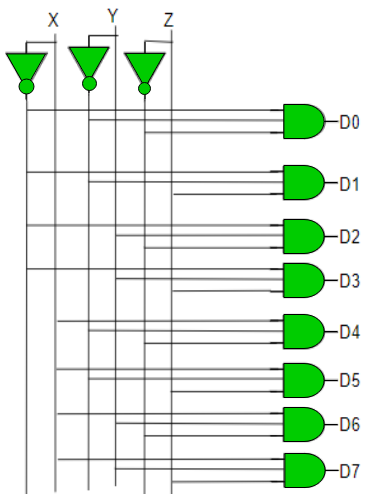
**Assignment 5**

Decoder: (3 to 8)

Truth Table

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | Y | Z | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Circuit Schematic Diagram



VERILOG CODE

module decoder(D,X,Y,Z);

output [7:0] D;

input X,Y,Z;

assign D[0] = ~X & ~Y & ~Z;

assign D[1] = ~X & ~Y & Z;

assign D[2] = ~X & Y & ~Z;

assign D[3] = ~X & Y & Z;

assign D[4] =  X & ~Y & ~Z;

assign D[5] =  X & ~Y & Z;

assign D[6] =  X & Y & ~Z;

assign D[7] =  X & Y & Z;

endmodule

**Problem Statement:**

1. **Write Verilog code for 4-bit ripple carry adder using full adder. Test using university wave form.**
2. **Write the Verilog code for a 4 bit carry-look-ahead Adder. Test using university wave form.**
3. **Write the Verilog code for a 3:8 Decoder. Test using university wave form.**
4. **Write the Verilog code for a BCD to 7 segment display Decoder. Test using university wave form.**