**DIGITAL LOGIC DESIGN LAB (EET1211)**

**LAB VI: DESIGN OF MAGNITUDE COMPARATOR, DECODER AND MULTIPLEXER CIRCUIT USING HDL**

**Siksha ‘O’ Anusandhan Deemed to be University, Bhubaneswar**

|  |  |  |  |
| --- | --- | --- | --- |
| **Branch:** Computer Science and Engineering **Section:** D | | | |
| **S. No.** | **Name** | **Registration No.** | **Signature** |
| 1 | Saswat Mohanty | 1941012407 | **D:\Pics and Sign\sign.jpg** |

**Marks: \_\_\_\_\_\_/10**

**Remarks:**

**Teacher’s Signature**

**I. OBJECTIVE:**

1. Design a combinational circuit for a 1 bit magnitude comparator.
2. Design a combinational circuit for a 2 X 1 Multiplexer that will select the binary information from one of the two input lines and direct it to a single output line based on the value of a selection line.
3. Design a combinational circuit for a full adder using 3 to 8 line decoder and external OR gates.

**II. PRE-LAB**

**For Obj. 1:**

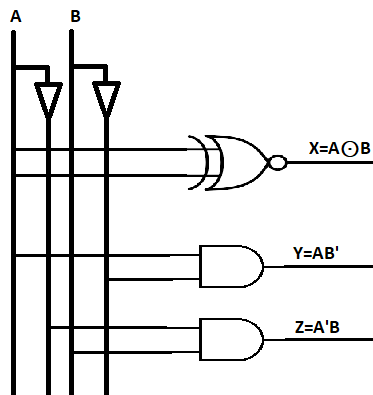
1. **Write the truth table for the circuit.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **X** | **Y** | **Z** |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |

1. **Derive the Minimized Boolean expression for each output of the circuit.**

Simplified Expression: **X = A ⊙ B**, **Y = AB’**, **Z = A’B**

1. **Draw the logic diagram for the circuit.**



1. **Write HDL code.**

**design.sv:**

*`default\_nettype none*

*module mod (*

*input A,*

*input B,*

*output X,*

*output Y,*

*output Z*

*);*

*// dataflow model*

*assign X=~(A^B);*

*assign Y= A&&~B;*

*assign Z= ~A&&B;*

*// gate-level model*

*xnor(X,A,B);*

*and a(Y,A,~B),*

*a1(Z,~A,B);*

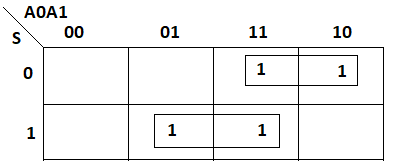
*endmodule*

**For Obj. 2:**

1. **Write the truth table for the circuit.**

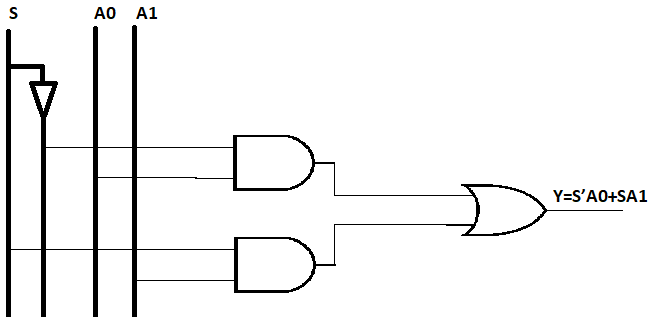
|  |  |  |  |
| --- | --- | --- | --- |
| **S** | **A0** | **A1** | **Y** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

1. **Derive the Minimized Boolean expression for each output of the circuit.**



Simplified Expression: **Y = S’A0 + SA1**

1. **Draw the logic diagram for the circuit.**



1. **Write HDL code.**

**design.sv:**

*`default\_nettype none*

*module mod (*

*input S,A0,A1,*

*output Y*

*);*

*wire W1,W2;*

*// dataflow model*

*assign Y= ~S&&A0||S&&A1;*

*// gate-level model*

*and a(W1,~S,A0),*

*a1(W2,S,A1);*

*or(Y,W1,W2);*

*endmodule*

**For Obj. 3:**

1. **Write the truth table for the circuit.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **Sum** | **Carry** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

