Digital Design

Ms. Umarah Qaseem

1

7

Chapter 4

Combinational Logic

2

Week 9

Combinational Circuits

3

Combinational Circuits

4

Logic Circuits

LOGIC CIRCUITS:

1. Combinational
2. Sequential

5

Combinational Circuits

Combinational logic circuits (circuits without a memory):

A combinational circuit consists of gates whose outputs at any time depend only on the current combination of inputs.

Sequential logic circuits (circuits with memory):

In sequential circuits, the outputs depend on the current inputs and the previous inputs. These networks employ storage elements and logic gates. [Chapters 5 and 9]

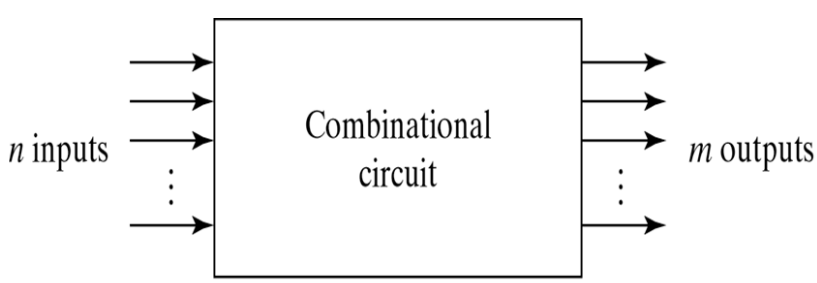
6

Combinational Circuits

- Recall
  - Single/multiple inputs → Single output
- Many realistic problems use multiple outputs
  - Named as combinational circuits
- Combinational circuit
  - Output depends only on input(s)

7

Combinational Circuits



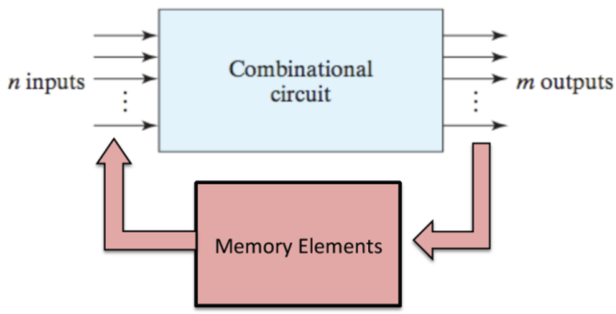
Block Diagram of Combinational Circuit

8

8

Sequential Circuits

- What happens if we add memory to the circuit?



- Becomes a feedback system
  - **Sequential Circuits**

9

Combinational Circuits

- Most important standard combinational circuits are:
  - Adders
  - Subtractors
  - Comparators
  - Decoders
  - Encoders
  - Multiplexers

Available in IC's as MSI

10

10

ANALYSIS

11

Combinational Circuit Analysis

- What is Combinational Circuit Analysis?

Ans: To determine the function of circuit!
(Instead of developing the circuit based on the function, as we did before)

- Two approaches of Circuit analysis

1- Determine the output functions as algebraic expressions

OR

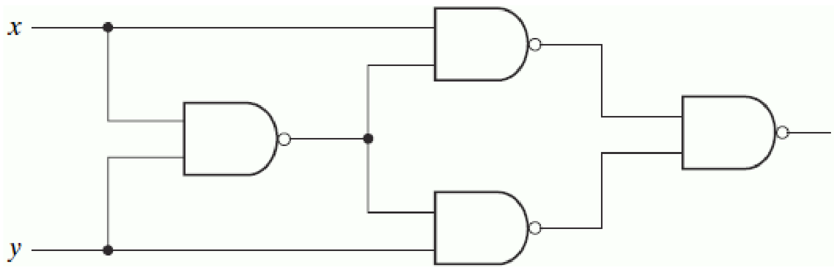
2- Determine the truth table of the outputs

12

Combinational Circuit Analysis: Example1

Approach 1: Determining the output functions as algebraic expressions

- **Analysis steps**
  1. Label all gate outputs with symbols
  2. Determine Boolean function at the output of each gate
  3. Express functions in terms of input variables + simplify



