**Department of Electrical Engineering**

**Faculty Member:** Arshad Nazir **Dated:** 11/12/2021

**Semester:** 3rd **Section:** BEE 12C

**Group No.: 7**

**EE-221: Digital Logic Design**

Lab 11: Voting Machine

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4/CLO4** | **PLO4/CLO4** | **PLO5/CLO5** | **PLO8/CLO6** | **PLO9/CLO7** |  |
| **Name** | **Reg. No** | **Viva / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Teamwork** | **Total marks Obtained** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **25 Marks** |
| **Muhammad Adil Azeem** | **342459** |  |  |  |  |  |  |
| **Muhammad Umer** | **345834** |  |  |  |  |  |  |
| **Raahim Ahmad Waqar** | **342287** |  |  |  |  |  |  |
| **Saad Bakhtiar** | **341150** |  |  |  |  |  |  |
| **Syed Aun Ali Kazmi** | **342384** |  |  |  |  |  |  |

## Lab 11: Voting Machine

This Lab experiment has been designed to familiarize the students with use of multiplexers to implement a Boolean function and use BCD-to-Seven-Segment Decoder to drive the 7-segment Display. This lab requires some knowledge of SSI/MSI combinational circuits like Multiplexers, decoders, and Numeric Read-out Display.

**Objectives:**

* Understand the function of Multiplexers and their uses in implementing a given Boolean function.
* Familiarization with BCD-to-Seven-Segment Decoder IC as a driver to drive Numeric Read-out.
* Transform any problem statement to truth table description, and choose output functions that need Multiplexers implementation or other simplification techniques using logic gates.
* Design and verify combinational circuit design

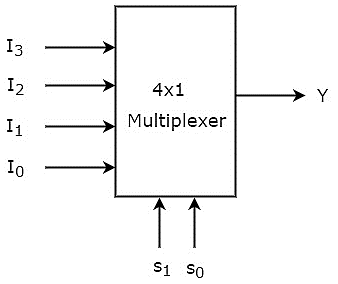
**Lab Instructions**

* This lab activity comprises three parts, namely Pre-lab, Lab tasks, and Post-Lab Viva session.
* The lab report will be uploaded on LMS three days before scheduled lab date. The students will get hard copy of lab report, complete the Pre-lab task before coming to the lab and deposit it with teacher/lab engineer for necessary evaluation. Alternately each group to upload completed lab report on LMS for grading.
* The students will start lab task and demonstrate design steps separately for stepwise evaluation (course instructor/lab engineer will sign each step after ascertaining functional verification)
* Remember that a neat logic diagram with pins numbered coupled with nicely patched circuit will simplify trouble-shooting process.
* After the lab, students are expected to unwire the circuit and deposit back components before leaving.
* The students will complete lab task and submit complete report to Lab Engineer before leaving lab.
* **The Total duration for the lab is 3 hrs. After lab duration, a deduction of 5 marks per day will be done for late submission.**
* **A lab with in-complete lab tasks will not be accepted.**
* There are related questions at the end of this activity. Give complete answers.

