001: x = 8'b1001_1000; //9
21 4'b1010: x = 8'b1000_1000; //A
22 4'b1011: x = 8'b1000_0011; //b
23 4'b1100: x = 8'b1100_0110; //C
24 4'b1101: x = 8'b1010_0001; //d
25 4'b1110: x = 8'b1000_0110; //E
26 4'b1111: x = 8'b1000_1110; //F
27 default: x = 8'b1111_1111;
28 endcase
29 end else begin
30 case (a2)
31 4'b0000: x = 8'b1100_0000; //0
32 4'b0001: x = 8'b1111_1001; //1
33 4'b0010: x = 8'b1010_0100; //2
34 4'b0011: x = 8'b1011_0000; //3
35 4'b0100: x = 8'b1001_1001; //4
36 4'b0101: x = 8'b1001_0010; //5
37 4'b0110: x = 8'b1000_0010; //6
38 4'b0111: x = 8'b1111_1000; //7
39 4'b1000: x = 8'b1000_0000; //8
40 4'b1001: x = 8'b1001_1000; //9
41 4'b1010: x = 8'b1000_1000; //A
42 4'b1011: x = 8'b1000_0011; //b
43 4'b1100: x = 8'b1100_0110; //C
44 4'b1101: x = 8'b1010_0001; //d
45 4'b1110: x = 8'b1000_0110; //E
46 4'b1111: x = 8'b1000_1110; //F
47 default: x = 8'b1111_1111;
48 endcase
49 end
50 end
Ln 21, Col 37 Spaces: 2 UTF-8 CRLF Verilog