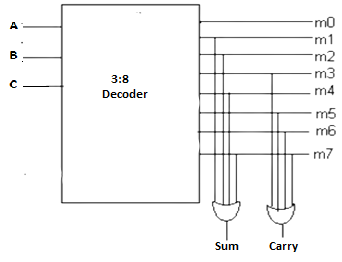
1. **Derive the Minimized Boolean expression for each output of the circuit.**

Expression:

Sum = **∑ m (1,2,4,7)**

Carry = **∑ m (3,5,6,7)**

1. **Draw the logic diagram for the circuit.**



1. **Write HDL code.**

**design.sv:**

*`default\_nettype none*

*module lab6 (A,B,C,Sum,Carry);*

*input A,B,C;*

*output Sum,Carry,D0,D1,D2,D3,D4,D5,D6,D7;*

*wire P,Q,R,S,T,U,V;*

*// dataflow model*

*assign D0=~A&&~B&&~C;*

*assign D1=~A&&~B&&C;*

*assign D2=~A&&B&&~C;*

*assign D3=~A&&B&&C;*

*assign D4=A&&~B&&~C;*

*assign D5=A&&~B&&C;*

*assign D6=A&&B&&~C;*

*assign D7=A&&B&&C;*

*assign Sum=~A&&~B&&C||~A&&B&&~C||A&&~B&&~C||A&&B&&C;*

*assign Carry= B&&C||A&&C||A&&B;*

*// gate-level model*

*and a1(D0,~A,~B,~C),*

*a2(D1,~A,~B,C),*

*a3(D2,~A,B,~C),*

*a4(D3,~A,B,C),*

*a5(D4,A,~B,~C),*

*a6(D5,A,~B,C),*

*a7(D6,A,B,~C),*

*a8(D7,A,B,C),*

*a9(P,~A,~B,C),*

*a10(Q,~A,B,~C),*

*a11(R,A,~B,~C),*

*a12(S,A,B,C),*

*a13(T,B,C),*

*a14(U,A,C),*

*a15(V,A,B);*

*or o1(Sum,P,Q,R,S),*

*o2(Carry,T,U,V);*

*endmodule*

**III. LAB:**

1. **Design a combinational circuit for a 1 bit magnitude comparator.**

**HDL Program:**

**design.sv:**

*`default\_nettype none*

*module mod (*

*input A,*

*input B,*

*output X,*

*output Y,*

*output Z*

*);*

*// dataflow model*

*assign X=~(A^B);*

*assign Y= A&&~B;*

*assign Z= ~A&&B;*

*// gate-level model*

*xnor(X,A,B);*

*and a(Y,A,~B),*

*a1(Z,~A,B);*

*endmodule*

**testbench.sv:**

*`default\_nettype none*

*module tb\_mod;*

*reg a, b;*

*wire x,y,z;*

*mod h\_dut(a,b,x,y,z);*

*initial*

*begin*

*$dumpfile("dump.vcd");*

*$dumpvars(0, h\_dut);*

*$display("Lab 6 Obj 1");*

*#1*

*a <= 0;*

*b <= 0;*

*#1*

*#1*

*a <= 0;*

*b <= 1;*

*#1*

*#1*

*a <= 1;*

*b <= 0;*

*#1*

*#1*

*a <= 1;*

*b <= 1;*

*#1*

