**DESIGN PROCEDURE FOR LOGIC CIRCUIT**

Applying logic gates and simplification in circuit design problems, we have to follow this design procedure.

Design Procedure:

The design of combinational circuits starts from the verbal outline of the problem and ends in a logic circuit diagram or a set of Boolean functions from which the logic diagram can be easily obtained. The procedure involves the following steps:

1. The problem is stated.
2. The number of available input variables and required output variables is determined.
3. The input and output variables are assigned letter symbols.
4. The truth table that defines the required relationships between inputs and outputs is derived.
5. The simplified Boolean function for each output is obtained.
6. The logic diagram is drawn.

Example 1:

Step 1: Design a combinational circuit with three inputs and one output. The output is equal to logic-l when the binary value of the input is less than 3. The output is logic-0 otherwise.

Step 2:

Number of inputs = 3

Number of output = 1

Step 3:

Number of inputs = 3 (x, y, z)

Number of output = 1 (A)

Step 4:

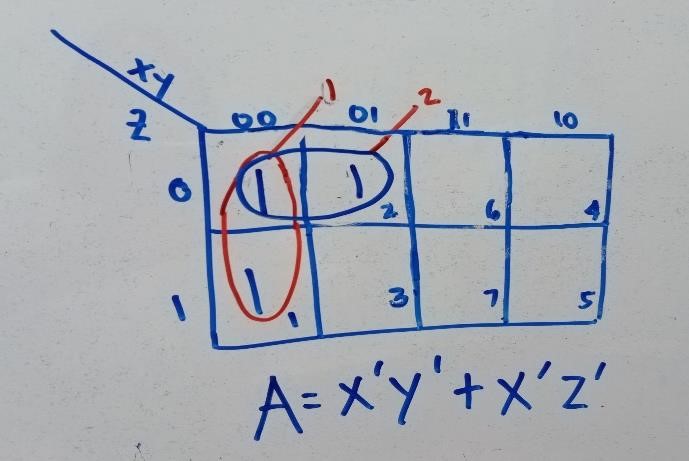
Truth Table:

Number of combinations = 2n = 23 = 8 (0 to 7)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | z |  | A |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 2 | 1 |
| 0 | 1 | 1 | 3 | 0 |
| 1 | 0 | 0 | 4 | 0 |
| 1 | 0 | 1 | 5 | 0 |
| 1 | 1 | 0 | 6 | 0 |
| 1 | 1 | 1 | 7 | 0 |

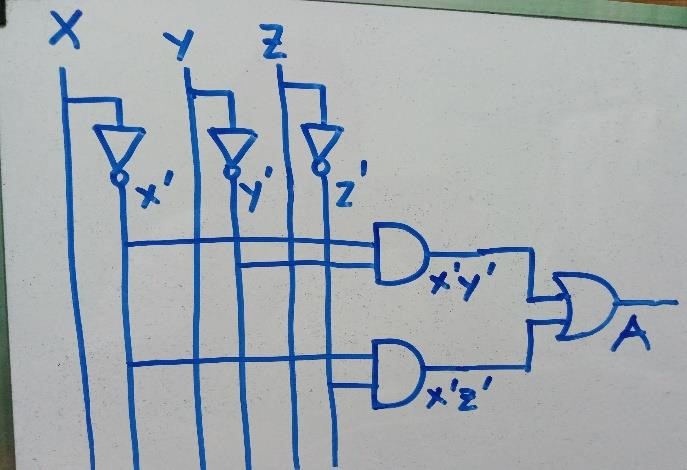
Step 5:

**Boolean Function For** A:

 To solve this K-map problem, we use a three-variable map because there are three inputs and one output. From the truth table, the minterms (output = 1) occur at locations 0, 1, and 2. These positions are marked with 1s on the K-map. By grouping adjacent 1s, we derive the simplified Boolean expression and trace its derivation step by step to ensure accuracy.

