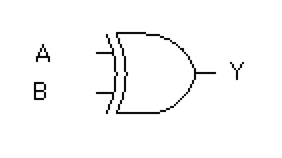
Digital Logic Design (CSNB163)

Module 8

Recaps.. Exclusive OR Gate

- In Module 3, we have learned about Exclusive OR (XOR) gate.
- Boolean Expression
  $AB' + A'B = Y$
  also $A \oplus B = Y$
- Logic Gate



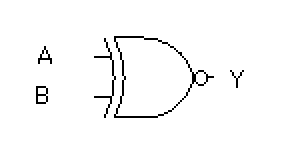
- Truth table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Recaps.. Exclusive NOR Gate

- In Module 3, we have learned about Exclusive NOR (XNOR) gate.
- Boolean Expression
  AB + A'B' = Y
  also (A ⊕ B)'
  also (AB' + A'B)' = Y
- Logic Gate

- Logic Gate



- Truth table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

XOR Properties

- The following identities apply to the XOR operation:
  - **$x \oplus 0 = x$**
  - **$x \oplus 1 = x'$**
  - **$x \oplus x = 0$**
  - **$x \oplus x' = 1$**

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Proving can be performed based on XOR truth table

XOR Properties (cont.)

- The following identities apply to the XOR operation:
  - **$A \oplus B = B \oplus A$**

    Proving based on Postulates 3 – Commutative

    Thus, the two inputs to an XOR can be interchanged!

  - **$(A \oplus B) \oplus C = A \oplus (B \oplus C)$**

    Proving based on Postulate 4 – Associative

    Thus, a three input XOR can be expressed in any manner with or without parenthesis.

XOR Properties (cont.)

- The following identities apply to the XOR operation:
  - **$x \oplus y' = x' \oplus y = (x \oplus y)'$**

    Proving

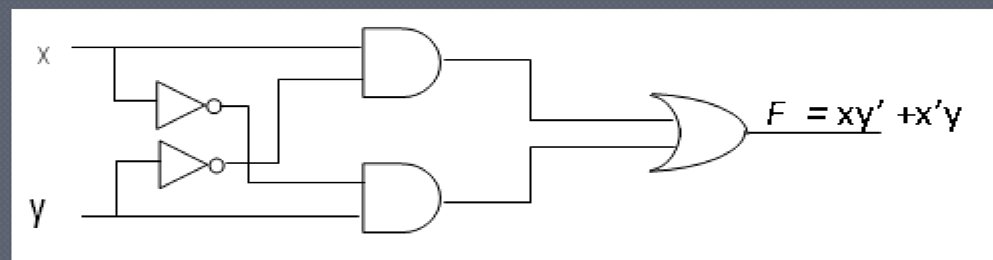
    $x \oplus y = x'y + xy'$ thus $x \oplus y' = x'y' + xy'' = x'y' + xy$

    $x \oplus y = x'y + xy'$ thus $x' \oplus y = x''y + x'y' = xy + x'y'$

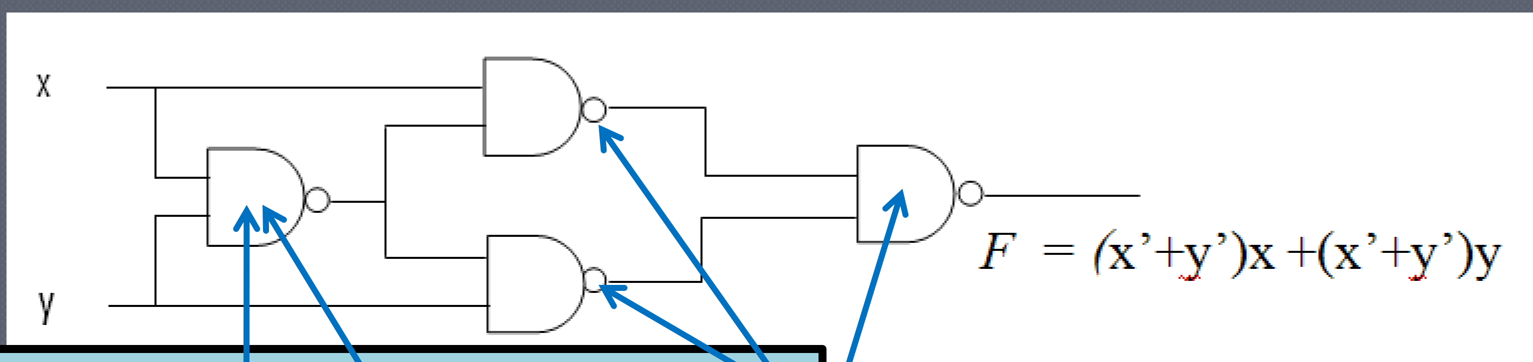
    $xy + x'y' = (x \oplus y)'$ – see XNOR

