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.

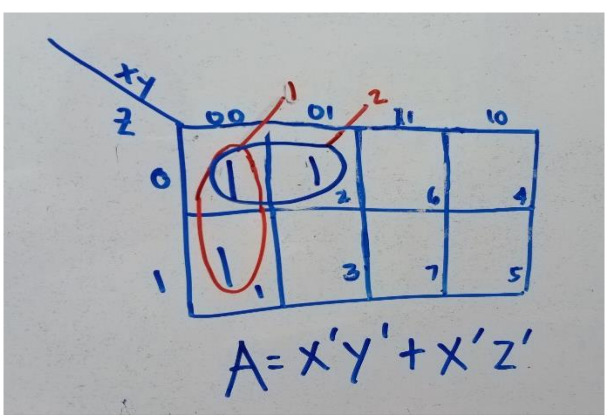
Step 2:
Number of inputs = 3
Number of output = 1

Step 3:
Number of inputs = 3 (x, y, z)
Number of output = 1 (A)

Step 4:
Truth Table:
Number of combinations = $2^n = 2^3 = 8$ (0 to 7)

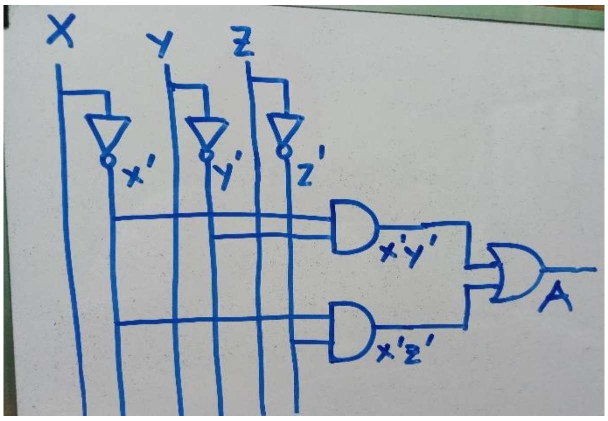
x	y	z		A
0	0	0	0	1
0	0	1	1	1
0	1	0	2	1
0	1	1	3	0
1	0	0	4	0
1	0	1	5	0
1	1	0	6	0
1	1	1	7	0

Step 5:
Boolean Function
For A:



For this k-map, we need a three variable map since the problem requires three inputs. We only need one map because there is only one output required in the problem. So tracing it from the truth table there are minterms (1) at location 0, 1 and 2. Hence, we place 1s on these locations and come up with the simplified Boolean expression. Place trace through why the Boolean expression was derived.

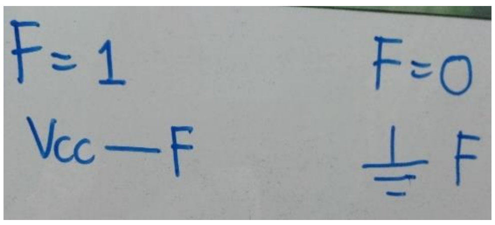
Step 6:
Logic Diagram:



For the logic diagram, we will only use the basic gates NOT, AND (2 inputs) and OR (2 inputs) gates in forming the diagram.
First note that what should be reflected are the normal values for the input variables such as in this case x, y, and z. Then beside each variable line, use a NOT gate to get the negated value. Take note of the x', y, and z' lines.
Then form each product term. Use the AND gate for the multiplication process. After forming all the product terms, add the product terms using the OR gate.

Note that the NOT is for negation, AND for multiplication and OR for addiction. Tap along the line which are needed to form terms. Note also that we have to use two AND gates to produce a single product term (as shown in the diagram for the next example).

In special cases, where a certain variable will have a final value of 1 or 0 in the Boolean expression, you should use this symbols in the diagram, just add them at the last part.



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:

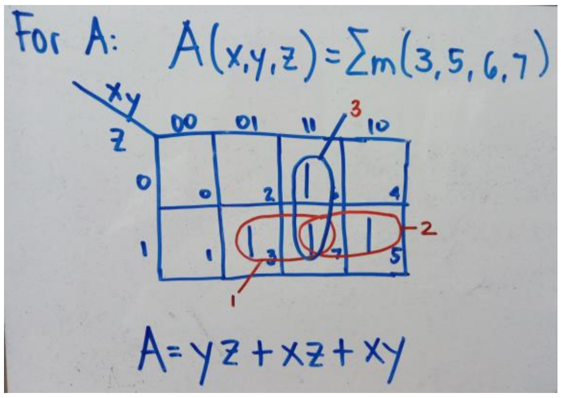
Design a combinational circuit with three inputs, *x, y,* and z, and three outputs, *A, B,* and C. When the binary input is 0, 1,2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is one less than the input.

