Mahidol University
Wisdom of the Land

Logic and Computer Design Fundamentals

Chapter 2 – Combinational Logic Circuits

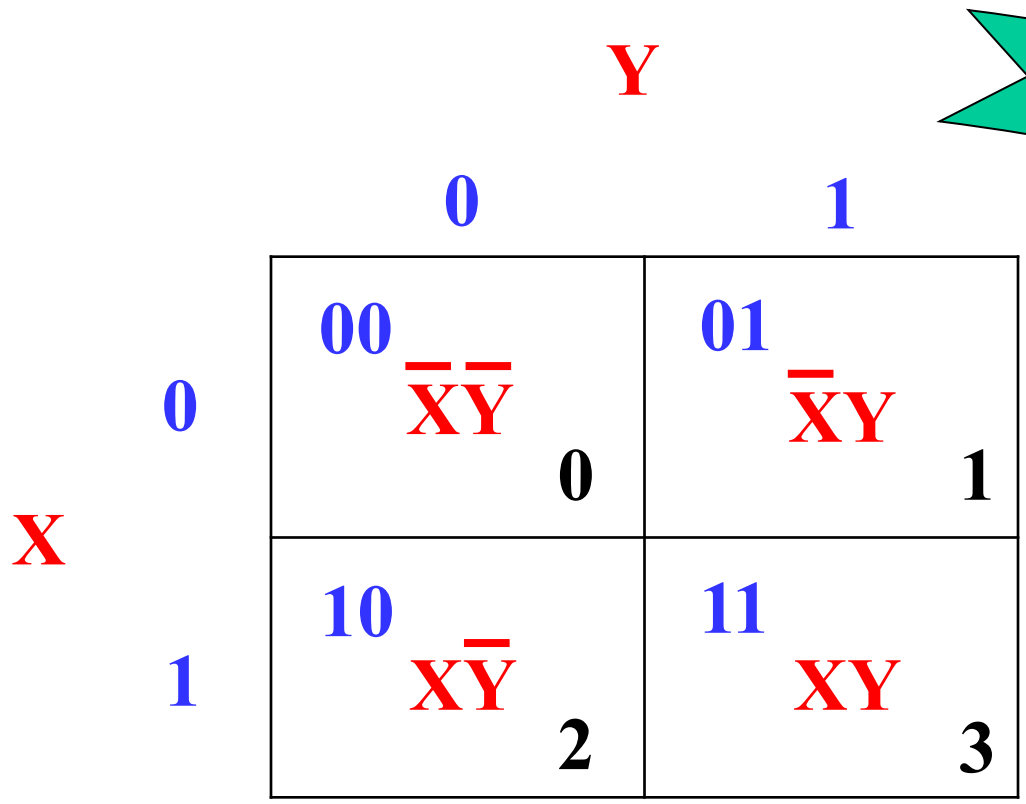
Part 7 – Circuit Optimization (Part 2)

Asst.Prof.Dr. Preecha Tangworakitthaworn
Semester 2/2023

Overview

- **Part 6 – Circuit Optimization**
  - **Review of Simplification using Karnaugh Map Technique**
  - **K-Map with Don't Care Concept**