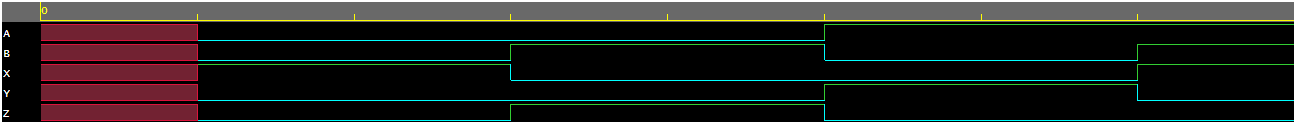
*$finish();*

*end*

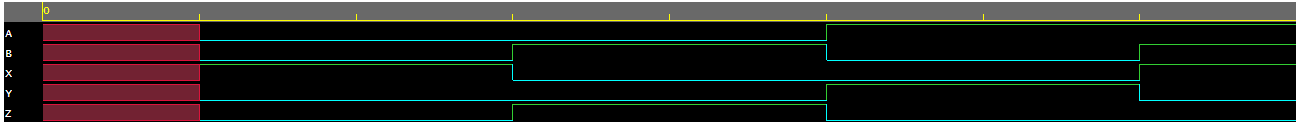
*endmodule*

**Links:** <https://www.edaplayground.com/x/nGTL>

**EP Waveform:**



***Dataflow model***



***Gate-level model***

**Observation:**

The following Truth table was obtained from the above EP Waveform.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **X** | **Y** | **Z** |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |

1. **Design a combinational circuit for a 2 X 1 Multiplexer that will select the binary information from one of the two input lines and direct it to a single output line based on the value of a selection line.**

**HDL Program:**

**design.sv:**

*`default\_nettype none*

*module mod (*

*input S,A0,A1,*

*output Y*

*);*

*wire W1,W2;*

*// dataflow model*

*assign Y= ~S&&A0||S&&A1;*

*// gate-level model*

*and a(W1,~S,A0),*

*a1(W2,S,A1);*

*or(Y,W1,W2);*

*endmodule*

**testbench.sv:**

*`default\_nettype none*

*module tb\_mod;*

*reg S,A0,A1;*

*wire Y;*

*mod h\_dut(S,A0,A1,Y);*

*initial*

*begin*

*$dumpfile("dump.vcd");*

*$dumpvars(0, h\_dut);*

*$display("Lab 6 Obj 2");*

*#1*

*S <= 0;*

*A0 <= 0;*

*A1 <= 0;*

*#1*

*#1*

*S <= 0;*

*A0 <= 0;*

*A1 <= 1;*

*#1*

*#1*

*S <= 0;*

*A0 <= 1;*

*A1 <= 0;*

*#1*

*#1*

*S <= 0;*

*A0 <= 1;*

*A1 <= 1;*

*#1*

*#1*

*S <= 1;*

*A0 <= 0;*

*A1 <= 0;*

*#1*

*#1*

*S <= 1;*

*A0 <= 0;*

*A1 <= 1;*

*#1*

*#1*

*S <= 1;*

*A0 <= 1;*

*A1 <= 0;*

*#1*

*#1*

*S <= 1;*

*A0 <= 1;*

*A1 <= 1;*

*#1*

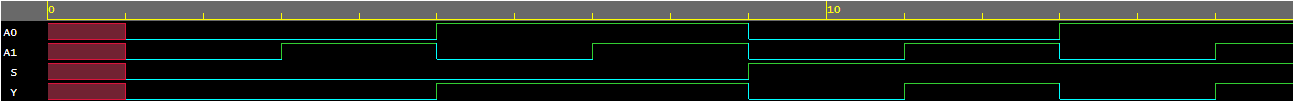
*$finish();*

*end*

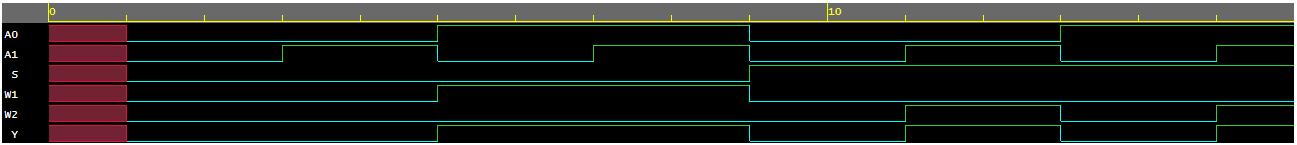
*endmodule*

**Links:** <https://www.edaplayground.com/x/CMsS>

**EP Waveform:**



***Dataflow model***



***Gate-level model***

**Observation:**

The following Truth table was obtained from the above EP Waveform.

