

**CSE 345: Digital Logic Design**

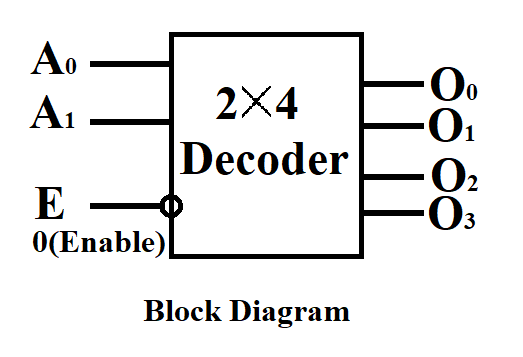
**Section: 1, Spring 2020**

**Project**

|  |
| --- |
|  |
| Submitted by:  **Group: 9**  Group Members:  AMIT ROY 2017-3-60-021  SIRAJUM MARIA MUNA 2017-3-60-020  SHEKHOR CHANDRA SAHA 2017-3-60-025  MD NABID HOSSAIN 2017-3-60-059 |
| Submitted to:  Musharrat Khan  Senior Lecturer, Dept. of CSE  East West University |
| Date of Submission:09-05-2020 |

**Question 1:**

Draw the block diagram, truth table, equation and logic diagram of a 2-to-4 lines decoder with active- LOW outputs and active-LOW enable input.

**Block diagram:**

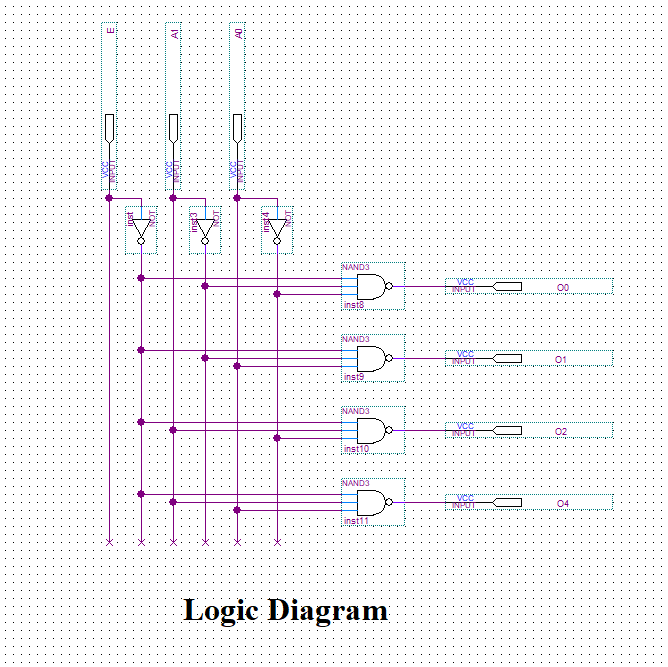
**Truth table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | Outputs | | | |
| E |  |  |  |  |  |  |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Equation:**

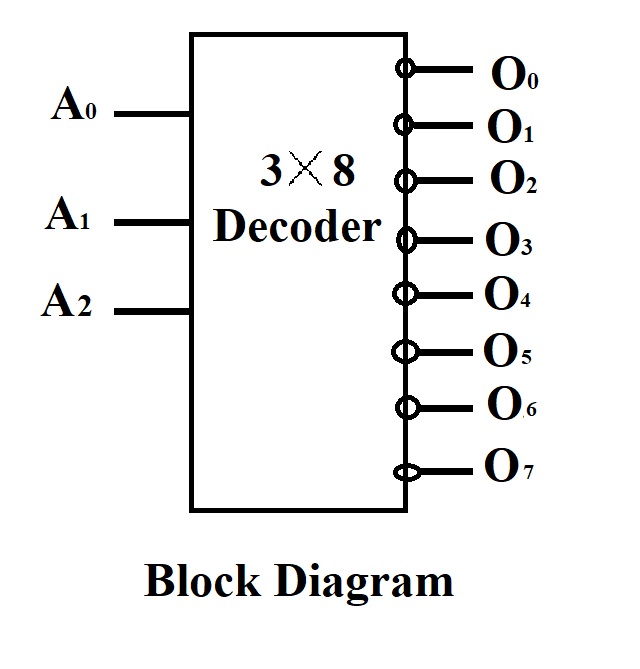
=E++= ( E ) =E++= ( E )

=E++= ( E ) =E++= ( E )

**Logic diagram:**

**Question 2:**

Draw the block diagram, truth table, equation and logic diagram of a 3-to-8-lines decoder with active-LOW outputs.

