Handout Combinational Circuits

Design Process:

i. Understand the problem statement
ii. Determine the number of inputs and outputs
iii. Derive the Truth Table that describes the behavior of digital circuit
iv. From the Truth Table derive Boolean Function in Sum of Min-terms form
v. Simplify Boolean Function for each input
vi. Draw the logic circuit
vii. Write down HDL for the chip and load it in h/w simulator to verify the logic

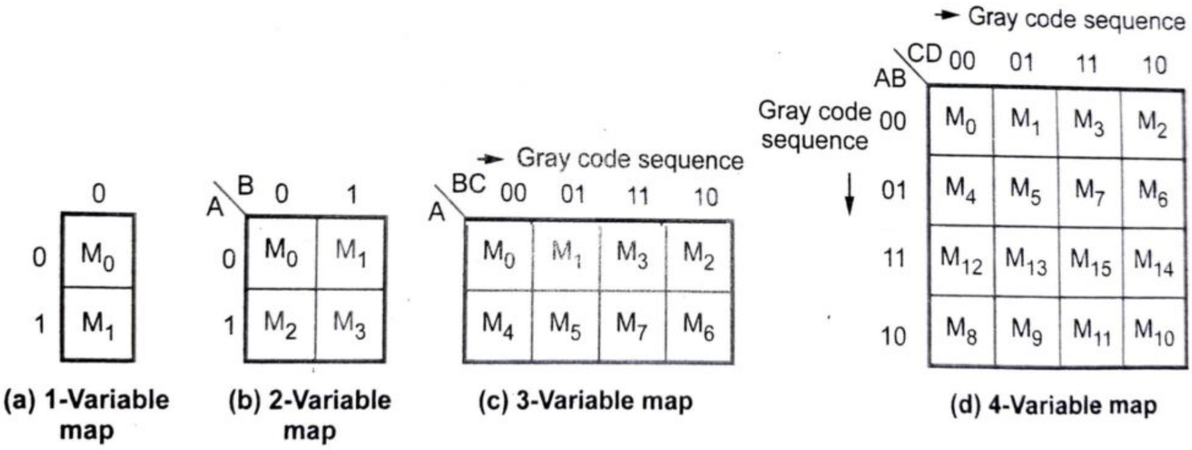
Example: Design a combinational circuit that takes a four-bit input and determines whether the input binary number is a prime number. In case if the number represents a prime number, then logic 1 is generated else logic 0 is generated. Draw the logic circuit, write down the HDL code and verify by loading the chip in H/W Simulator. (No need to simplify)

Techniques for Boolean Function Simplification:

- Using Boolean Laws
- Using K-Map
- Using Tabulation Method (Quine McClusky method)

Karnaugh Map

- K-Map is a technique to simplify a Boolean Function given in Sum of Min-terms form and is a graphical representation of a Truth Table
- There are $\mathbf{2^n}$ cells in a K-Map where **n** is the number of Boolean variables and each cell corresponds to one min-term or max-term.
- K-Map uses Grey Code, which is an ordering of binary number system, where two successive values differ in only one bit (0, 1, 3, 2, 6, 7, 5, 4, 12, 13, 15, 14, 10, 11, 9, 8)
- A K-Map can be of two, three, four variables. K-Map of larger than four variables are a little bit difficult to visualize, therefore, for Boolean function larger than 4 variables we use Quine McClusky Method.