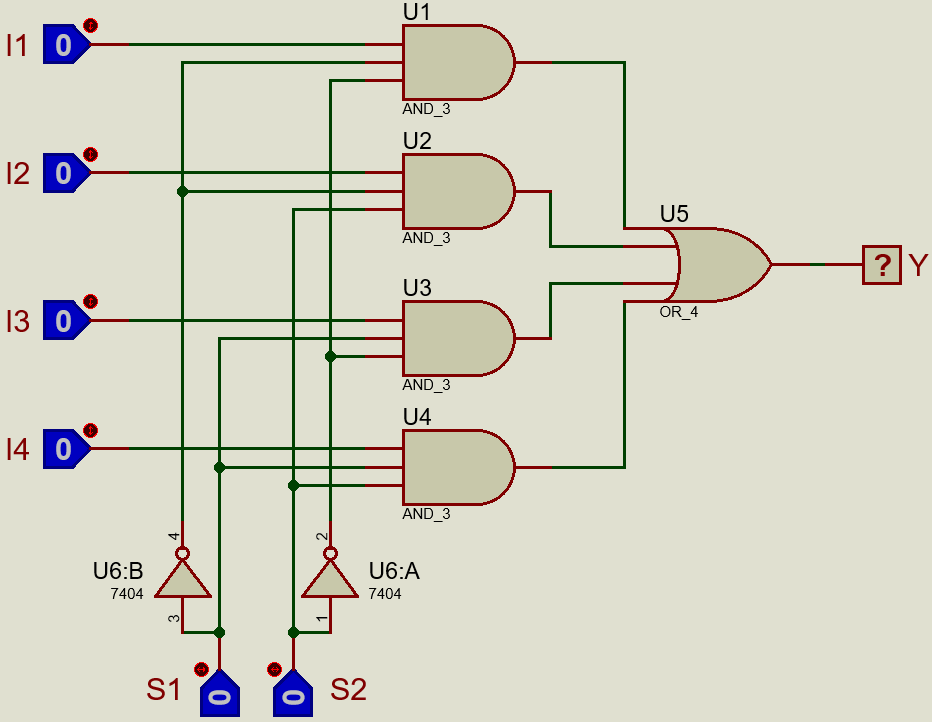
**Pre – Lab Tasks**

**Pre – Lab Task 1**

**Draw the Logic Diagram for a 4 to 1 MUX and also give its Truth Table and Block Diagram Representation.**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input Lines** | | | | **Selection Lines** | | **Output** |
| **X3** | **X2** | **X1** | **X0** | **S1** | **S0** | **Y** |
| X | X | X | **I0** | **0** | **0** | **I0** |
| X | X | **I1** | X | **0** | **1** | **I1** |
| X | **I2** | X | X | **1** | **0** | **I2** |
| **I3** | X | X | X | **1** | **1** | **I3** |



**Pre – Lab Task 2**

**For implementing a Boolean function of n variables, a MUX with how many selection lines and inputs are needed.**

For implementing a function of n – variables, we require **n – 1** selection lines and **2(n – 1)** input lines.

*Note that the aforementioned general formula is providing us with the efficient implementation of the function. The same function could be accomplished using n – selection lines and 2n input lines.*

**Pre – Lab Task 3**

**For implementing the Function F(x,y,z) = ∑(1,2,6,7), a MUX with how many selection lines is needed?**

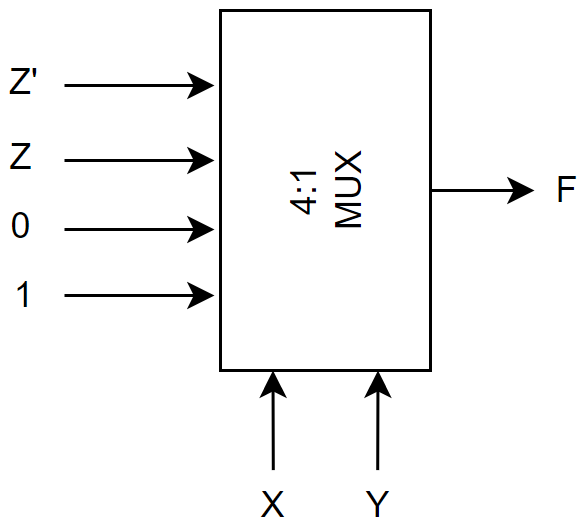
We can treat the last variable, z, as data line and then we will only require two selection lines to implement the function with a 4:1 MUX.

**Pre – Lab Task 4**

**Implement the above function (Draw its Logic Diagram) F with the MUX suggested in answer to above question?**

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **Z** | **F** |
| 0 | 0 | 0 | **0** |
| 0 | 0 | 0 | **1** |
| 0 | 0 | 1 | **1** |
| 0 | 0 | 1 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 0 | **0** |
| 0 | 1 | 1 | **1** |
| 0 | 1 | 1 | **1** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input Lines** | | | |
|  | **I0** | **I1** | **I2** | **I3** |
| **Z’** | 0 | **2** | 4 | **6** |
| **Z** | **1** | 3 | 5 | **7** |
| **=>** | **Z’** | **Z** | **0** | **1** |



**Lab Tasks**

**Voting Machine Problem**

A small corporation has 9 shares of stock. The corporation board consists of 4 members (A Chairman, 2 senior members and 1 junior member).

