**DIGITAL LOGIC DESIGN LAB (EET1211)**

**LAB V: DESIGN AND TEST VARIOUS CODE**

**CONVERTER CIRCUITS USING HDL**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Branch: Section:** | | | |
| **S. No.** | **Name** | **Registration No.** | **Signature** |
| 1 | Saswat Mohanty | 1941012407 | **E:\sign.jpg** |

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

**Remarks:**

**Teacher’s Signature**

**I. OBJECTIVE**

1. Design a combinational circuit with four input lines that represent a decimal digit in BCD and four output lines that generate the 9’s complement of the input digit.
2. Design a combinational circuit with four inputs and four outputs that converts a 4bit binary number into the equivalent 4bit Gray code.
3. Design a combinational circuit that accepts a 2-bit number and generates an output binary number equal to the square of the input number.

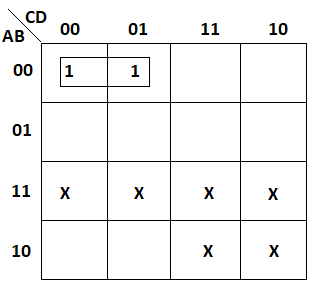
**II. PRE-LAB**

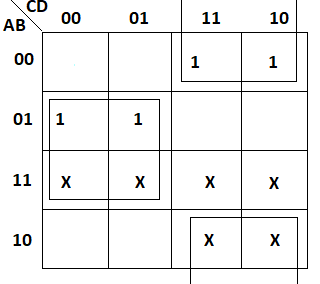
**For Objective - 1:**

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **W** | **X** | **Y** | **Z** |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X |

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

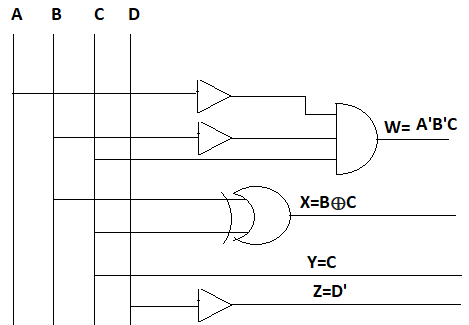




Simplified Expression: **W = A’B’C X = BC’ + B’C = B⊕C**

Simplified Expression: **Y = C Z = D’**

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



1. **Write HDL code.**

**design.sv:**

*`default\_nettype none*

*module lab5 (*

*input A,*

*input B,*

*input C,*

*input D,*

*output W,X,Y,Z*

*);*

*// dataflow model*

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

*assign X=B^C;*

*assign Y=C;*

*assign Z=~D;*

*// gate-level model*

*and(W,~A,~B,~C);*

*xor(X,B,C);*

*buf(Y,C);*

*not(Z,D);*

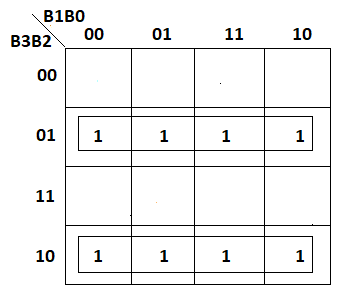
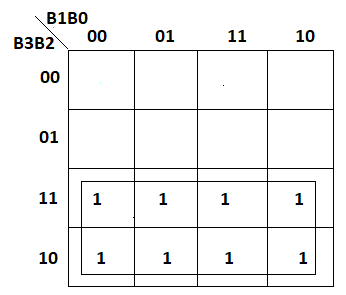
*endmodule*

**For Objective - 2:**

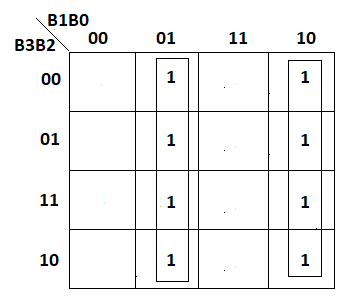
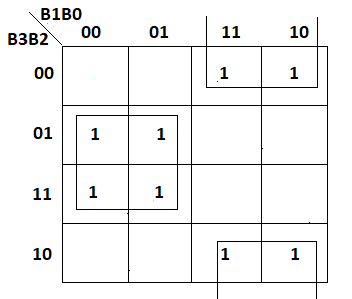
1. **Write the truth table for the circuit.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **B3** | **B2** | **B1** | **B0** | **G3** | **G2** | **G1** | **G0** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