XOR Implementation

- XOR gates are difficult to be fabricated.
- Most often we implement XOR using:
  - basic gates



  - NAND gates



$$
\begin{aligned}
xy' + x'y &= [xy'+xx'] + [x'y+yy'] \\
&= x(x'+y')+y(x'+y') \\
&= x(x.y)' + y(x.y)' \quad \text{DeMorgan's}
\end{aligned}
$$

$(x')' = x$ Involution

Sum of Minterms of XOR

- XOR can be represented in sum of minterms for any number of inputs:
  - 2 inputs

$$x \oplus y = x'y + xy'$$

  - 3 inputs

$$
\begin{aligned}
x \oplus y \oplus z \quad & = (x \oplus y) \oplus z \\
\text{If} \quad o \quad & = (x \oplus y) = x'y + xy' \\
o' \quad & = (x \oplus y)' = xy + x'y' \\
\text{So, } x \oplus y \oplus z \quad & = (x \oplus y) \oplus z = o \oplus z \\
& = o'z + oz' \\
& = (xy + x'y')z + (x'y + xy')z' \\
& = xyz + x'y'z + x'yz' + xy'z' \\
& = \sum(m7, m1, m2, m4) \\
& = \sum(m1, m2, m4, m7) \text{ -- rearrange}
\end{aligned}
$$

Exercise 1

- Derive the Sum of Minterms for:

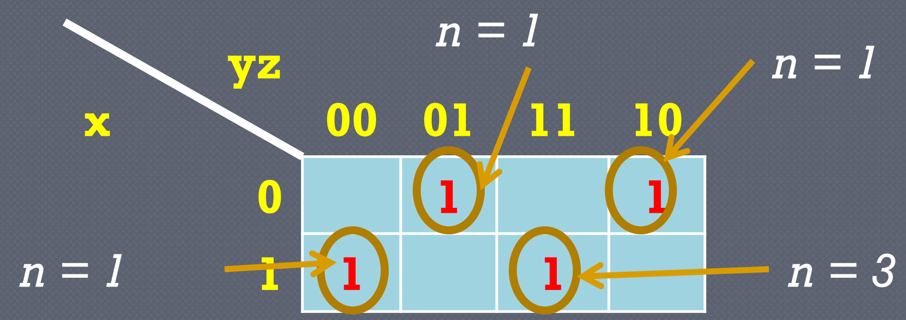
$$F(a, b, c, d) = a \oplus b \oplus c \oplus d$$

$$
\begin{aligned}
a\oplus b\oplus c\oplus d \quad &= (a\oplus b) \oplus (c\oplus d) \\
\text{If } x \quad = a\oplus b \quad &= a'b + ab' \\
x' \quad = (a\oplus b)' \quad &= ab + a'b' \\
\text{If } y \quad = c\oplus d \quad &= c'd + cd' \\
y' \quad = (c\oplus d)' \quad &= cd + c'd' \\
a\oplus b\oplus c\oplus d \quad &= x \oplus y \\
&= x'y + xy' \\
&= (ab + a'b')(c'd + cd') + (a'b + ab')(cd + c'd') \\
&= abc'd+abcd'+a'b'c'd+a'b'cd'+a'bcd+a'bc'd'+ab'cd+ab'c'd' \\
&= \sum(m13,\ m14, m1, m2, m7, m4, m11, m8) \\
&= \sum(m1, m2, m4, m7, m8, m11, m13, m14) \text{ -- rearrange}
\end{aligned}
$$