Two Variable Maps



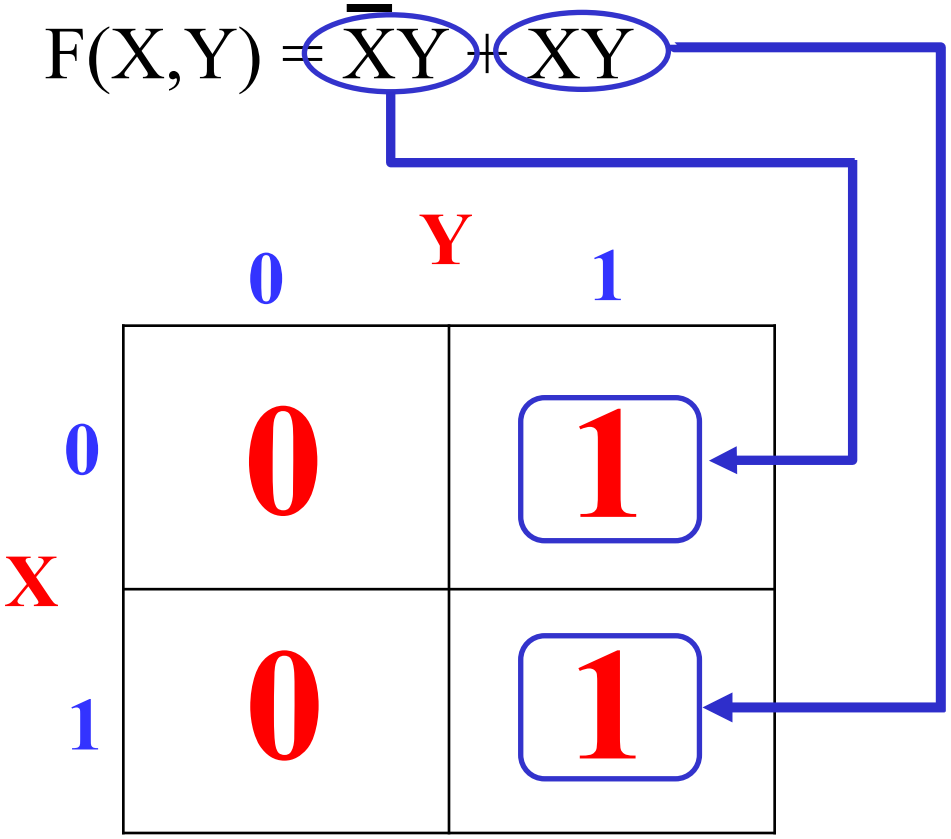
Recommend

Assigned values