Step 6:

Logic Diagram:

To create the logic diagram, use basic gates: NOT, AND (2-input), and OR (2-input). Start with the input variables xx, yy, and zz, and apply NOT gates to generate their complements (x′x', y′y', z′z'). Use AND gates to form the product terms for the simplified Boolean expression. Finally, combine these product terms using OR gates to complete the diagram.

To construct the logic diagram:

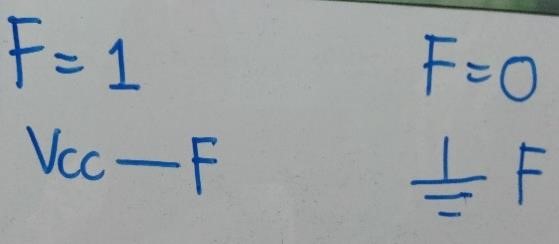
1. Start with the normal input variables (x,y,zx, y, z).
2. Use NOT gates to create their negated forms (x′,y′,z′x', y', z').
3. Form product terms using AND gates for multiplication.
4. Combine all product terms with an OR gate to generate the final output.

.

To create the logic diagram:

1. **Gate Functions**:
   * Use NOT gates for negation.
   * Use AND gates for multiplication.
   * Use OR gates for addition.
2. **Term Formation**:
   * Tap along input lines as needed to form product terms with AND gates.
   * Combine these terms with OR gates.
3. **Special Cases**:
   * If a variable has a final value of 1 or 0, include these directly in the diagram at the final stage.
   * Use additional AND gates as required to handle more complex terms.

.

**Vcc stands for Voltage at Common Collector**.

You use this symbol if the variable final expression is equal to 1. Say when F=1.

If the expression results in a 0, say when F=0, you use the ground symbol.

Example 2:

Design a combinational circuit with three inputs, *x, y,* and z, and three outputs, *A, B,* and C. When the binary input is 0, 1,2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is one less than the input.

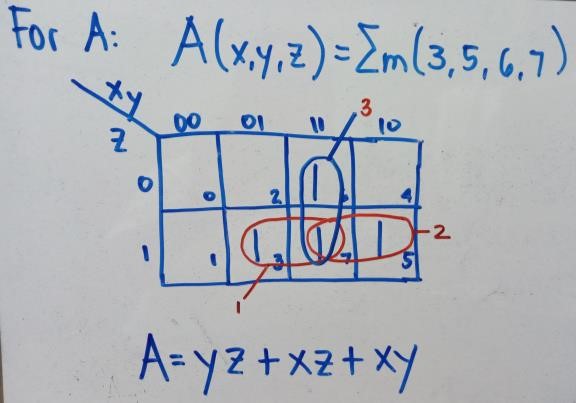
Number of inputs = 3 (x, y, z)

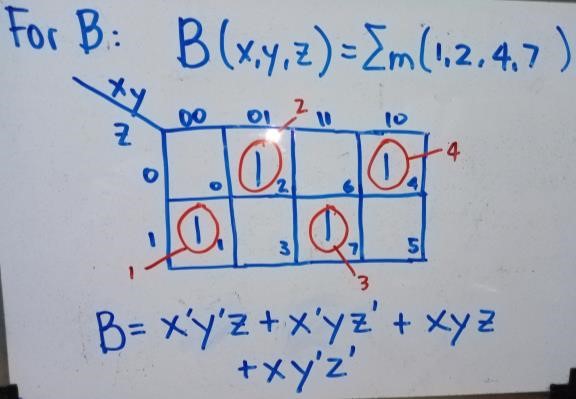
Number of output = 1 (A, B, C)

