George Suarez

CSE 310

Homework 4

1. Textbook Problems 4.37: Write the HDL gate-level hierarchical description of a four-bit adder-subtractor for unsigned binary numbers. The circuit is similar to Fig.4.13 but without output V. You can instantiate the four-bit full adder described in HDL Example 4.2. (see Problems 4.13 and 4.40.) Write a test bench to test your module.

*4bit\_adder\_subtractor.v*

module half\_adder(output S, C, input x, y);

// Instantiate primitive gates

xor (S, x, y);

and (C, x, y);

endmodule

module full\_adder(output S, C, input x, y, z);

wire S1, C1, C2;

// Instantiate half\_adders

half\_adder HA1 (S1, C1, x, y);

half\_adder HA2 (S, C2, S1, z);

or G1 (C, C2, C1);

endmodule

module ripple\_carry\_4\_bit\_adder(output [3:0] Sum, output C4,

input [3:0] A, B, input C0);

wire C1, C2, C3; // Intermediate Carries

// Instantiate full\_adders

full\_adder FA0 (Sum[0], C1, A[0], B[0], C0),

FA1 (Sum[1], C2, A[1], B[1], C1),

FA2 (Sum[2], C3, A[2], B[2], C2),

FA3 (Sum[3], C4, A[3], B[3], C3);

endmodule

*4bit\_adder\_subtractor\_tb.v:*

module four\_bit\_adder\_subtractor\_tb;

// Inputs

reg [3:0] A;

reg [3:0] B;

reg Cin;

// Outputs

wire Cout;

wire [3:0] Sum;

// Instantiate the Unit Under Test (UUT)

ripple\_carry\_4\_bit\_adder uut(.Sum(Sum), .C4(Cout), .A(A), .B(B), .C0(Cin));

initial

begin

$display("Time\t A \t B \t Cin \t | Sum \t Cout");

$monitor("%3g\t %b \t %b \t %b \t | %b %b", $time, A, B, Cin, Sum, Cout);

end

initial

begin

A = 0000; B = 0001; Cin = 0;

#10 A=0100; B=1011; Cin=0;

#10 A=1000; B=0010; Cin=0;

#10 A=0100; B=0011; Cin=1;

#10 A=1100; B=0011; Cin=1;

#10 A=1100; B=0100; Cin=1;

#10 A=0100; B=0000; Cin=0;

#10 A=0100; B=0001; Cin=0;

#10 A=0100; B=0010; Cin=0;

#10 A=0100; B=0000; Cin=1;

#10 A=0100; B=0001; Cin=1;

#10 A=0100; B=0010; Cin=1;

#10 $finish;

end

endmodule

Output:

Time A B Cin |Sum Cout

0 0000 0001 0 | 0001 0

10 0100 0011 0 | 0111 0

20 1000 1010 0 | 0010 1

30 0100 1011 1 | 0000 1

40 1100 1011 1 | 1000 1

50 1100 0100 1 | 0001 1

60 0100 0000 0 | 0100 0

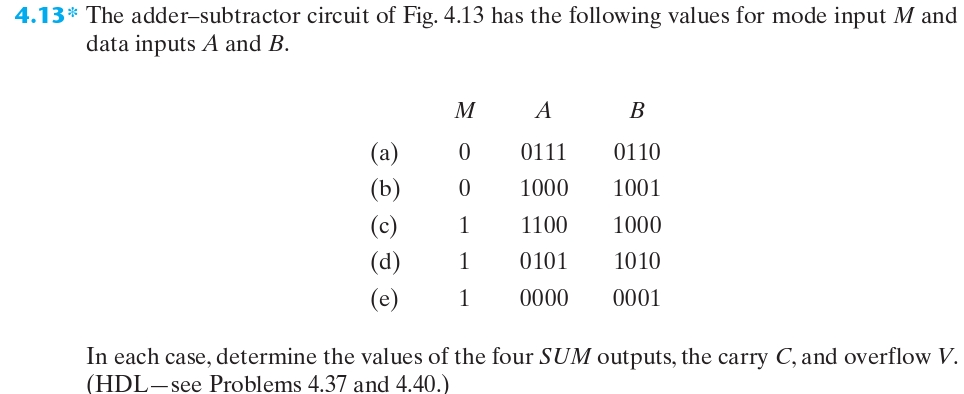
70 0100 0001 0 | 0101 0

80 0100 1010 0 | 1110 0

90 0100 0000 1 | 0101 0

100 0100 0001 1 | 0110 0

1. 0100 1010 1 | 1111 0
2. Textbook Problems 4.13 (p.184):





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin = 0 | C3 | C2 | C1 | C0 |
|  | 1 | 1 | 0 | 0 |
| A | 0 | 1 | 1 | 1 |
| B | 0 | 1 | 1 | 0 |
| A + B =  Cout = 0 | 1 | 1 | 0 | 1 |