Odd Function

- Odd function is a function that ONLY returns the value '1' when:
  - ***n* is odd** whereby ***n*** is **the total number of input signals with value '1'**
- XOR of 3 or more input variables makes an odd function.
- E.g.

$$F(x, y, z) = x \oplus y \oplus z \quad = \sum(m1, m2, m4, m7)$$
$$= x'y'z + x'yz' + xy'z' + xyz$$



Exercise 2

- Prove that the given expression is an odd function using a truth table. Derive the K-map.

$$F = a \oplus b \oplus c \oplus d = \sum(m1, m2, m4, m7, m8, m11, m13, m14)$$

a	b	c	d	F	n
0	0	0	0	0	0
0	0	0	1	**1**	1
0	0	1	0	**1**	1
0	0	1	1	0	2
0	1	0	0	**1**	1
0	1	0	1	0	2
0	1	1	0	0	2
0	1	1	1	**1**	3

a	b	c	d	F	n
1	0	0	0	**1**	1
1	0	0	1	0	2
1	0	1	0	0	2
1	0	1	1	**1**	3
1	1	0	0	0	2
1	1	0	1	**1**	3
1	1	1	0	**1**	3
1	1	1	1	0	4

Exercise 2 (cont.)

- K-map:

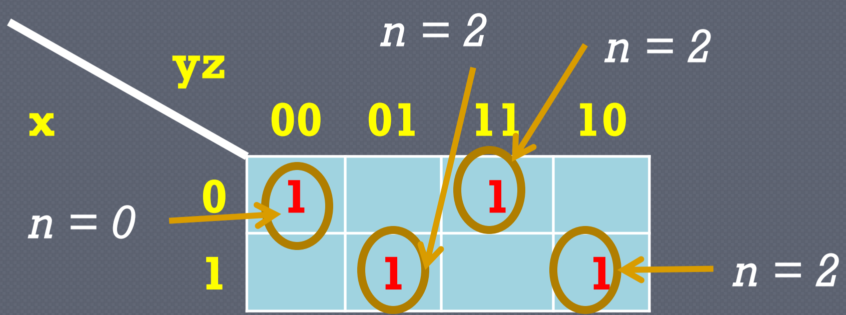
$$F = a \oplus b \oplus c \oplus d = \sum(m1, m2, m4, m7, m8, m11, m13, m14)$$

ab \ cd	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

Even Function

- Even function is a function that ONLY returns the value '1' when:
  - ***n* is even** whereby ***n*** is **the total number of input signals with value '1'**
- The invert of XOR of 3 or more input variables makes an even function.
- E.g.

$$F(x, y, z) = (x \oplus y \oplus z)' = \sum(m0, m3, m5, m6)$$
$$= x'y'z' + xy'z' + x'yz' + x'y'z$$



Exercise 3

- Derive the sum of minterms for the given expression and its K-map: $F = (a \oplus b \oplus c \oplus d)'$

If $F1 = (a \oplus b \oplus c \oplus d)$
$= \sum(m1, m2, m4, m7, m8, m11, m13, m14)$

Then $F = (a \oplus b \oplus c \oplus d)' = F1'$
$= \sum(m0, m3, m5, m6, m9, m10, m12, m15)$

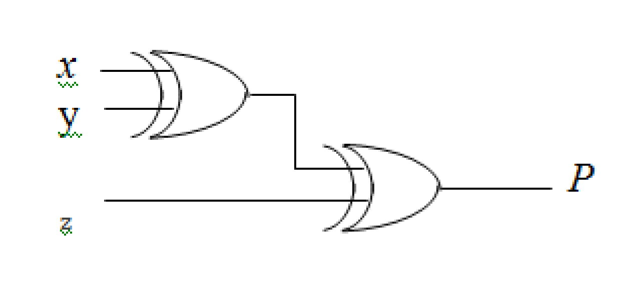
ab \ cd	00	01	11	10
00	1		1	
01		1		1
11	1		1	
10		1		1