and variables to K-Map

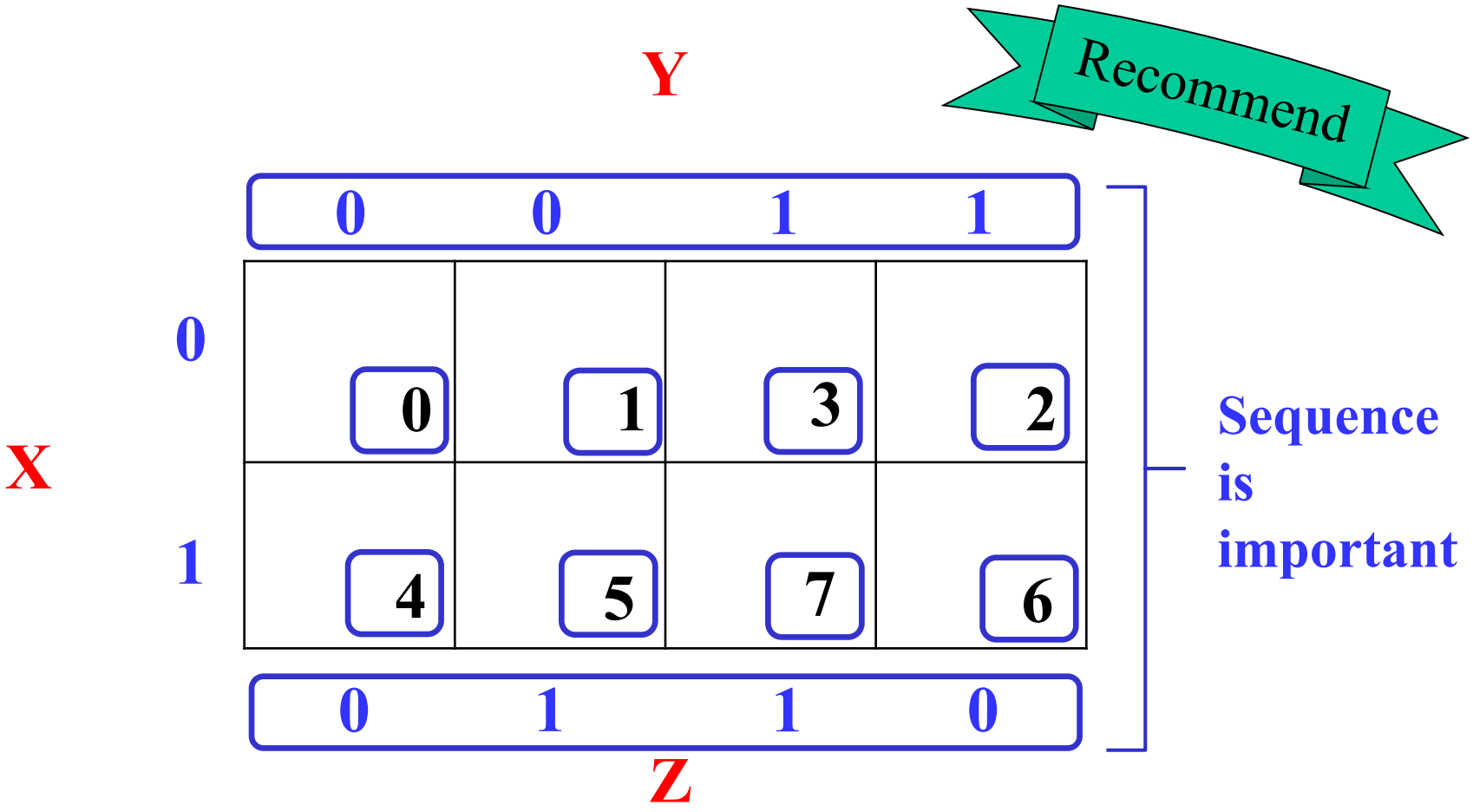
How to draw K-Map?