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

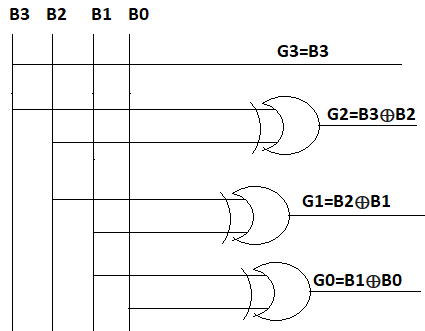


Simplified Expression: **G3 = B3 G2 = B3 ⊕ B2**

****

Simplified Expression: **G1 = B2 ⊕ B1 G0 = B1 ⊕ B0**

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



1. **Write HDL code.**

**design.sv:**

*`default\_nettype none*

*module lab5 (*

*input B3,*

*input B2,*

*input B1,*

*input B0,*

*output G3,G2,G1,G0*

*);*

*//dataflow model*

*assign G3=B3;*

*assign G2=B2^B3;*

*assign G1=B1^B2;*

*assign G0=B0^B1;*

*// gate-level model*

*buf(G3,B3);*

*xor x1(G2,B2,B3),*

*x2(G1,B2,B1),*

*x3(G0,B1,B0);*

*endmodule*

**For Objective - 3:**

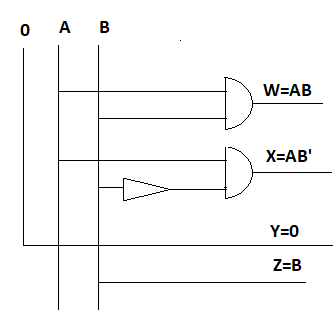
1. **Write the truth table for the circuit.**

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

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

Simplified Expression: **W = AB, X = AB’, Y = 0, Z = 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 W,X,Y,Z*

*);*

*//dataflow model*

*assign W=A&&B;*

*assign X=A&&~B;*

*assign Y=0;*

*assign Z=B;*

*//gate-level model*

*and a1(W,A,B),*

*a2(X,A,~B);*

*buf b1(Y,0),*

*b2(Z,B);*

*endmodule*

**III. LAB:**

1. **Design a combinational circuit with four input lines that represent a decimal digit in BCD and four output lines that generate the 9’s complement of the input digit.**

**HDL Program:**

**design.sv:**

*`default\_nettype none*

*module lab5 (*

*input A,*

*input B,*

*input C,*

*input D,*

*output W,X,Y,Z*

*);*

*// dataflow model*

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

*assign X=B^C;*

*assign Y=C;*

*assign Z=~D;*

*// gate-level model*

*and(W,~A,~B,~C);*

*xor(X,B,C);*

*buf(Y,C);*

*not(Z,D);*

*endmodule*

**testbench.sv:**

*`default\_nettype none*

*module dl\_lab5;*

*reg a, b, c, d,cin;*

*wire S0,S1,C0,C1;*

*lab5 parms(a,b,c,d,S0,S1,C0,C1);*

*initial*

*begin*

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

*$dumpvars(0, parms);*

*$display("Lab 5 Obj 1");*

*#1*

*a<=0;*

*b<=0;*

*c<=0;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=0;*

*c<=0;*

*d<=1;*

*#1*

*#1*

*a<=0;*

*b<=0;*

*c<=1;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=0;*

*c<=1;*

*d<=1;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=0;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=0;*

*d<=1;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=1;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=1;*