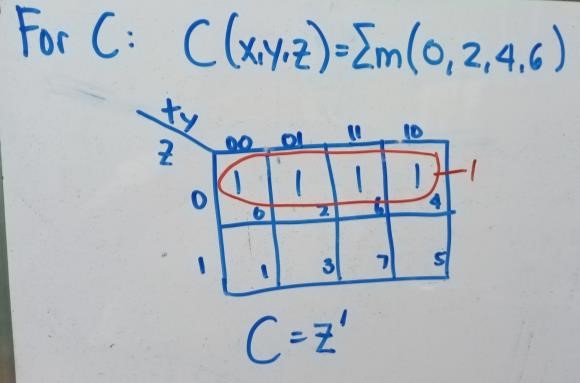
Truth Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | y | z |  | A | B | C |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 2 | 0 | 1 | 1 |
| 0 | 1 | 1 | 3 | 1 | 0 | 0 |
| 1 | 0 | 0 | 4 | 0 | 1 | 1 |
| 1 | 0 | 1 | 5 | 1 | 0 | 0 |
| 1 | 1 | 0 | 6 | 1 | 0 | 1 |
| 1 | 1 | 1 | 7 | 1 | 1 | 0 |

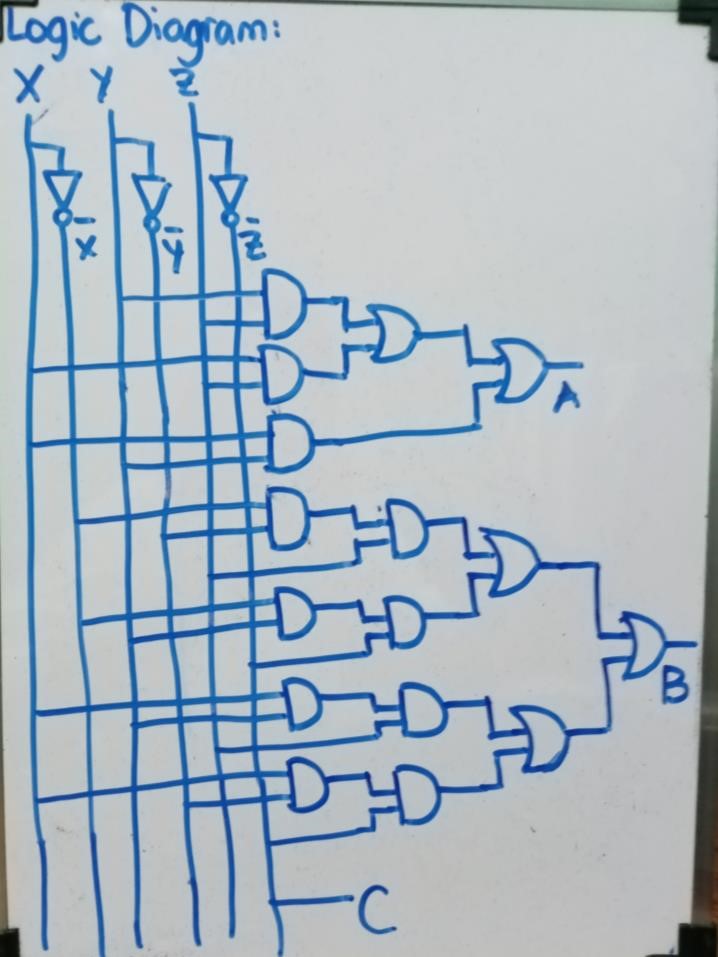
Boolean Functions:







Logic Diagram:



Assessment: Follow the design procedure for the solution. You may do this by pair or individually. Original by pair or individual as submitted on the first week of class. Solutions should be handwritten and pictures should be taken clearly. Compile all pictures in a single pdf file. Names should be written on each page. One pdf file per problem.

1. A majority function is generated in a combinational circuit when the output is equal to 1 if the input variables have more 1 's than 0's. The output is 0 otherwise. Design a 3-input majority function.

1. Design a combinational circuit that adds 2 two-bits given four bits S0S1S2S3 for the inputs. The sum is S0S1 +S2S3. Example if S0S1 =10 and S2S3=01, then the sum is 10 + 01 =11. Remember that there could be a carry bit for sum values. Be sure to include the carry as one of the variables.

1. A committee of three individuals decides issues for an organization. Each individual votes either yes or no for each proposal that arises. A proposal is passed if it receives at least two yes votes. Design a circuit that determines whether a proposal passes.