**Block diagram:**

**Truth table:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | | | outputs | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Equation:**

=++= ( )

=++= ( )

=++= ( )

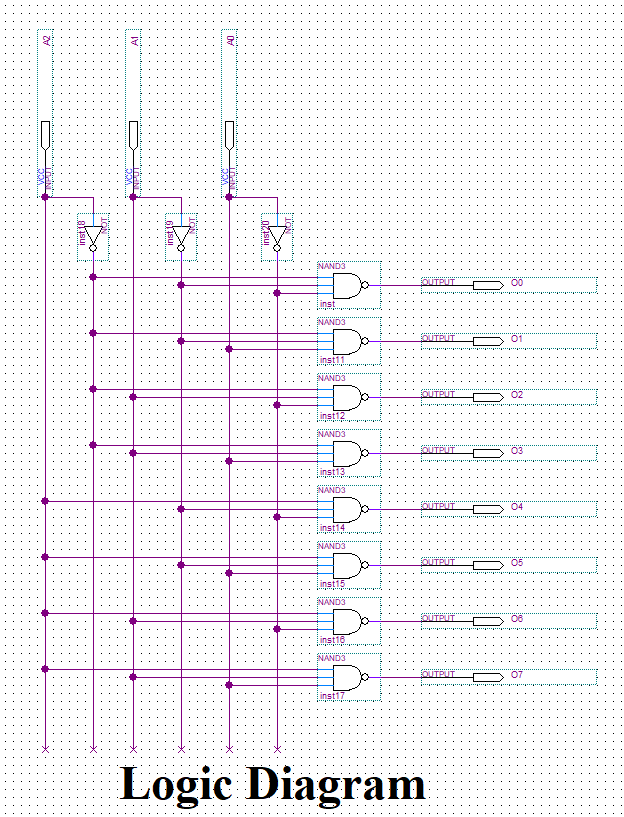
=++= ( )

=++= ( )

=++= ( )

=++= ( )

=++= ( )

**Logic Diagram:**

**Question 3:**

Write Structural Verilog code for logic diagram for 2-to-4-lines decoder with active-LOW outputs and active-LOW enable input and simulate it using Quartus II software.

**Structural Verilog code:**

module ques3(input E,A1,A0,output o0,o1,o2,o3);