*d<=1;*

*#1*

*#1*

*a<=1;*

*b<=0;*

*c<=0;*

*d<=0;*

*#1*

*#1*

*a<=1;*

*b<=0;*

*c<=0;*

*d<=1;*

*#1*

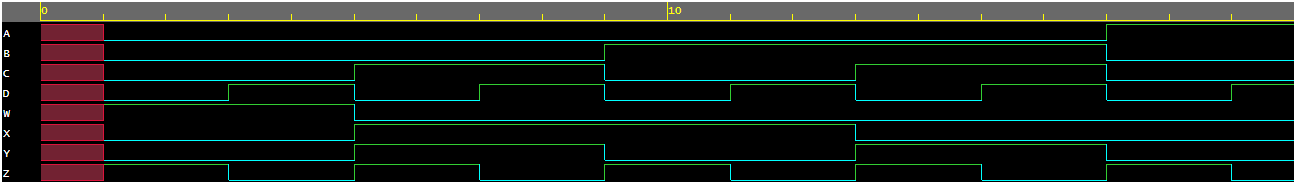
*$finish();*

*end*

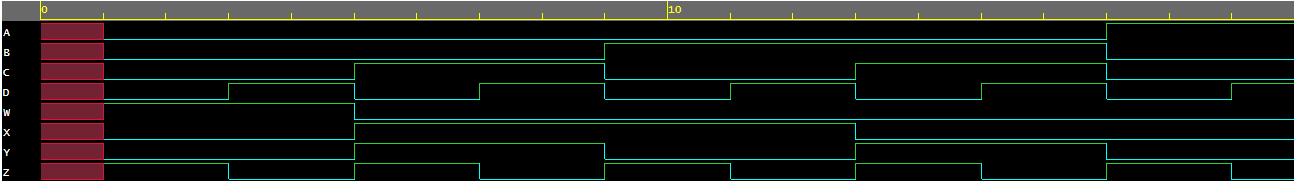
*endmodule*

**Links:** <https://www.edaplayground.com/x/BX38>

**EPWaveform:**



***Dataflow model***



***Gate-level model***

**Observation:**

The following Truth Table was obtained from the above EP Waveforms:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **W** | **X** | **Y** | **Z** |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X |

1. **Design a combinational circuit with four inputs and four outputs that converts a 4bit binary number into the equivalent 4bit Gray code.**

**HDL Program:**

**design.sv:**

*`default\_nettype none*

*module lab5 (*

*input B3,*

*input B2,*

*input B1,*

*input B0,*

*output G3,G2,G1,G0*

*);*

*//dataflow model*

*assign G3=B3;*

*assign G2=B2^B3;*

*assign G1=B1^B2;*

*assign G0=B0^B1;*

*// gate-level model*

*buf(G3,B3);*

*xor x1(G2,B2,B3),*

*x2(G1,B2,B1),*

*x3(G0,B1,B0);*

*endmodule*

**testbench.sv:**

*`default\_nettype none*

*module dl\_lab5;*

*reg a, b, c, d,cin;*

*wire S0,S1,C0,C1;*

*lab5 parms(a,b,c,d,S0,S1,C0,C1);*

*initial*

*begin*

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

*$dumpvars(0, parms);*

*$display("Lab 5 Obj 2");*

*#1*

*a<=0;*

*b<=0;*

*c<=0;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=0;*

*c<=0;*

*d<=1;*

*#1*

*#1*

*a<=0;*

*b<=0;*

*c<=1;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=0;*

*c<=1;*

*d<=1;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=0;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=0;*