Each member of the Corporation Board has a voting right at a Corporation Financial meeting. The committee has agreed to assign the following share(s) to vote of each member.

|  |  |
| --- | --- |
| **Chairman** | **4 Shares** |
| **Senior Member** | **2 Shares** |
| **Senior Member** | **2 Shares** |
| **Junior Member** | **1 Shares** |

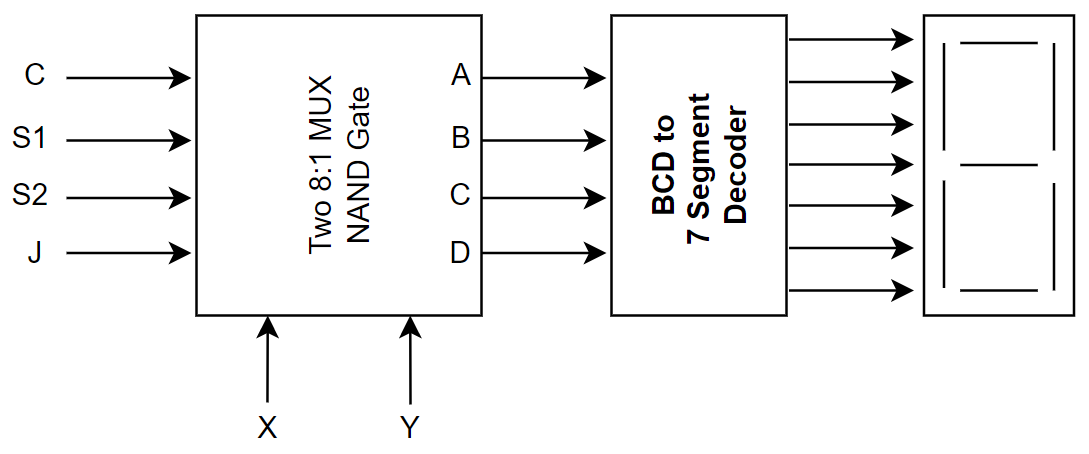
The Committee wants to automate the voting process and get a voting machine that gets input from each member and then display the total number of shares.

Each of these persons has a switch to close when voting yes (Logic 1) and to open when voting No (Logic 0) for his share.

You are a Design Engineer in the same corporation and have been asked to design such a voting machine. The only ICs available to you are two 8x1 multiplexers a 3-input NAND gates and a 7-Segment Decoder IC and a 7segemt LED Display.

**Task 1**

**Draw a simplified Block Diagram of such a voting machine (You do not need to make exact Logic Diagram).**



**Task 2**

**There are four inputs to this machine (one for each member) and the output is a binary number (4-bit). Draw the truth table for the Output Function.**

Following truth table demonstrates the functioning of this block diagram and also lists the output, a 4 – bit binary number for all possible combinations of the inputs.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | | **Output** | | | |
| **C** | **S1** | **S2** | **J** | **A** | **B** | **C** | **D** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

**Task 3**

**You are provided with two 8 – to – 1 MUXs, so how many functions can you implement with them?**

The maximum functions you can implement with two 8:1 line MUXs are the same as that the output lines they possess. Hence, two functions can be implemented through them.

**Task 4**

**Which of the above functions would you implement with MUX? Specify and also give their Logic Diagram using MUX.**

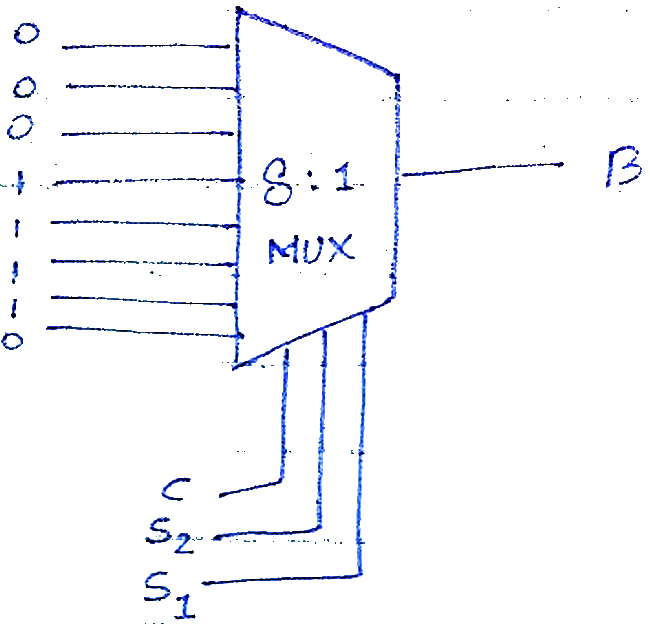
From observation of the above truth table, we can deduce that there is no need to implement the function D, as it is the same as the input variable J.