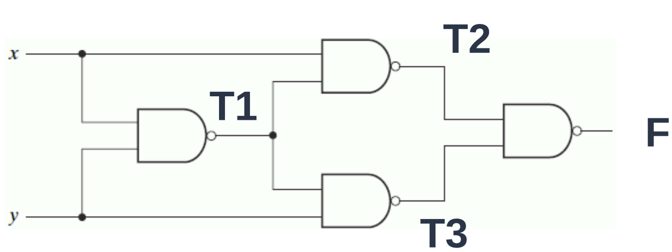
13

Combinational Circuit Analysis: Example1

Approach 1: Determining the output functions as algebraic expressions

- **Step 1**

Label all gate outputs with symbols



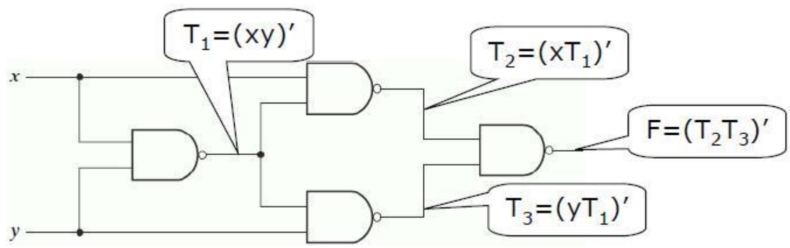
14

Combinational Circuit Analysis: Example1

Approach 1: Determining the output functions as algebraic expressions

Step 2

Determine Boolean function at the output of each gate



15

Combinational Circuit Analysis: Example1

Approach 1: Determining the output functions as algebraic expressions

Step 3

Express functions in terms of input variables + simplify

$$
\begin{aligned}
T_1 &= (xy)' \\
T_2 &= (xT_1)' \\
T_3 &= (yT_1)' \\
F &= (T_2T_3)' = ((xT_1)'(yT_1)')' \\
&= xT_1 + yT_1 = x(xy)' + y(xy)' \\
&= x(x' + y') + y(x' + y') \\
&= xx' + xy' + x'y + yy' \\
&= xy' + x'y = x \oplus y
\end{aligned}
$$

16

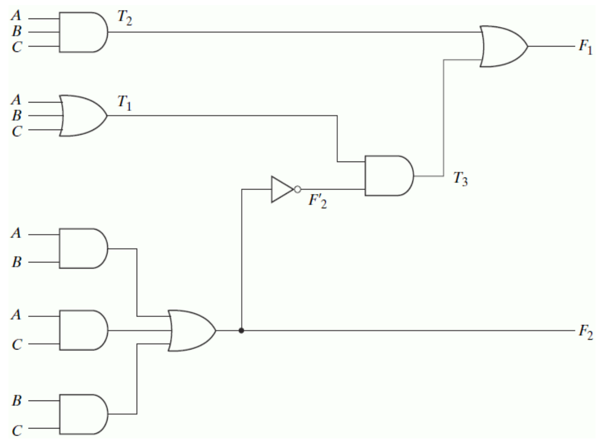
16

Combinational Circuit Analysis: Example2

Approach 2: Determining the truth table of outputs

Analysis steps

1. Label all gate outputs with symbols
2. Determine Boolean function at the output of each gate
3. Determine the truth table of outputs



