Digital Design

Ms. Umarah Qaseem

1

7

Chapter 3

Gate Level Minimization

2

Karnaugh Map
Don't care states

4

Don't Care States

- You don't always need all 2^n input combinations in an n-variable function;
- If you can guarantee that certain input combinations never occur
- If some outputs aren't used in the rest of the circuit

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	X
1	1	1	1

5

Don't Care States

- An **input-sequence for which the function output does not matter** is called don't care.
- When we do not care either 0 or 1 are used as the sets of inputs. It is such a condition that doesn't make any difference on output.

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	X
1	1	1	1

6

Incompletely Specified Functions

- Sometimes a function is not completely specified
  - For instance, when BCD numbers are handled
- A BCD number is represented by 4 bits with 6 values undefined
  - What to do with these unused (undefined) values?
  - **Answer:** Use them as **don't care** conditions

Decimal Digit	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Note: 1010, 1011, 1100, 1101, 1110, and 1111 are **INVALID CODE!**

7

Don't Care States

- We mark don't-care outputs in truth tables and K-maps with Xs.
- Within a K-map, each X can be considered as either 0 or 1. You should pick the interpretation that allows for the most simplification.

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	X
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	X
1	1	1	1

8

Don't Care States

f(A,B,C,D) = m(1,3,5,7,9) + d(6,8,9)

AB \ CD	00	01	11	10
00	0	1	1	0
01	0	1	1	X
11	0	1	0	0
10	X	X	0	0

C B A D

f = A'D + BC'D without don't cares

f = A'D + C'D with don't cares

9

Example

Simplify the Boolean function

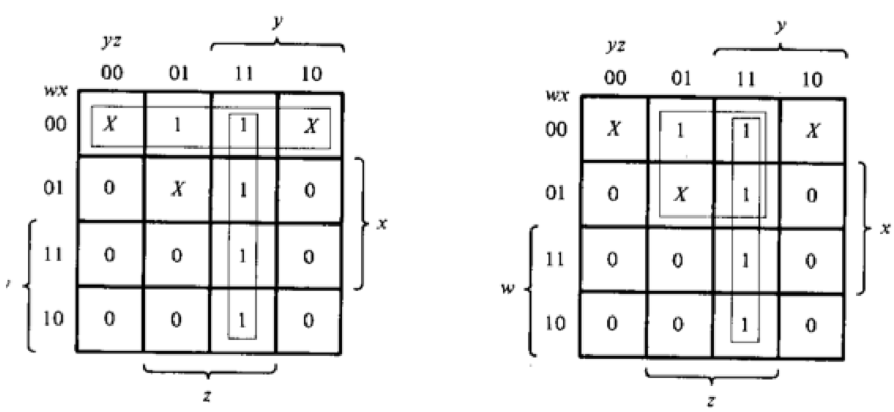
$$F(w, x, y, z) = \Sigma(1, 3, 7, 11, 15)$$

that has the don't-care conditions

$$d(w, x, y, z) = \Sigma(0, 2, 5)$$

11

11



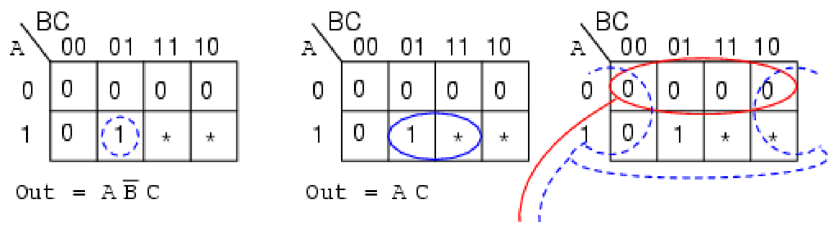
(a) $F = yz + w'x'$

(b) $F = yz + w'z$

12

12

- What should be the valid answer:



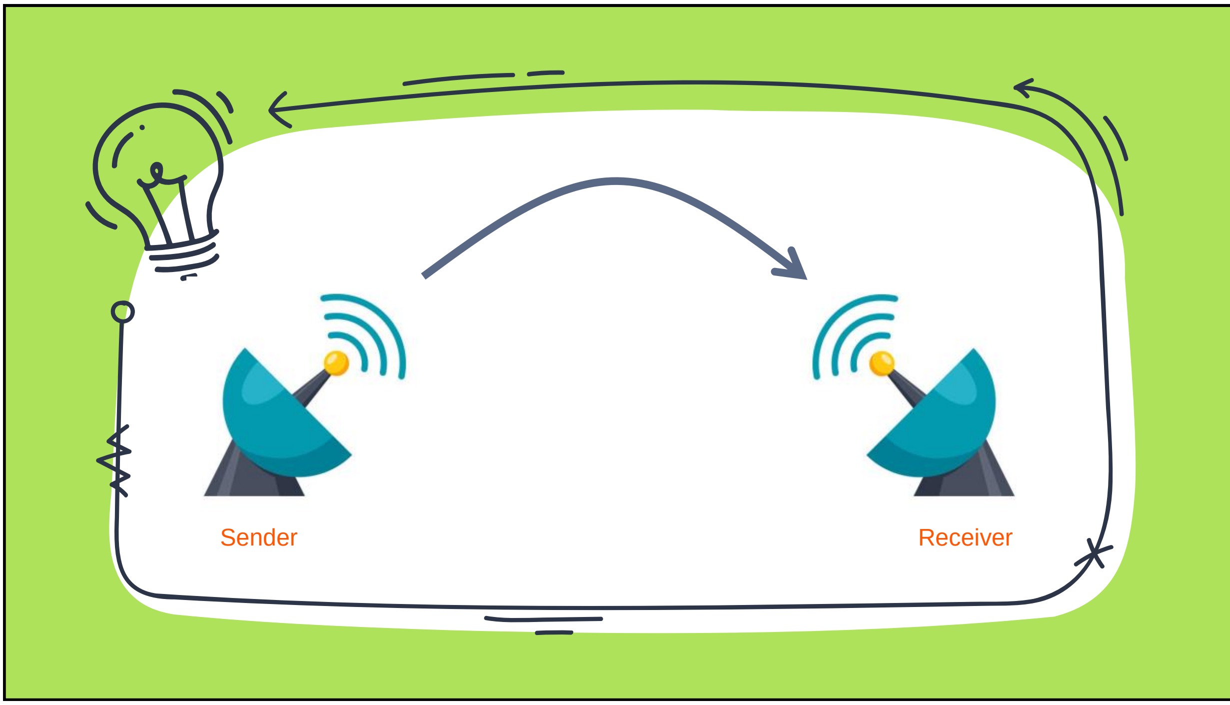
- Ans

13

13

Parity Generator
Parity Checker

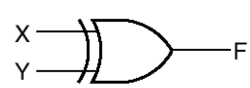
14



15

Exclusive-OR Function

- Exclusive-OR (XOR) function is equal to one if input is different!
- *Function = $x \oplus y = x'y + xy'$*
  (written through minterms)

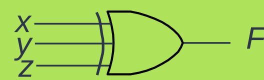


$$F = X\overline{Y} + \overline{X}Y$$
$$= X \oplus Y$$

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

16

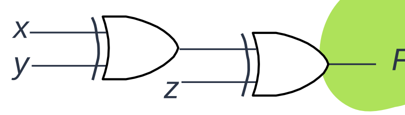
Remember 3-input XOR?



$$F = X \oplus Y \oplus Z$$

- Do you see a pattern?