Sum = 1101

C = Cout = 0

V = (Cout ^ C3) = 0 ^ 1 = 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin = 0 | C3 | C2 | C1 | C0 |
|  | 0 | 0 | 0 | 0 |
| A | 1 | 0 | 0 | 0 |
| B | 1 | 0 | 0 | 1 |
| A + B =  Cout = 1 | 0 | 0 | 0 | 1 |

Sum = 0001

C = Cout = 1

V = (Cout ^ C3) = (1 ^ 0) = 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin = 1 | C3 | C2 | C1 | C0 |
|  | 1 | 1 | 1 | 1 |
| A | 1 | 1 | 0 | 0 |
| B | 0 | 1 | 1 | 1 |
| A + B =  Cout = 1 | 0 | 1 | 0 | 0 |

Sum = 0111

C = Cout = 1

V = (Cout ^ C3) = (1 ^ 1) = 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin = 1 | C3 | C2 | C1 | C0 |
|  | 1 | 0 | 1 | 1 |
| A | 0 | 1 | 0 | 1 |
| B | 0 | 1 | 0 | 1 |
| A + B =  Cout = 0 | 1 | 0 | 1 | 1 |

Sum = 1011

C = Cout = 0

V = (Cout ^ C3) = (0 ^ 1) = 1

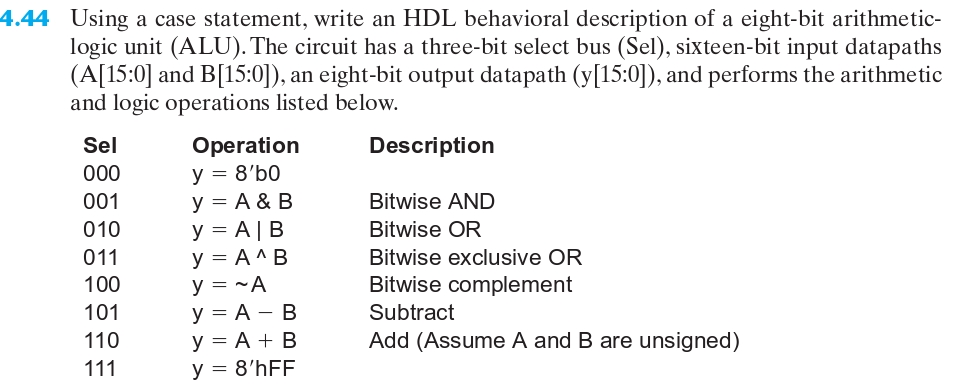
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin = 1 | C3 | C2 | C1 | C0 |
|  | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 0 | 1 |
| A + B =  Cout = 0 | 1 | 1 | 1 | 1 |

Sum = 1111

C = Cout = 0

V = (Cout ^ C3) = (0 ^ 0) = 0

1. Textbook Problems 4.44:



*ALU.v:*

module ALU ( output [7:0] Y, input[7:0] A, B, input [2:0] Sel);

always @ (A, B, Sel)

begin

y = 0;

case (Sel)

3'b000: y = 8'b0;

3'b001: y = A & B;

3'b010: y = A | B;

3'b011: y = A ^ B;

3'b100: y = A + B;

3'b101: y = A - B;

3'b110: y = ~A;

3'b111: y = 8'hFF;

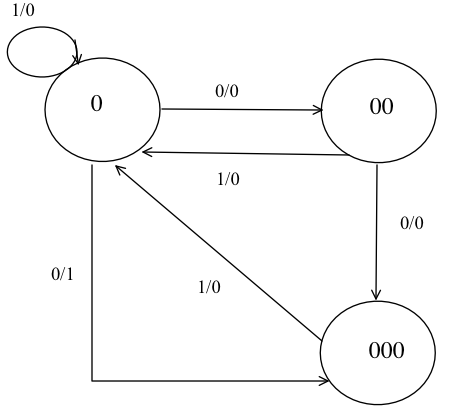
endcase

end

endmodule

~

1. Design a binary sequence detector that detects the sequence *000*. Overlap is allowed. You may use either D flip-flops or JK flip-flops. Write a Verilog program to verify your design.
2. State Diagram



(ii) Assign States

(iii) State Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |

Using D flip-flops:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| I |  | 00 | 01 | 11 | 10 |
|  | 0 | 0 | 0 | X | 1 |
|  | 1 | 0 | 1 | X | 0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| I |  | 00 | 01 | 11 | 10 |
|  | 0 | 1 | 0 | X | 0 |
|  | 1 | 0 | 0 | X | 1 |

(iv) Circuit Diagram:



**Self-Evaluation:** I have answered every question in this homework assignment to the best of my ability. I would give myself the maximum points allowed for this assignment which is 55 points.