Also, A having only two HIGH – states can be implemented through the use of a 3 – input NAND gate

For B and C, we can treat other variables as selection inputs other than the ones we intend to use and implement them through an 8:1 MUX. Following implementation table and the block diagram demonstrate this:

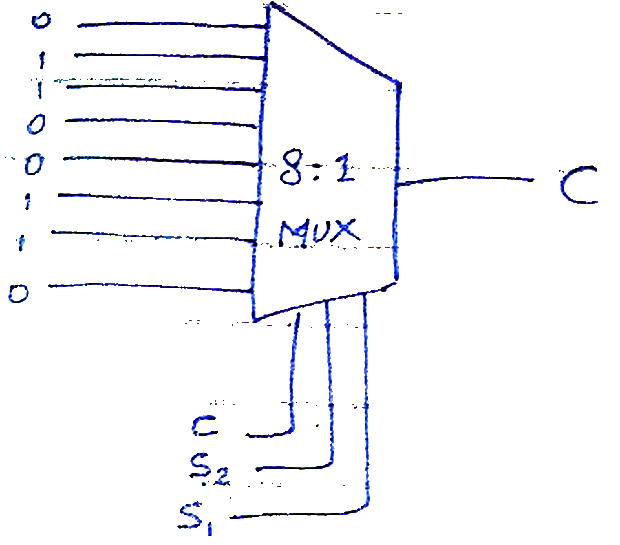
**For B:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Lines** | | | | | | | |
|  | **I0** | **I1** | **I2** | **I3** | **I4** | **I5** | **I6** | **I7** |
| **J’** | **0** | **2** | **4** | **6** | **8** | **10** | **12** | **14** |
| **J** | **1** | **3** | **5** | **7** | **9** | **11** | **13** | **15** |
| **=>** | **0** | **0** | **0** | **1** | **1** | **1** | **1** | **0** |



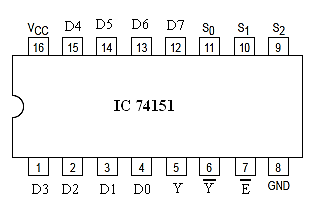
**For C:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Input Lines** | | | | | | | |
|  | **I0** | **I1** | **I2** | **I3** | **I4** | **I5** | **I6** | **I7** |
| **J’** | **0** | **2** | **4** | **6** | **8** | **10** | **12** | **14** |
| **J** | **1** | **3** | **5** | **7** | **9** | **11** | **13** | **15** |
| **=>** | **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** |