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

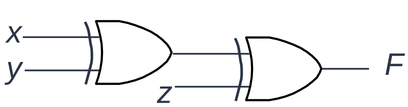


20

Remember 3-input XOR?

$$F = X \oplus Y \oplus Z$$

- Can you get min-terms from this table?
- $A \oplus B \oplus C = A'B'C + A'BC' + AB'C' + ABC$
- $= \Sigma(1, 2, 4, 7)$

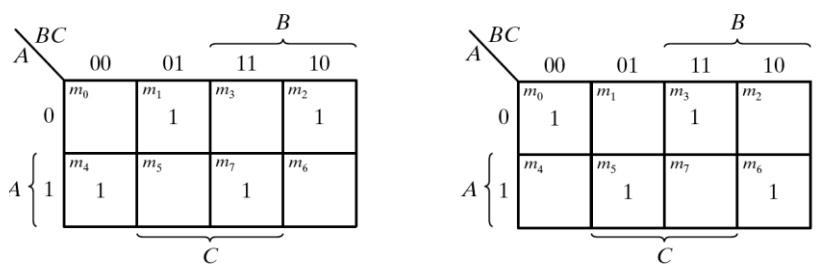


X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

21

It's an Odd Function!

- $A \oplus B \oplus C = AB'C' + A'BC' + ABC + A'B'C = \Sigma(1, 2, 4, 7)$
- XOR is a odd function → an odd number of 1's, then $F = 1$.
- XNOR is a even function → an even number of 1's, then $F = 1$.

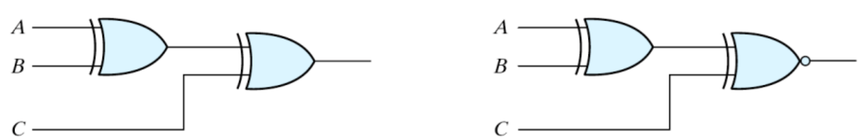


(a) Odd function $F = A \oplus B \oplus C$ (b) Even function $F = (A \oplus B \oplus C)'$

22

XOR & XNOR

- Logic diagrams of odd and even functions



(a) 3-input odd function (b) 3-input even function

Logic Diagrams of Odd and Even Functions

23

Parity - Recall

- What was parity? (In chapter 1)

Three-Bit Message			Parity Bit
x	y	z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Even Parity

24

Parity Generation & Checking

- XOR functions are very useful in systems requiring error-detection and correction codes.
  - A circuit that generates a parity bit is called a **parity generator**.
  - The circuit that checks the parity is called a **parity checker**.

Three-Bit Message			Parity Bit
x	y	z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

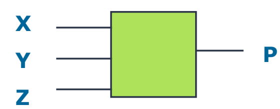
Even Parity

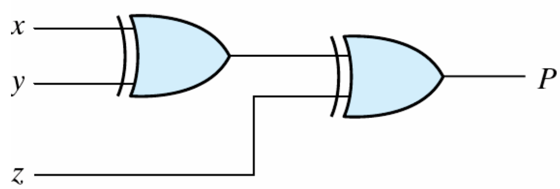
25

Parity Generator

Three-Bit Message			Parity Bit
x	y	z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

A parity bit: P = $x \oplus y \oplus z$





3-bit even parity generator

26

Parity Checker

- Parity check:
  - C = x⊕y⊕z⊕P
  - C=1: one bit error or an odd number of data bit error
  - C=0: correct or an even # of data bit error

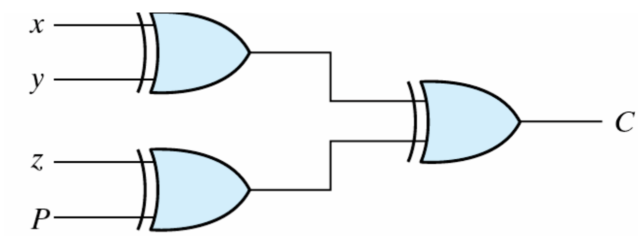
Even-Parity-Checker Truth Table

Four Bits Received				Parity Error Check
x	*y*	*z*	*P*	*C*
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

27

Parity Checker

- Parity check: C = $x \oplus y \oplus z \oplus P$
  - C=1: one bit error or an odd number of data bit error
  - C=0: correct or an even # of data bit error



Parity checker

28

References

- Chapter 3 – Digital Design Morris Mano
- Template is taken from slides carnival.

29