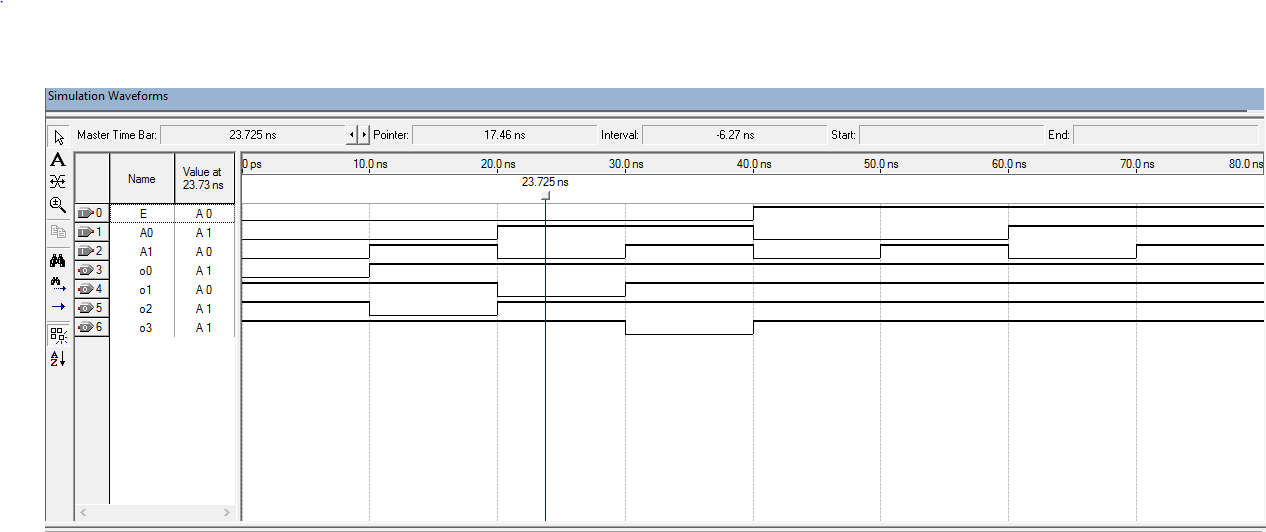
nand (o0,~E,~A1,~A0),

(o1,~E,~A1,A0),

(o2,~E,A1,~A0),

(o3,~E,A1,A0);

Endmodule

**Simulation:**

**Question 4:**

Write Behavorial Verilog code for 3-to-8-lines decoder with active-LOW outputs and simulate it using Quartus II software.

**Behavorial verilog code:**

**Procedural model:**

Module3by8decoder( input A2, A1, A0, output reg o0, o1, o2, o3, o4, o5, o6, o7);

always@( A2, A1,A0) begin

case( {A2, A1, A0})

3b 000 : o0=0;

3b 001 :o1=0;

3b 010: o2=0;

3b 011: o3=0;

3b 100: o4=0;

3b 101: o5=0;

3b 110: 06=0;

3b 111: o7=0;

End

Endcase

Continuous assign statement:

module 3by8decoderassign( input A2, A1, A0, output o0, o1, o2, o3, o4, o5, o6, o7);

assign o0=( ˜A2 & ˜A1 & ˜A0);

assign o1=( ˜A2 & ˜A1 & A0);

assign o2=( ˜A2 & A1 & ˜A0);

assign o3=( ˜A2 & A1 & A0);

assign o4=( A2 & ˜A1 & ˜A0);

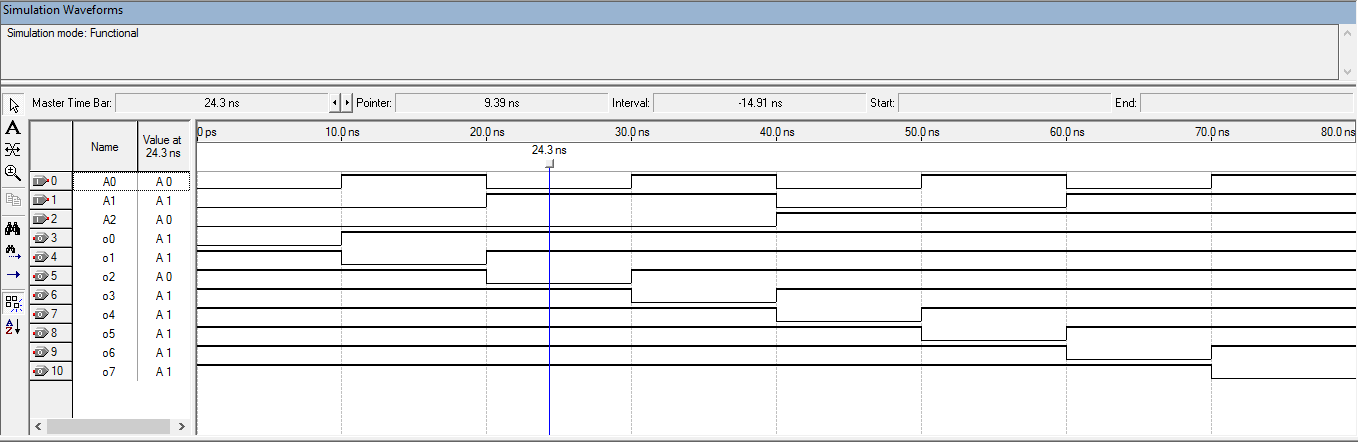
assign o5=( A2 & ˜A1 & A0);

assign o6=( A2 & A1 &˜ A0);

assign o7=( A2 & A1 & A0);

endmodule

**Simulation:**

****