****

**Task 5**

**Give the PIN Configuration of the 8-to-1 MUX provided to you in the lab.**

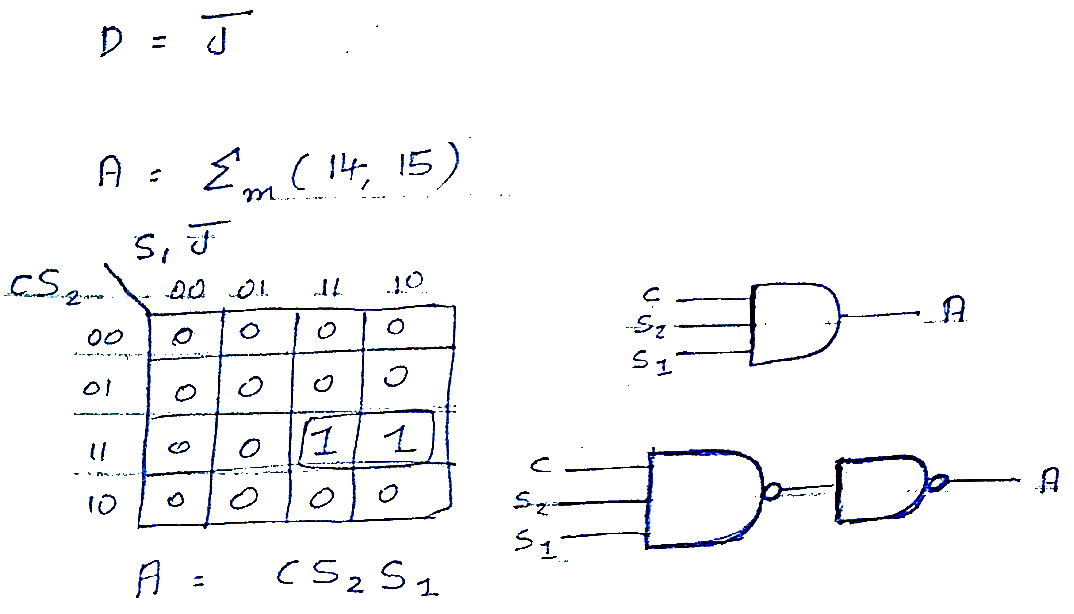


Where:

* D0 – D7 are Data Inputs
* S0 – S2 are Selection Inputs
* Y and Y’ are Outputs
* E’ is Enable Input

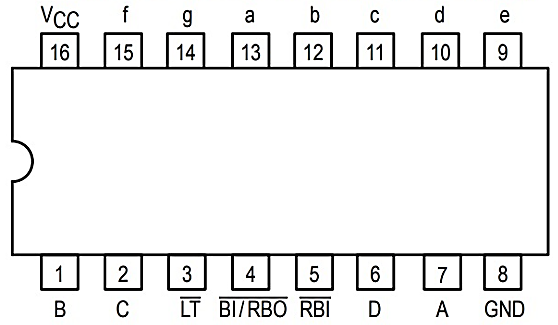
**Task 6**

**Give the Boolean expressions for the remaining two functions and also give their Logic Diagram (You can only use 3-input NAND Gate).**



**Task 7**

**The 4 outputs ABCD are in binary (BCD), so we will have to use a BCD to 7-segment Decoder to drive the 7-segment LED Display. Look for the BCD to 7-segment Decoder IC in the lab and give its number and PIN Configuration. Look up on internet for the data sheet of this IC.**



Where:

* LT is Lamp Test – To find if the seven segment decoder works properly or not.

**Task 8**

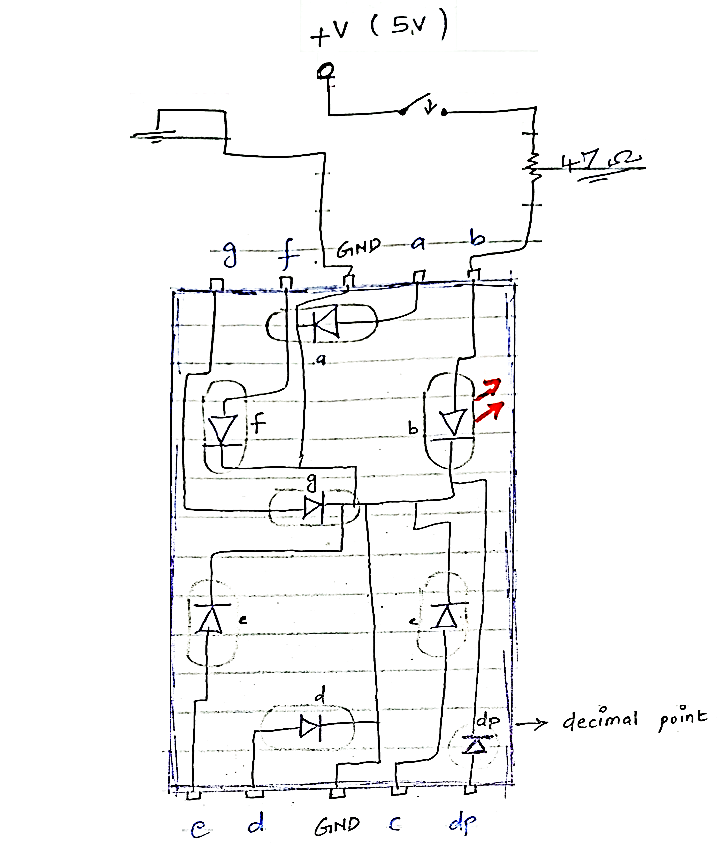
**Draw the truth table for the above IC.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Inputs** | | | | **Outputs** | | | | | | |
| **A** | **B** | **C** | **D** | **a** | **b** | **c** | **d** | **e** | **f** | **g** |
| 0 | 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | **1** | **0** |
| 0 | 0 | 0 | 1 | **0** | **1** | **1** | **0** | **0** | **0** | **0** |
| 0 | 0 | 1 | 0 | **1** | **1** | **0** | **1** | **1** | **0** | **1** |
| 0 | 0 | 1 | 1 | **1** | **1** | **1** | **1** | **0** | **0** | **1** |
| 0 | 1 | 0 | 0 | **0** | **1** | **1** | **0** | **0** | **1** | **1** |
| 0 | 1 | 0 | 1 | **1** | **0** | **1** | **1** | **0** | **1** | **1** |
| 0 | 1 | 1 | 0 | **1** | **0** | **1** | **1** | **1** | **1** | **1** |
| 0 | 1 | 1 | 1 | **1** | **1** | **1** | **0** | **0** | **0** | **0** |
| 1 | 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| 1 | 0 | 0 | 1 | **1** | **1** | **1** | **1** | **0** | **1** | **1** |
| 1 | 0 | 1 | 0 | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| 1 | 0 | 1 | 1 | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| 1 | 1 | 0 | 0 | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| 1 | 1 | 0 | 1 | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| 1 | 1 | 1 | 0 | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| 1 | 1 | 1 | 1 | **X** | **X** | **X** | **X** | **X** | **X** | **X** |