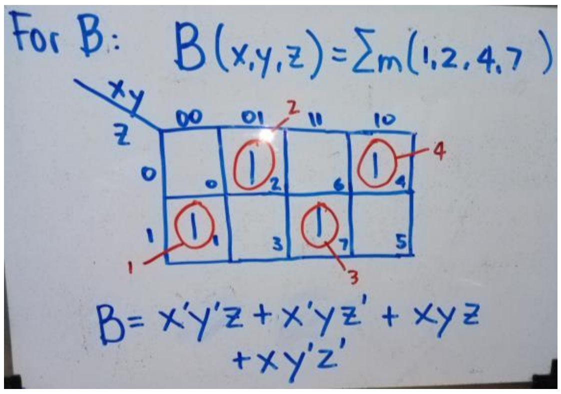
Number of inputs = 3 (x, y, z)
Number of output = 1 (A, B, C)

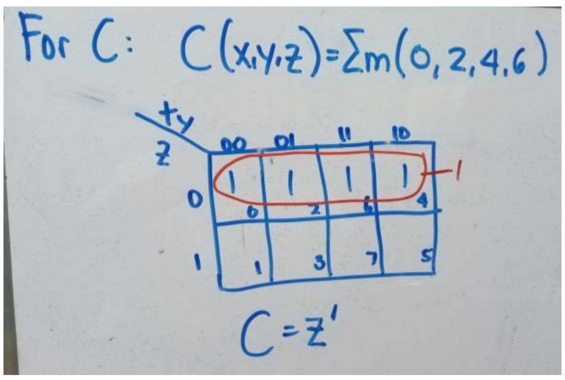
Truth Table:

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

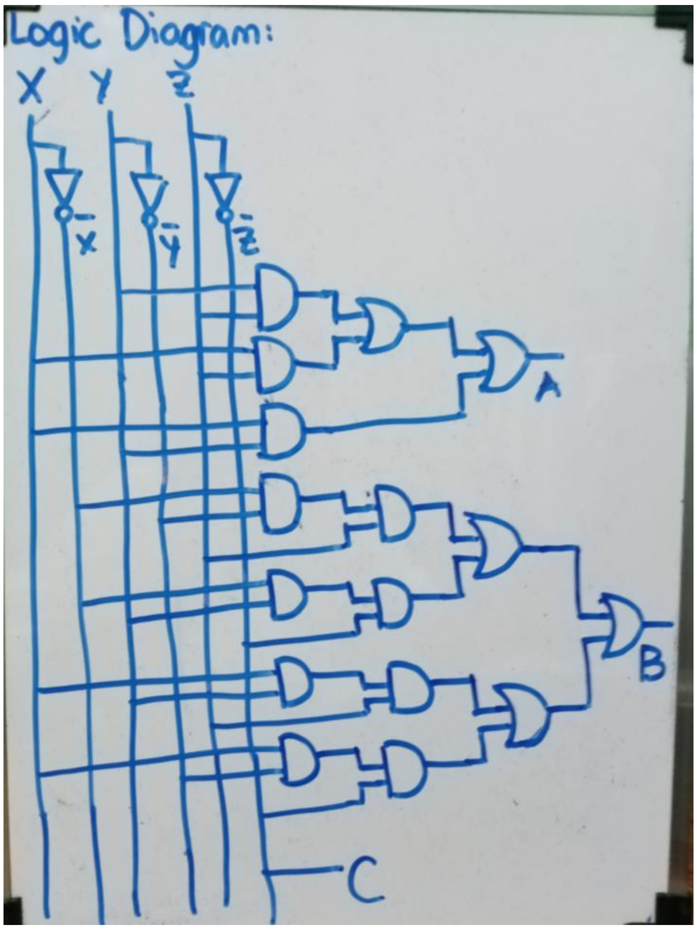
Boolean Functions:







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.

2. Design a combinational circuit that adds 2 two-bits given four bits $S_0S_1S_2S_3$ for the inputs. The sum is S_0S_1 $_+$ S_2S_3. Example if S_0S_1 =10 and S_2S_3=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.

3. 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.