HW1 Solutions (CS552 Spring 2013)

Grading Scheme: (Total: 60)

- Written part of each problem carries 5 points (Total: 15 points)
- Verilog submission of each problem carries 15 points (Total: 45 points)

Grading of written submission:

1. Any missing waveform: -2
2. Ques 2.5 missing: -0.5
3. State diagram missing: -2

Grading of online submission:

1. Points for a compiling design:	3
2. Points for functional tests(40 to 100 in number):	12 (One point deducted per failure)
3. Penalty for errors in .vcheck.out:	50% of the points awarded for the particular problem

Problem1)

```
////////////////////////////////////////////////quadmux4_1.v
module quadmux4_1 (InA, InB, InC, InD, S, Out);

input [3:0] InA;
input [3:0] InB;
input [3:0] InC;
input [3:0] InD;
input [1:0] S;
output [3:0] Out;

mux4_1 mux[3:0] (.InA(InA),.InB(InB),.InC(InC),.InD(InD),.S(S),.Out(Out));

endmodule
////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////mux4_1.v
module mux4_1 (InA, InB, InC, InD, S, Out);

input InA;
input InB;
input InC;
input InD;
input [1:0] S;
output Out;

wire m1;
wire m2;

mux2_1 mux_ab (.InA(InA),.InB(InB),.S(S[0]),.Out(m1));
mux2_1 mux_cd (.InA(InC),.InB(InD),.S(S[0]),.Out(m2));
mux2_1 mux_out (.InA(m1),.InB(m2),.S(S[1]),.Out(Out));

endmodule
////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////mux2_1.v
module mux2_1 (InA, InB, S, Out);

input InA;
input InB;
input S;
output Out;

wire S_;
wire intA;
wire intB;

not1 sbar(.in1(S),.out(S_));
nand2 A_nand (.in1(S_),.in2(InA), .out(intA));
nand2 B_nand (.in1(S),.in2(InB), .out(intB));
nand2 Out_nand (.in1(intA), .in2(intB), .out(Out));
```

```
endmodule
////////////////////////////////////////////////////////////////////////
```

Problem2)

```
////////////////////////////////////////////////fulladder16.v
module fulladder16 (A,B,SUM,CO);

input [15:0] A;
input [15:0] B;
output [15:0] SUM;
output CO;

wire C_3_4;
wire C_7_8;
wire C_11_12;

fulladder4 bits_3_0   (.A(A[3:0]),.B(B[3:0]),.CI(1'b0),.S(SUM[3:0]),.CO(C_3_4));
fulladder4 bits_7_4   (.A(A[7:4]),.B(B[7:4]),.CI(C_3_4),.S(SUM[7:4]),.CO(C_7_8));
fulladder4 bits_11_8  (.A(A[11:8]),.B(B[11:8]),.CI(C_7_8),.S(SUM[11:8]),.CO(C_11_12));
fulladder4 bits_15_12 (.A(A[15:12]),.B(B[15:12]),.CI(C_11_12),.S(SUM[15:12]),.CO(CO));

endmodule
////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////fulladder4.v
module fulladder4 (A,B,CI,S,CO);

input [3:0] A;
input [3:0] B;
input CI;
output [3:0] S;
output CO;

wire C_0_1;
wire C_1_2;
wire C_2_3;

fulladder bit0 (.A(A[0]),.B(B[0]),.Cin(CI),.S(S[0]),.Cout(C_0_1));
fulladder bit1 (.A(A[1]),.B(B[1]),.Cin(C_0_1),.S(S[1]),.Cout(C_1_2));
fulladder bit2 (.A(A[2]),.B(B[2]),.Cin(C_1_2),.S(S[2]),.Cout(C_2_3));
fulladder bit3 (.A(A[3]),.B(B[3]),.Cin(C_2_3),.S(S[3]),.Cout(CO));

endmodule
////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////fulladder.v
module fulladder (A,B,Cin,S,Cout);

input A;
input B;
input Cin;
output S;
output Cout;

wire w1;
wire w2;
wire w3;

xor2 x1 (.in1(A),.in2(B),.out(w1));
xor2 x2 (.in1(w1),.in2(Cin),.out(S));

nand2 n1 (.in1(A),.in2(B),.out(w2));
nand2 n2 (.in1(w1),.in2(Cin),.out(w3));
nand2 n3 (.in1(w2),.in2(w3),.out(Cout));

endmodule
////////////////////////////////////////////////////////////////////////
```

Problem 3)

```
////////////////////////////////////////////////seqdec_85.v
module seqdec_85 (InA, Clk, Reset, Out);

input Clk;
input Reset;
input InA;

output Out;

wire [3:0] state;
```

```
reg [3:0] next_state;

assign Out = state[3];

dff state_flops[3:0] (
  .d(next_state),
  .q(state),
  .clk(Clk),
  .rst(Reset)
);

always @(InA or state)
begin
  casex({InA,state})
    5'b0_0000:
      next_state = 4'h0;
    5'b1_0000:
      next_state = 4'h1;

    5'b0_0001:
      next_state = 4'h2;
    5'b1_0001:
      next_state = 4'h1;

    5'b0_0010:
      next_state = 4'h3;
    5'b1_0010:
      next_state = 4'h1;

    5'b0_0011:
      next_state = 4'h4;
    5'b1_0011:
      next_state = 4'h1;

    5'b0_0100:
      next_state = 4'h5;
    5'b1_0100:
      next_state = 4'h1;

    5'b0_0101:
      next_state = 4'h0;
    5'b1_0101:
      next_state = 4'h6;

    5'b0_0110:
      next_state = 4'h7;
    5'b1_0110:
      next_state = 4'h1;

    5'b0_0111:
      next_state = 4'h3;
    5'b1_0111:
      next_state = 4'h8;

    5'b0_1000:
      next_state = 4'h2;
    5'b1_1000:
      next_state = 4'h1;

    default:
      next_state = {4{1'bx}};
  endcase
end

endmodule
////////////////////////////////////////////////////////////////////////
```