Recommend

Assign Boolean expression into K-Map

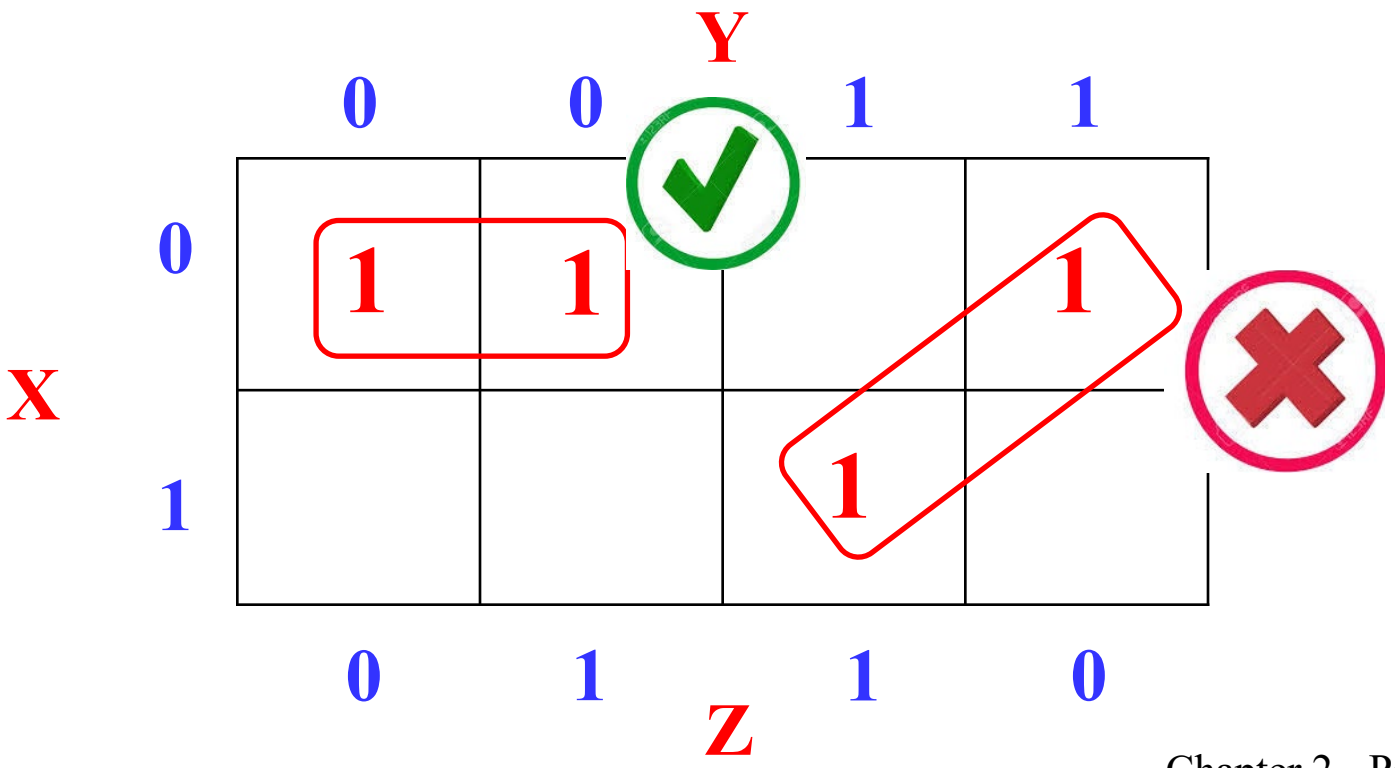


K-Map Template (Three Variables)