|  |  |  |  |
| --- | --- | --- | --- |
| **S** | **A0** | **A1** | **Y** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

1. **Design a combinational circuit for a full adder using 3 to 8 line decoder and external OR gates.**

**HDL Program:**

**design.sv:**

*`default\_nettype none*

*module lab6 (A,B,C,Sum,Carry);*

*input A,B,C;*

*output Sum,Carry,D0,D1,D2,D3,D4,D5,D6,D7;*

*wire P,Q,R,S,T,U,V;*

*// dataflow model*

*assign D0=~A&&~B&&~C;*

*assign D1=~A&&~B&&C;*

*assign D2=~A&&B&&~C;*

*assign D3=~A&&B&&C;*

*assign D4=A&&~B&&~C;*

*assign D5=A&&~B&&C;*

*assign D6=A&&B&&~C;*

*assign D7=A&&B&&C;*

*assign Sum=~A&&~B&&C||~A&&B&&~C||A&&~B&&~C||A&&B&&C;*

*assign Carry= B&&C||A&&C||A&&B;*

*// gate-level model*

*and a1(D0,~A,~B,~C),*

*a2(D1,~A,~B,C),*

*a3(D2,~A,B,~C),*

*a4(D3,~A,B,C),*

*a5(D4,A,~B,~C),*

*a6(D5,A,~B,C),*

*a7(D6,A,B,~C),*

*a8(D7,A,B,C),*

*a9(P,~A,~B,C),*

*a10(Q,~A,B,~C),*

*a11(R,A,~B,~C),*

*a12(S,A,B,C),*

*a13(T,B,C),*

*a14(U,A,C),*

*a15(V,A,B);*

*or o1(Sum,P,Q,R,S),*

*o2(Carry,T,U,V);*

*endmodule*

**testbench.sv:**

*`default\_nettype none*

*module dl\_lab6;*

*reg a,b,c;*

*wire sum,carry;*

*lab6 h\_dut(a,b,c,sum,carry);*

*initial*

*begin*

*$dumpfile("dump.vcd");*

*$dumpvars(0, h\_dut);*

*$display("Lab 6 Obj 3");*

*#1*

*a <= 0;*

*b <= 0;*

*c <= 0;*

*#1*

*#1*

*a <= 0;*

*b <= 0;*

*c <= 1;*

*#1*

*#1*

*a <= 0;*

*b <= 1;*

*c <= 0;*

*#1*

*#1*

*a <= 0;*

*b <= 1;*

*c <= 1;*

*#1*

*#1*

*a <= 1;*

*b <= 0;*

*c <= 0;*

*#1*

*#1*

*a <= 1;*

*b <= 0;*

*c <= 1;*

*#1*

*#1*

*a <= 1;*

*b <= 1;*

*c <= 0;*

*#1*

*#1*

*a <= 1;*

*b <= 1;*

*c <= 1;*

*#1*

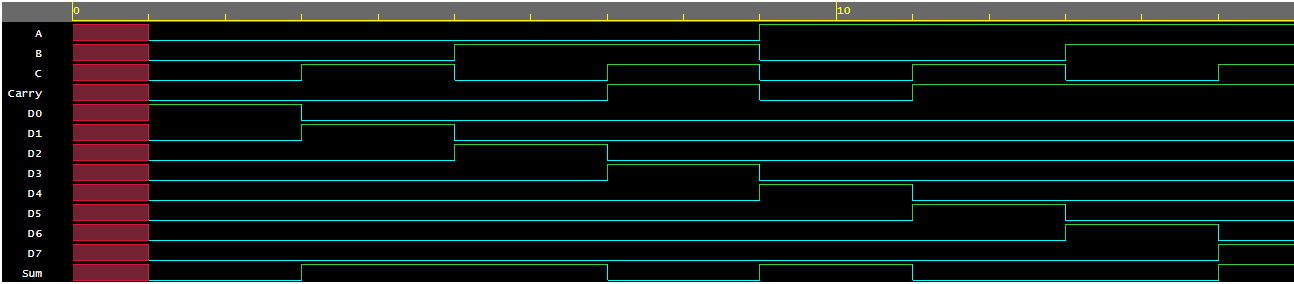