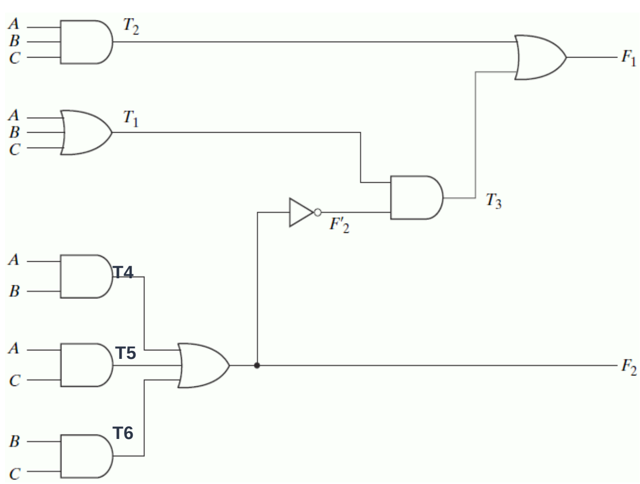
17

Combinational Circuit Analysis: Example2

Approach 2: Determining the truth table of outputs

Step 1

Label all gate outputs with symbols



18

Combinational Circuit Analysis: Example2

Approach 2: Determining the truth table of outputs

- **Step 2**

  Determine Boolean function at the output of each gate

- Here
  - $T_2 = ABC$
  - $T_1 = A+B+C$
  - $T4=AB$, $T5=AC$, $T6=BC$
  - $F_2 = T4+T5+T6$

    $=AB + AC + BC$
  - $T_3 = F_2' T_1$
  - $F_1 = T_3 + T_2$

19

Combinational Circuit Analysis: Example2

Approach 2: Determining the truth table of outputs

- **Step 3**

  Determine the truth table of outputs

A	B	C	T1=A +B+C	T2= A.B.C	T4=A.B	T5=A.C	T6=B.C	F2=T4+T5 +T6	F2’	T3= F2’.T1	F1=T2 +T3
0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	1	1	1
0	1	0	1	0	0	0	0	0	1	1	1
0	1	1	1	0	0	0	1	1	0	0	0
1	0	0	1	0	0	0	0	0	1	1	1
1	0	1	1	0	0	1	0	1	0	0	0
1	1	0	1	0	1	0	0	1	0	0	0
1	1	1	1	1	1	1	1	1	0	0	1

20

DESIGNING

21

Designing a Combinational Circuit

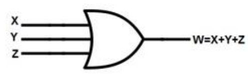
- From the Specification of the circuit, determine the number of inputs and output – Assign a symbol to each
- Derive the Truth Table that defines required relationship between inputs and outputs
- Obtain (simplified) Boolean function for each output as a function of the input variable
- Draw the logic diagram and verify the correctness of the design

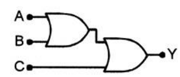
22

22

Combinational Circuit Design

- **Design procedure**
  1. Determine the number of inputs and outputs
  2. Assign symbols
  3. Derive the truth table
  4. Obtain minimized output functions
  5. Obtain simplified output functions
  6. Draw the logic diagram
- Truth tables: input and output columns
- Multiple methods to solve
  - Boolean algebra, map methods, computer aided solution
- Issues to consider
  - Number of gates
  - Gate inputs
  - Propagation delay
  - Number of interconnections





23

Combinational Circuit Design: Example

- Design a circuit that converts a BCD digit to Excess-3 code

24

Circuit Design - Example

- BCD to Excess-3 Code Converter

Specification

- Transforms BCD code for the decimal digits to Excess-3 code for the decimal digits
- BCD code words for digits 0 through 9: 4-bit patterns 0000 to 1001, respectively
- Excess-3 code words for digits 0 through 9: 4-bit patterns consisting of 3 (binary 0011) added to each BCD code word
- 4-Inputs & 4-outputs

25

25

Combinational Circuit Design: Example

- Design a circuit that converts a BCD digit to Excess-3 code
- Step 1&2: Inputs and Outputs
  - Input: BCD digit
    - 4 inputs: A, B, C, D
  - Output: Excess-3 digit
    - 4 outputs: w, x, y, z

26

Combinational Circuit Design: Example

- Design a circuit that converts a BCD digit to Excess-3 code
- Step 1&2: Inputs and Outputs
  - Input: BCD digit
    - 4 inputs: A, B, C, D
  - Output: Excess-3 digit
    - 4 outputs: w, x, y, z
- Step 3: Truth table

A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

27

Circuit Design - Example

- BCD to Excess-3 Code Converter
- Step 3: Truth table

Truth Table

BCD:
A,B,C,D

Excess-3
W,X,Y,Z

Don't Cares
- BCD 1010 to 1111

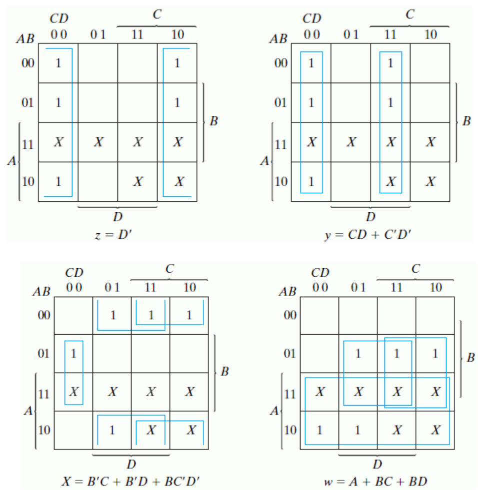
Input BCD A B C D	Output Excess-3 WXYZ
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0