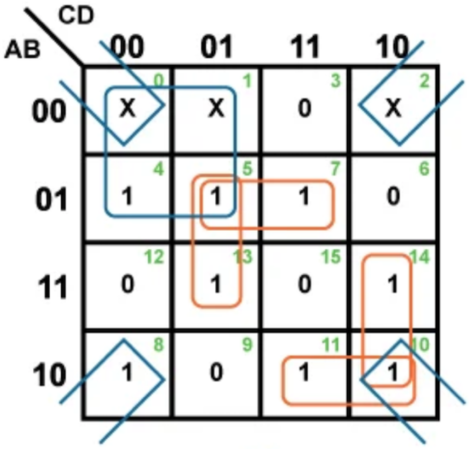


Grouping in K-Map

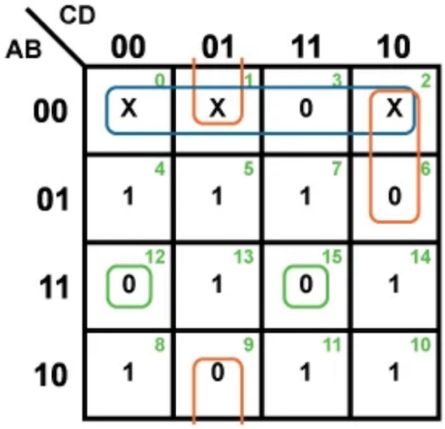
- Once you have drawn the K-Map and populated it with ones, you make groups (which are called implicants).
- A group must contain 2^n min-terms, i.e., a group of 2, 4, 8, or 16 min-terms is allowed.
- A group can cover min-terms horizontally or vertically, but not diagonally.
- A group can cover min-terms of the two horizontal corners OR two vertical corners OR all the four corners.
- Try making a group first that covers maximum number of min-terms (make largest group first).
- Try identifying the min-term which can go in one group only and make its group
- As the size of the group increases, the number of variables in its corresponding term decreases.

Steps to Simplify a Boolean Function using K-Map:

- Step1: Convert Truth Table to K-Map, by populating the 1's (as per the number of variables)
- Step2: Group adjacent 1's (using the rules discussed above)
- Step3: Select the groups that result in minimal sum of product terms
- Step4: Write the terms for each group
- Step5: The simplified Boolean Function is the sum of all the terms you have achieved in step4



(a)



(b)

Implicant vs Prime Implicant vs Essential Prime Implicant:

- **Implicant:** The corresponding terms of all possible groups that exist in a K-map, are called implicants.
- **Prime Implicant**: An implicant that is not a proper subset of any other implicant i.e. it is not completely covered by any single implicant
- **Essential Prime Implicant**: The prime implicant which always appear in the solution. It contains at least one min-term that is not covered by any other prime implicant

What are Don't Care Conditions?

- A Don't care can occur in the truth table under two scenarios.
  - When output do not matter for a set of input sequence.
  - For a input sequence that will never occur.
- In simplification using K-Map, don't care can be considered to make the group larger. It is not mandatory to cover all the don't cares.

Simplify the following Boolean functions, using K-Map. Write down both the original and simplified function. Calculate the number of gates that you need for the original and the simplified function. Finally, do identify the prime implicants and essential prime implicants

1. F(x,y)=Σ(0, 1, 3)
2. F(x,y,z)=Σ(0, 2, 4, 5)
3. F(x,y,z)=Σ(0, 1, 2, 3, 5)
4. F(x,y,z)=Σ(1, 2, 3, 7)
5. F(x,y,z)=Σ(0, 1, 5, 7)
6. F(w,x,y,z)=Σ(1, 3, 6, 11, 14)
7. F(w,x,y,z)=Σ(3, 7, 11, 13, 14, 15)
8. F(w,x,y,z)=Σ(2, 3, 12, 13, 14, 15)
9. F(w,x,y,z)=Σ(11, 12, 13, 14, 15)
10. F(w,x,y,z)=Σ(8, 10, 12, 13, 14)
11. F(w,x,y,z)=Σ(0, 1, 4, 5, 10, 11, 14, 15)
12. F(w,x,y,z)=Σ(1, 4, 6, 7, 8, 9, 10, 11, 15)
13. F(w,x,y,z)=Σ(2, 3, 6, 7, 8, 9, 12, 13)
14. F(w,x,y,z)=Σ(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)
15. F(w,x,y,z)=Σ(1, 3, 4, 5, 7, 9, 11, 14)
16. F(w,x,y,z)=Σ(2, 3, 4, 5, 6, 7, 9, 11, 12, 13)
17. F(w,x,y,z)=Σ(0, 2, 3, 7, 8, 9, 10, 15)
18. F(w,x,y,y,z) =Σ(0, 1, 2, 8, 14), and D=(w,x,y,y,z) =Σ(9, 10)
19. F(w,x,y,y,z) =Σ(3, 7, 9, 11, 13, 15), and D=(w,x,y,y,z) =Σ(4, 5, 6)
20. F(w,x,y,y,z) =Σ(4, 5, 9, 14, 15), and D=(w,x,y,y,z) =Σ(1, 2, 7, 11, 13)