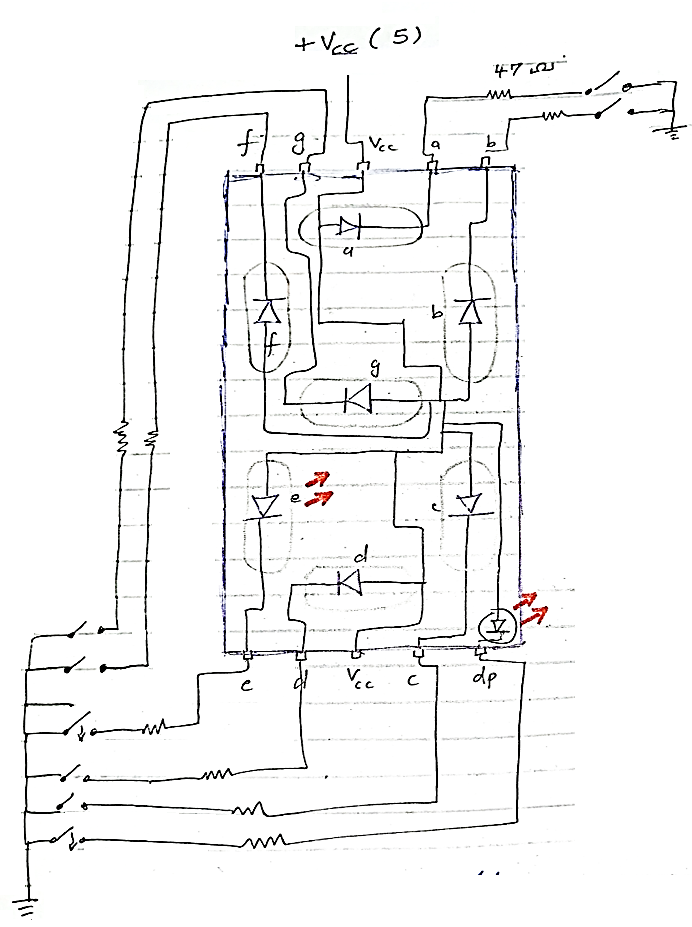
**Task 9**

**After selecting the decoder IC you need a 7-Segment LED Display. It comes in two configurations Common Anode and Common Cathode. Differentiate between them by drawing their Circuit Configuration.**

**Common Cathode Configuration**



**Common Anode Configuration**



*The mahor difference between these two configurations is the signal at which the 7 – Segment LEDs illuminate; They illuminate upon applying a HIGH signal (1) for Common Cathode configuration whereas they illuminate upon applying a LOW signal (0) for Common Anode configuration.*

**Task 10**

**What configuration have you decided to choose and why?**

We chose to go with the Common Cathode configuration of the 7-Segment Led Display as we use the 74LS48 IC which is used to drive CC segment circuits.