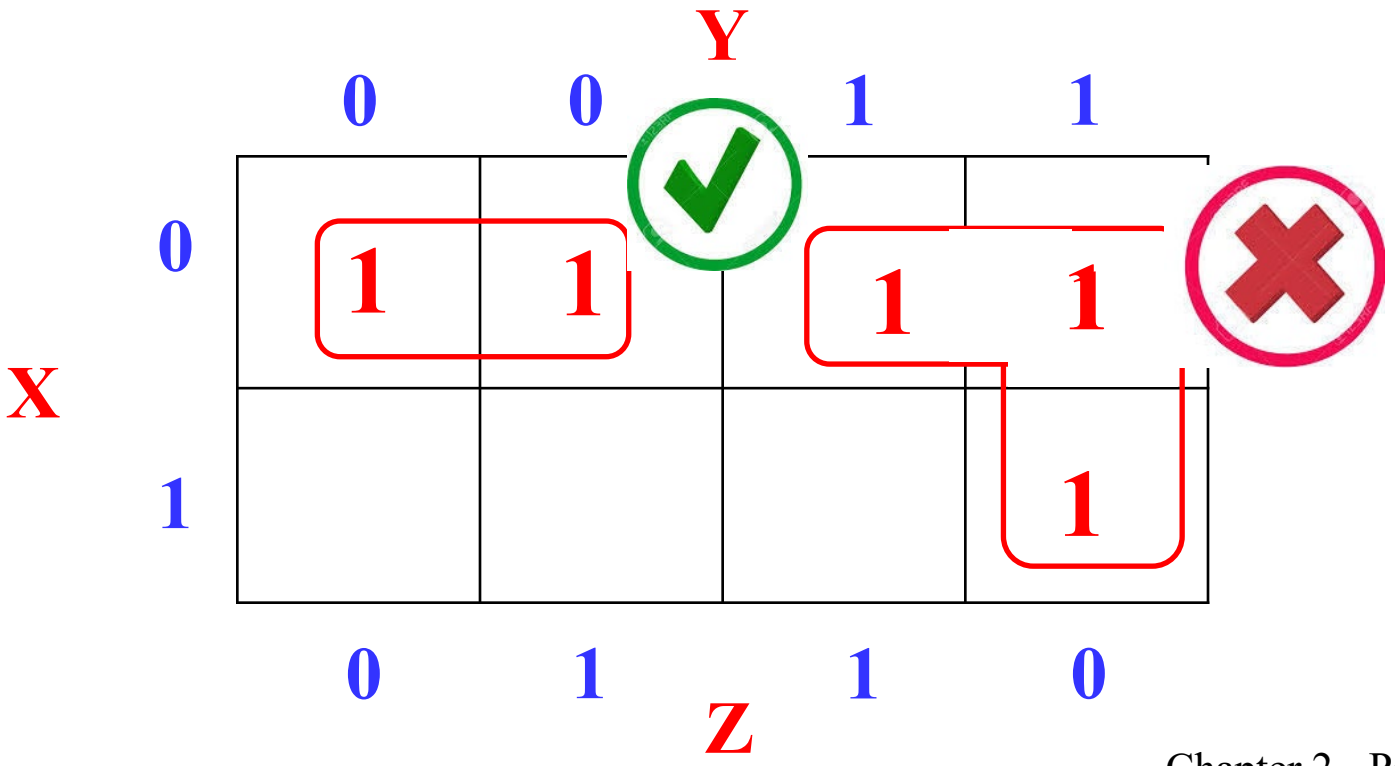
Determination of Adjacent Cell

- if the binary value for an index differs in one bit position, the minterms are adjacent on the K-Map



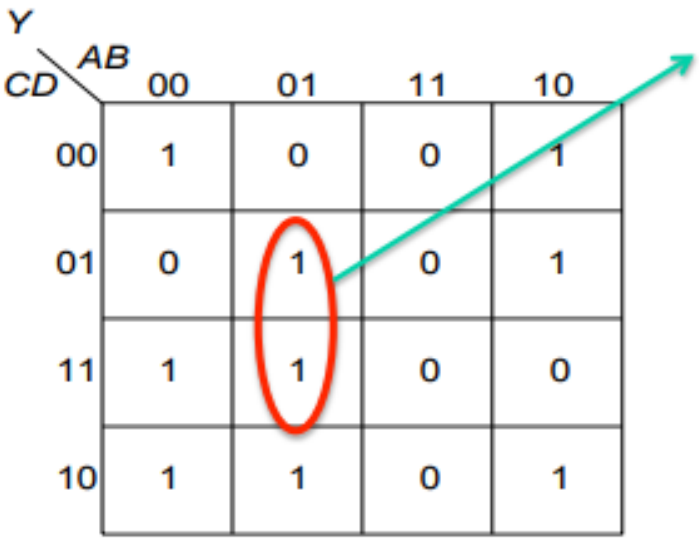
Determination of Adjacent Cell

- Number of cell for Adjacent on the K-Map must equal to 2^n
- Thus, Adjacent must be 1, 2, 4, 8, etc.



Prime Implicants

- Implicant: A product term that has non-empty intersection with on-set F and does not intersect with off-set R
- 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



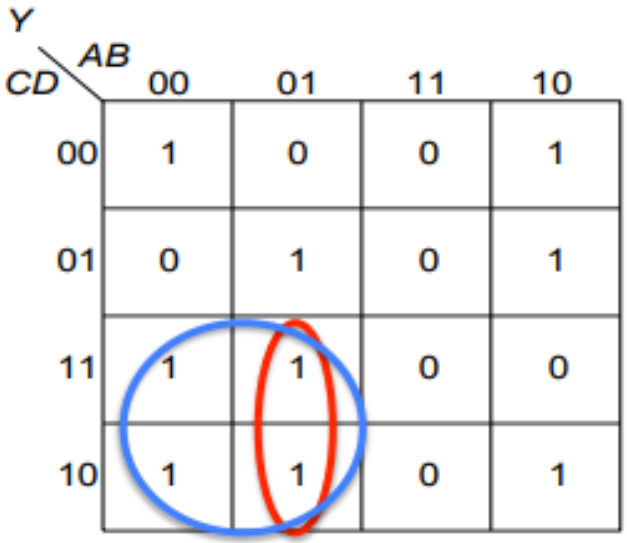
Q: Is this a prime implicant?