*d<=1;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=1;*

*d<=0;*

*#1*

*#1*

*a<=0;*

*b<=1;*

*c<=1;*

*d<=1;*

*#1*

*#1*

*a<=1;*

*b<=0;*

*c<=0;*

*d<=0;*

*#1*

*#1*

*a<=1;*

*b<=0;*

*c<=0;*

*d<=1;*

*#1*

*#1*

*a<=1;*

*b<=0;*

*c<=1;*

*d<=0;*

*#1*

*#1*

*a<=1;*

*b<=0;*

*c<=1;*

*d<=1;*

*#1*

*#1*

*a<=1;*

*b<=1;*

*c<=0;*

*d<=0;*

*#1*

*#1*

*a<=1;*

*b<=1;*

*c<=0;*

*d<=1;*

*#1*

*#1*

*a<=1;*

*b<=1;*

*c<=1;*

*d<=0;*

*#1*

*#1*

*a<=1;*

*b<=1;*

*c<=1;*

*d<=1;*

*#1*

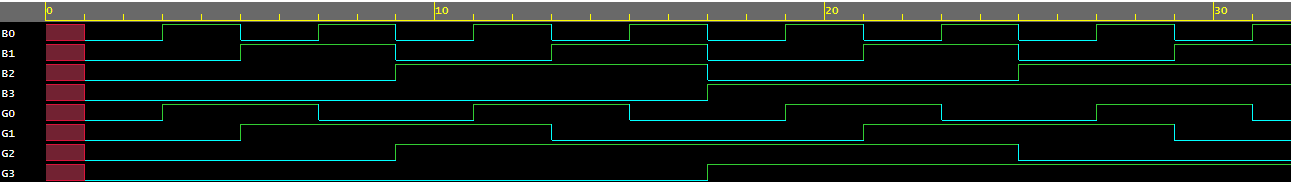
*$finish();*

*end*

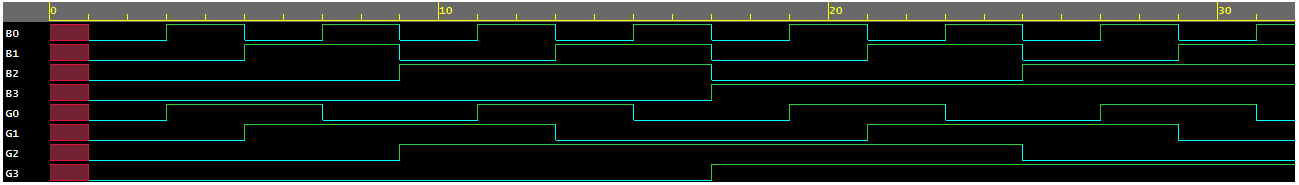
*endmodule*

**Links:** <https://www.edaplayground.com/x/7ner>

**EPWaveform:**



***Dataflow model***



***Gate-level model***

**Observation:**

The following Truth Table was obtained from the above EP Waveforms:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **B3** | **B2** | **B1** | **B0** | **G3** | **G2** | **G1** | **G0** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

1. **Design a combinational circuit that accepts a 2-bit number and generates an output binary number equal to the square of the input number.**