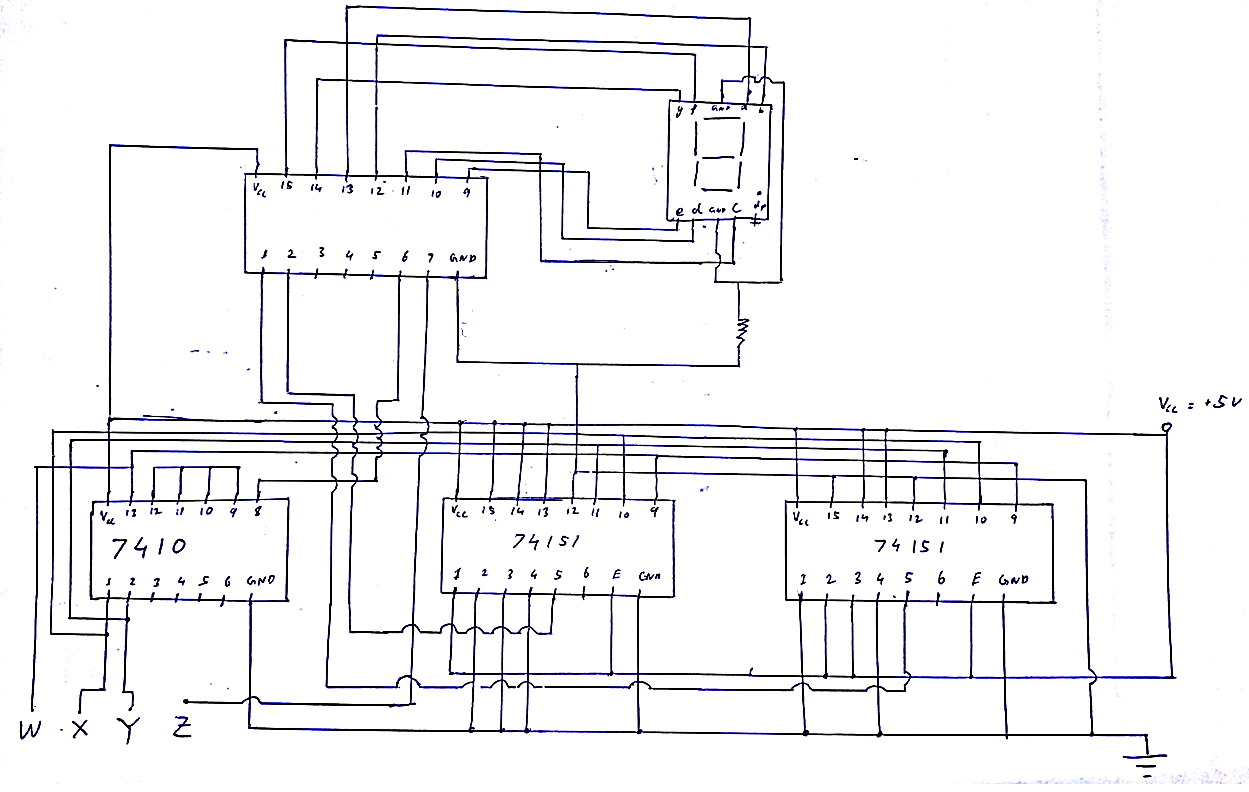
**Task 11**

**If you were to choose the other configuration of 7-Segment Display what change would you need in your circuit?**

For the other configuration, we could have either used the 74LS77 IC or to modify the current circuit without changing or swapping out any parts, we could have used the CA configuration which would in – turn complement our truth table since our LEDs now light up at LOW logic level.

**Task 12**

**Show your complete circuit in working condition after completing the Logic Diagram for the whole voting Machine System mentioning the ICs name and Configuration and implement it in hardware.**



**Proteus Simulation**