A. Yes
B. No

Maximum Adjacent Cells=2

Prime Implicants

- Implicant: A product term that has non-empty intersection with on-set F and does not intersect with off-set R
- 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



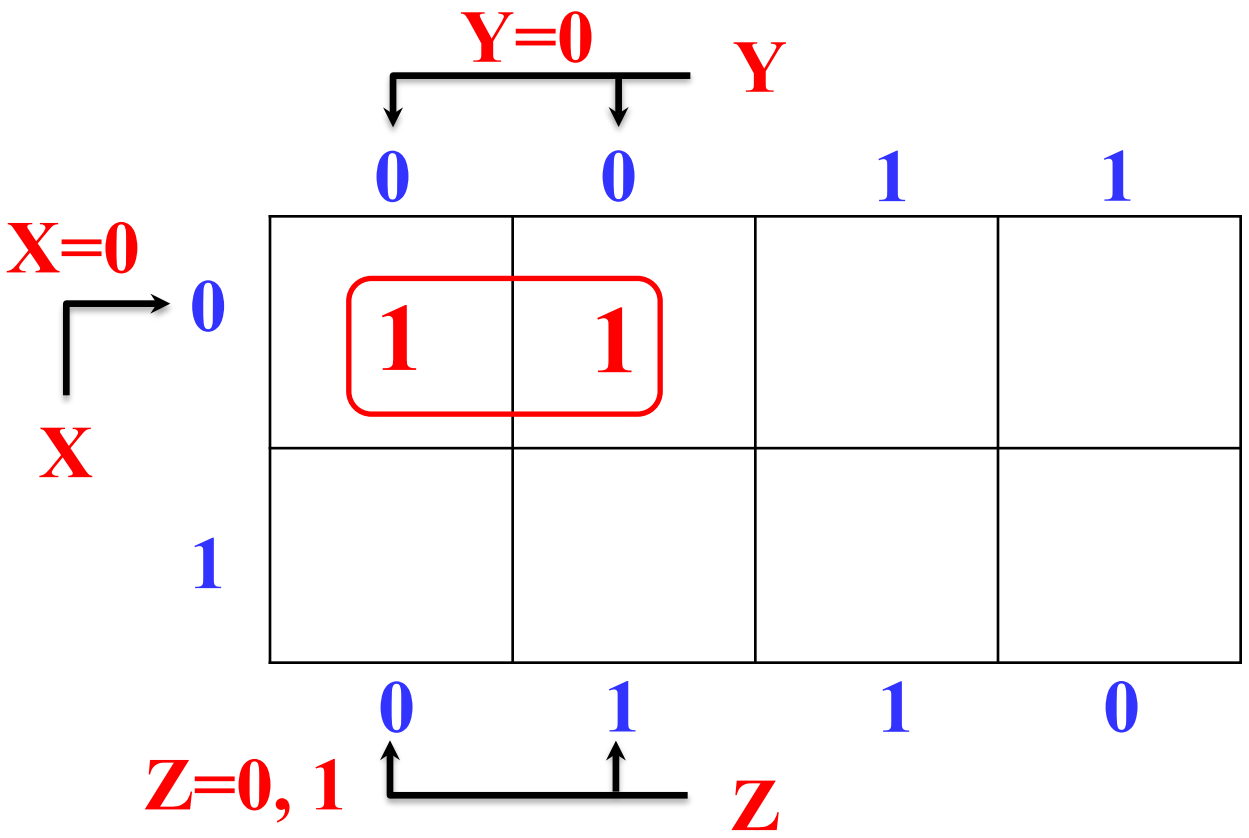
Q: Is the red group a prime implicant?

A. Yes
B. No: Because it is covered by a larger group

Maximum Adjacent Cells=4

Simplification using K-Map

- Step 2: Verify each variable (one by one),
  - 2.1: Adjacent covers either 0 or 1, keep this variable
  - 2.2: Adjacent covers both 0 and 1, remove this variable