**HDL Program:**

**design.sv:**

*`default\_nettype none*

*module mod (*

*input A,*

*input B,*

*output W,X,Y,Z*

*);*

*//dataflow model*

*assign W=A&&B;*

*assign X=A&&~B;*

*assign Y=0;*

*assign Z=B;*

*//gate-level model*

*and a1(W,A,B),*

*a2(X,A,~B);*

*buf b1(Y,0),*

*b2(Z,B);*

*endmodule*

**testbench.sv:**

*`default\_nettype none*

*module tb\_mod;*

*reg a, b;*

*wire W,X,Y,Z;*

*mod h\_dut(a,b,W,X,Y,Z);*

*initial*

*begin*

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

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

*$display("Lab 5 Obj 3");*

*#1*

*a <= 0;*

*b <= 0;*

*#1*

*#1*

*a <= 0;*

*b <= 1;*

*#1*

*#1*

*a <= 1;*

*b <= 0;*

*#1*

*#1*

*a <= 1;*

*b <= 1;*

*#1*

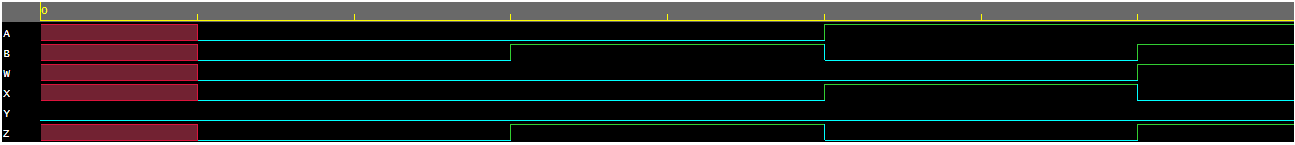
*$finish();*

*end*

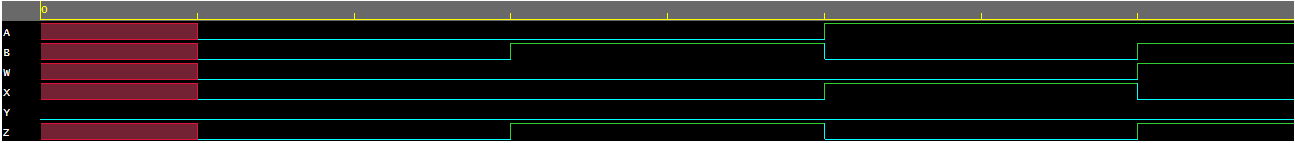
*endmodule*

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

**EPWaveform:**



***Dataflow model***



***Gate-level model***

**Observation:**

The following Truth Table was obtained from the above EP Waveforms:

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

**Conclusion:**

**Objective 1:**

It can be concluded that to 9’s complement of BCD input digit we need 3 and gates and not gates, 2 buffers and 1xor gate and circuit leads to the functions

**W = A’B’C**

**X = B ⊕ C**

**Y = C**

**Z = D’**

**Objective 2:**

It can be concluded the required function is generated by taking the XOR of consecutive binary bits and a total of 3 xor gates and a buffer is required to make the circuit.

**Objective 3:**

It can be concluded that circuit which accept 2-bit number and generates an output binary number equal to the square of the input numberleads to function

**W = AB**

**X = AB’**

**Y = 0**

**Z = B**

**IV. POST LAB:**

1. **Using the circuit you have designed to get 9's complement of 4 bit BCD input, find the 9's complement of (26)10 .**

**Ans: -** (26)10 in BCD=0010 0110

Complement for 0110 as per the circuit 0011

Complement for 0010 as per the circuit 0111

Final Expression of BCD is 0111 0011

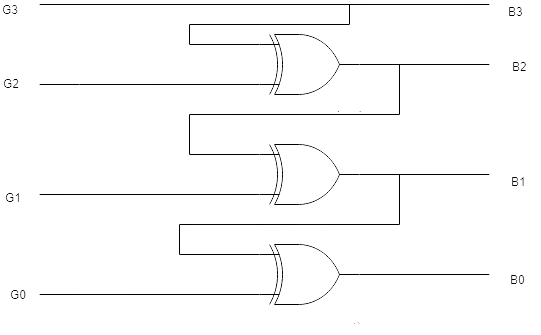
0111 0011 BCD to decimal (73) hence 9's complement of (26)10 is (73)10.

1. **What is the advantage of gray code?**

**Ans: -** The advantages of Gray Code are:

1. In Gray code, if we go from one decimal number to next, only one bit of the gray code changes. Because of this feature, an amount of switching is minimized and the reliability of the switching systems is improved.
2. Advantage of grey code over binary is only one-bit changes for each step. This will be useful in circuits that are sensitive to glitches.
3. **Draw the logic circuit that converts a 4-bit Gray code to binary code**.

**Ans:-**



**V. HDL PROGRAM LINK:**

**Objective 1:** [**https://www.edaplayground.com/x/BX38**](https://www.edaplayground.com/x/BX38)

**Objective 2:** [**https://www.edaplayground.com/x/7ner**](https://www.edaplayground.com/x/7ner)

**Objective 3:** [**https://www.edaplayground.com/x/RcMb**](https://www.edaplayground.com/x/RcMb)