*$finish();*

*end*

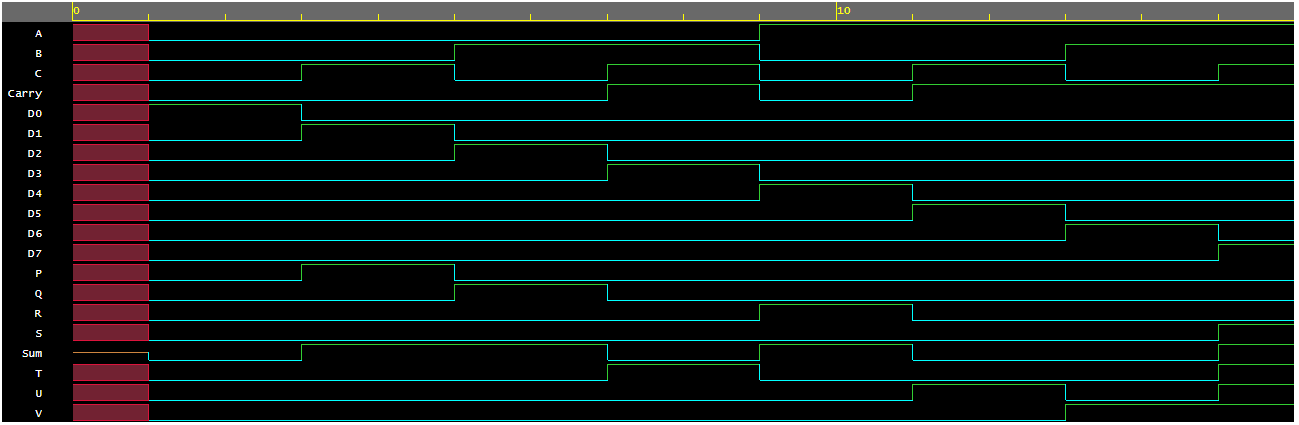
*endmodule*

**Links:** <https://www.edaplayground.com/x/LN3N>

**EP Waveform:**



***Dataflow model***



***Gate-level model***

**Observation:**

The following Truth table was obtained from the above EP Waveform.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **Sum** | **Carry** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Conclusion:**

**Objective 1:** It can be concluded that to design a combinational circuit for a 1 - bit magnitude comparator 1 xnor and 2 and and not are required and circuit leads to the functions

**X = A** ⊙ **B**

**Y = AB’**

**Z = A’B**

**Objective 2:** It can be concluded that to design a combinational circuit for a 2 X 1 Multiplexer 2 and and 1 or gates are required and circuit leads to the function

**Y=S’A0+SA1**

**Objective 3:** It can be concluded that to implement a full adder using 3-to-8-line decoder and external OR gates a decoder and 2 or gates are required and circuit leads to function

**Sum=** **∑m (1,2,4,7)**

**Carry= ∑m (3,5,6,7)**

**IV. POST LAB:**

1. **Logically derive the Boolean expressions for the output variables of a 2-bit magnitude comparator.**

**Ans: -** **A > B:** A1B1’ + A0B1’B0’ + A1A0B0’

**A=B:** (A0 ⊙ B0) (A1 ⊙ B1)

**A<B:** A1’B1 + A0’B1B0 + A1’A0’B0

1. **Why is a multiplexer known as data selector?**

**Ans: -** Multiplexer known as data selector because it selects which data input is to be sent.

**V. HDL PROGRAM LINK:**

**Objective 1:** <https://www.edaplayground.com/x/nGTL>

**Objective 2:** <https://www.edaplayground.com/x/CMsS>

**Objective 3:** <https://www.edaplayground.com/x/LN3N>