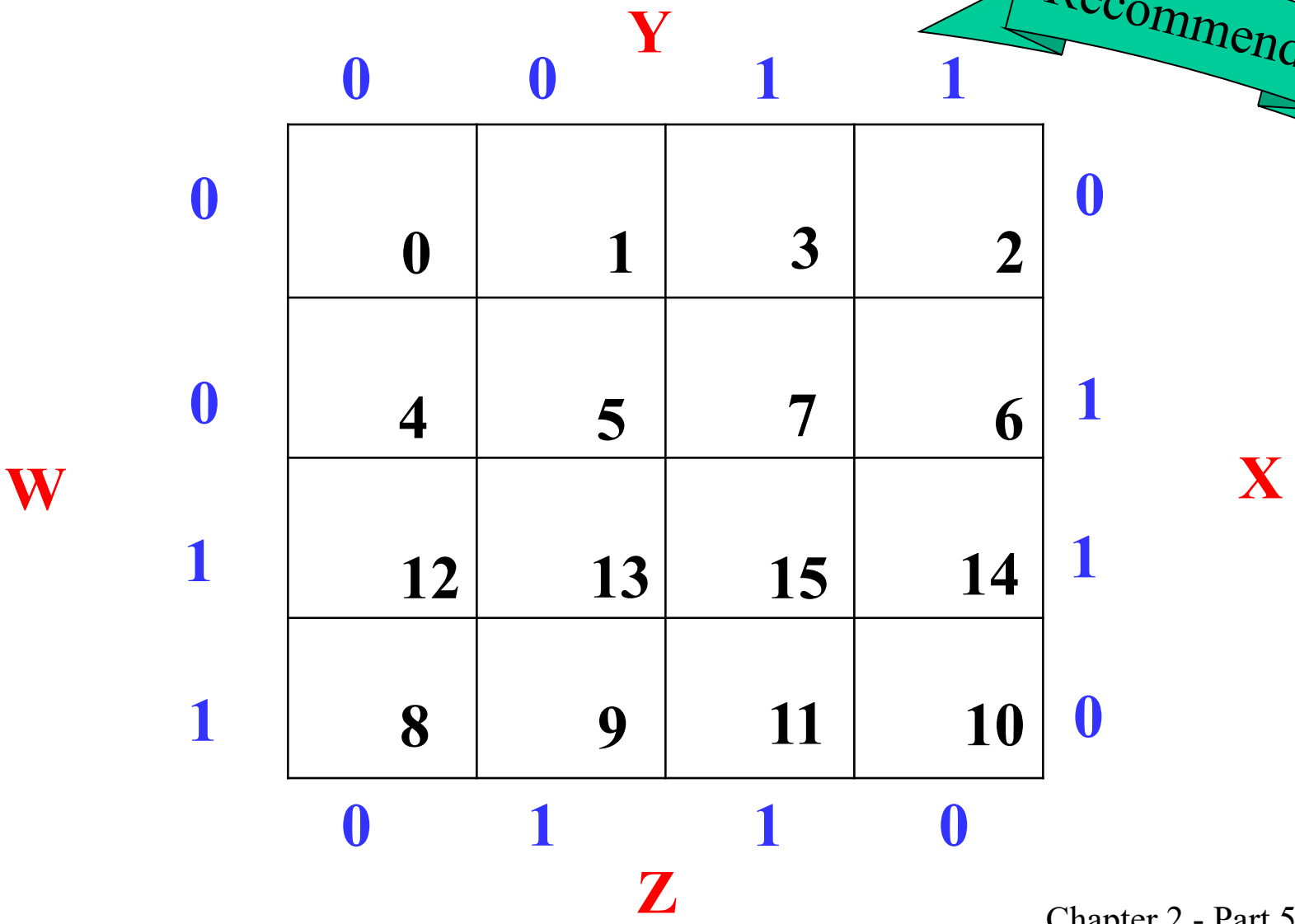


X=0
Y=0
Z=0,1
Thus, remove Z

Solution
F=$\overline{X}\,\overline{Y}$

K-Map Template (Four Variables)

Recommend



Don't Cares in K-Maps

- **Sometimes a function table or map contains entries for which it is known:**
  - **the input values for the minterm will never occur, or**
  - **The output value for the minterm is not used**
- **In these cases, the output value need not be defined**
- **Instead, the output value is defined as a "don't care"**
- **By placing "don't cares" (an "x" entry) in the function table or map, the cost of the logic circuit may be lowered.**
- **Example 1: A logic function having the binary codes for the BCD digits as its inputs. Only the codes for 0 through 9 are used. The six codes, 1010 through 1111 never occur, so the output values for these codes are "x" to represent "don't cares."**