Parity Generation and Checking

- Recaps…
  - XOR of 3 and more inputs makes an odd function
  - The invert of XOR of 3 and more inputs makes an even function
- These properties of XOR are used to build circuits to provide for **error detection and correction**.
- This can be accomplished through parity generator and checker.
  - **Parity generator**: a circuit that produces parity bit(s) – often resides at transmitter
  - **Parity checker**: a circuit that checks the parity at receiver

Parity Generator

- A **parity bit** is an extra bit included in the codeword to make the ***n*** (total number of '1's) either odd or even.
- An **even parity generator** utilizes **odd function** to make the number of '1's even (vice versa).
- E.g. $P = x \oplus y \oplus z = \sum(m1, m2, m4, m7)$

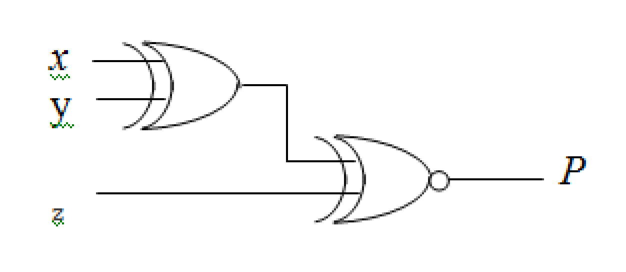


x	*y*	*z*	*P*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Exercise 4

- Design a circuit for an odd parity generator P for three input variable x, y, z. Derive the truth table.

$$P = (x \oplus y \oplus z)' = \sum(m0, m3, m5, m6)$$



x	*y*	*z*	*P*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

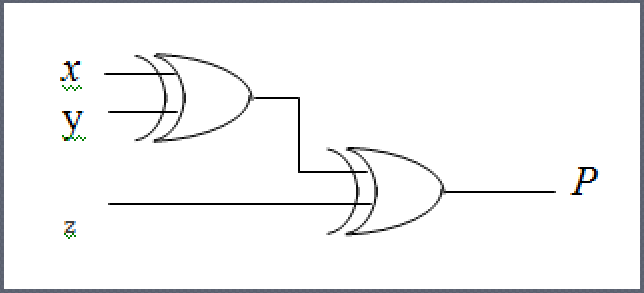
Parity Checker

- If at the transmitter for every **3 data *(x, y, z)*** bits, 1 parity *(P)* bit are constructed for every codeword using **even parity generator**, then the receiver must also be of an **even parity checker** consists of four input variables *(x, y, z and P)*.
- An **even parity checker** utilizes **odd function** to make the number of '1's even(vice versa).
- The output of the parity checker denoted by ***C*** indicates:
  - **No error if *C* = 0**
  - **Error if *C* = 1**

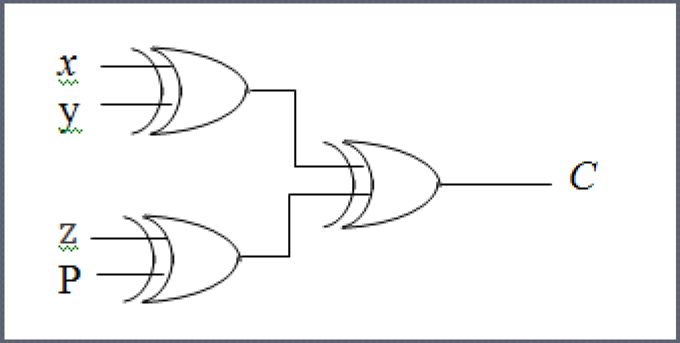
Parity Checker (cont.)

- Example below describes a pair of parity generator and checker:

Parity Generator:
$P = x \oplus y \oplus z$



Parity Checker:
$C = x \oplus y \oplus z \oplus P$



  - If 1100 is received; C = 0 no error.
  - If 1101 is received; C = 1 ERROR!

Digital Logic Design (CSNB163)

End of Module 8