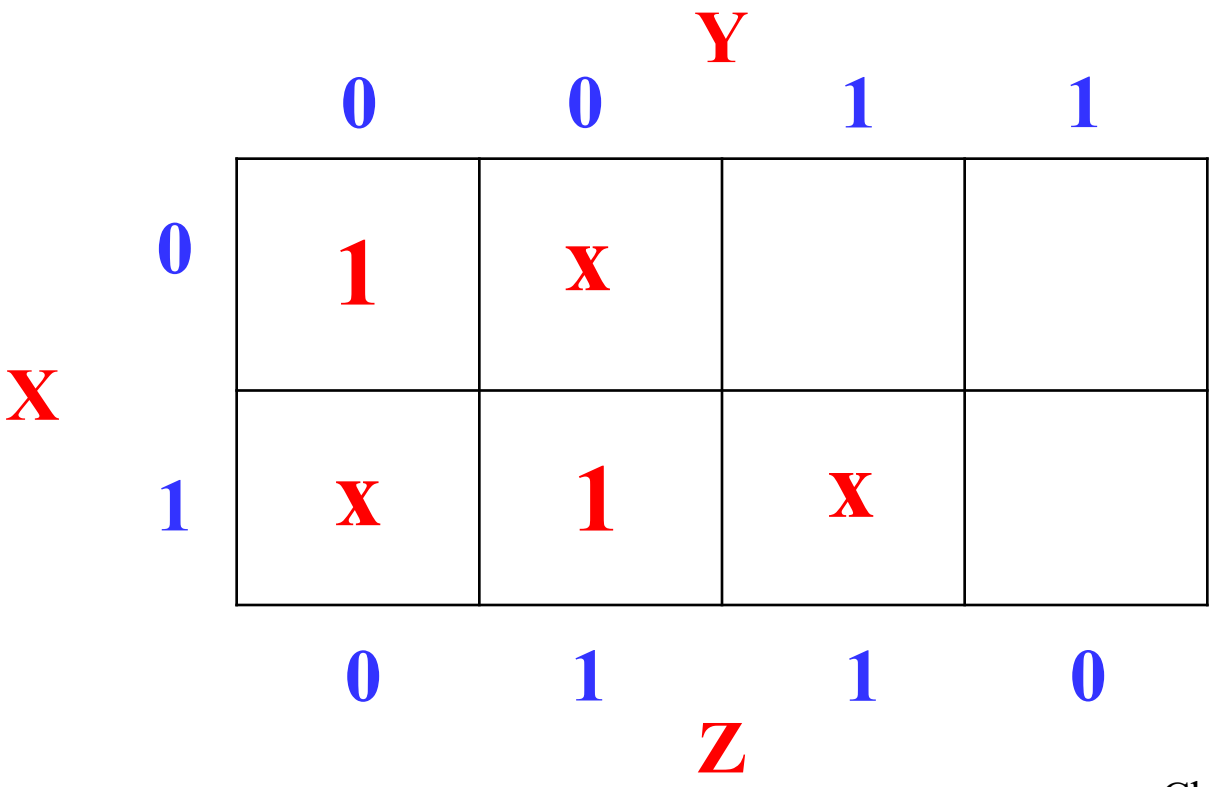
Don't Cares in K-Maps

- **Each don’t care “x” entry may take on either a 0 or 1 value in resulting solutions**
  - If X represents 1, this don’t’ care X can be grouped as adjacent.
  - If X represents 0, this don’t care X can be ignored.

Don't Cares in K-Maps

- **Placing "X" for don't care:**

$$\mathbf{F(X,Y,Z)} = \sum_m (0,5) \text{ and } \mathbf{d(X,Y,Z)} = \sum_d (1,4,7)$$



Determination of Adjacent for Don't Cares

F(X,Y,Z)= $\sum_m$ (0,5) and **d(X,Y,Z)=** $\sum_d$ (1,4,7)