28

28

Combinational Circuit Design: Example

- Step 4: Minimize output functions



29

- Step 4: Minimize output functions

Circuit Design - Example

Boolean Function

z	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

z=D'

y	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

y=CD+C'D'

x	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

x=B'C+B'D+BC'D' =B'(C+D)+BC'D'

w	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

w=A+BC+BD = A+B(C+D)

30

30

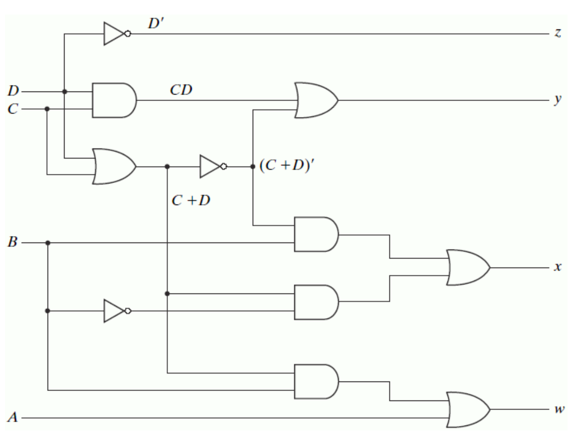
Combinational Circuit Design: Example

- Step 5: Simplification
  - *z = D’*
  - *y = CD+C’D’*
    = CD+(C+D)’
  - *x = B’C+B’D+BC’D’*
    = B’(C+D)+BC’D’
    = B’(C+D)+B(C+D)’
  - *w = A+BC+BD*
    = A+B(C+D)

31

Combinational Circuit Design: Example

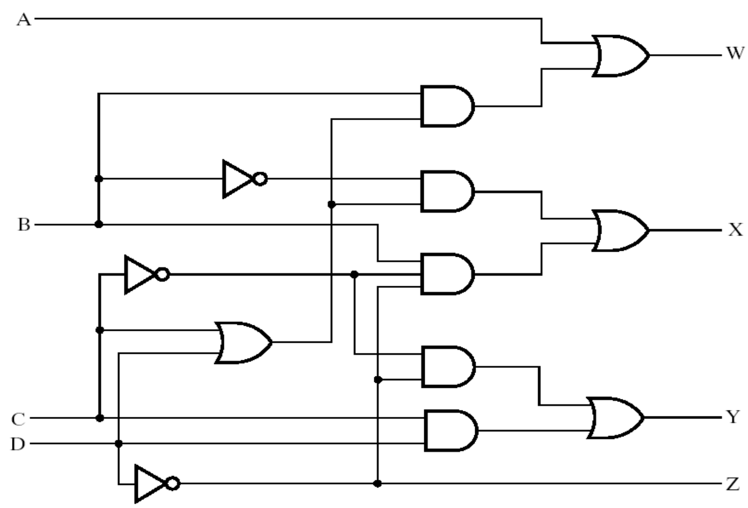
- Step 6: Circuit Diagram



32

Circuit Design - Example

Circuit Diagram



33

33

Circuit Design - Example

Verification

- Find the circuit truth table from the equations and compare to specification truth table:

Input BCD A B C D	Output Excess3 WXYZ
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 0 1 1

The tables match!

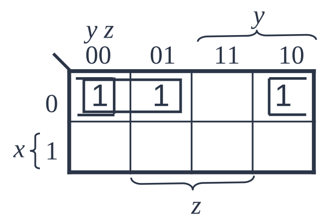
Chapter 3 - Part 1
34

34

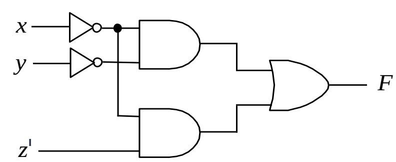
Circuit Design – Example 2

Design a circuit with three inputs and one output. The output is a 1 when the binary value is less than three. The output is 0 otherwise.

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

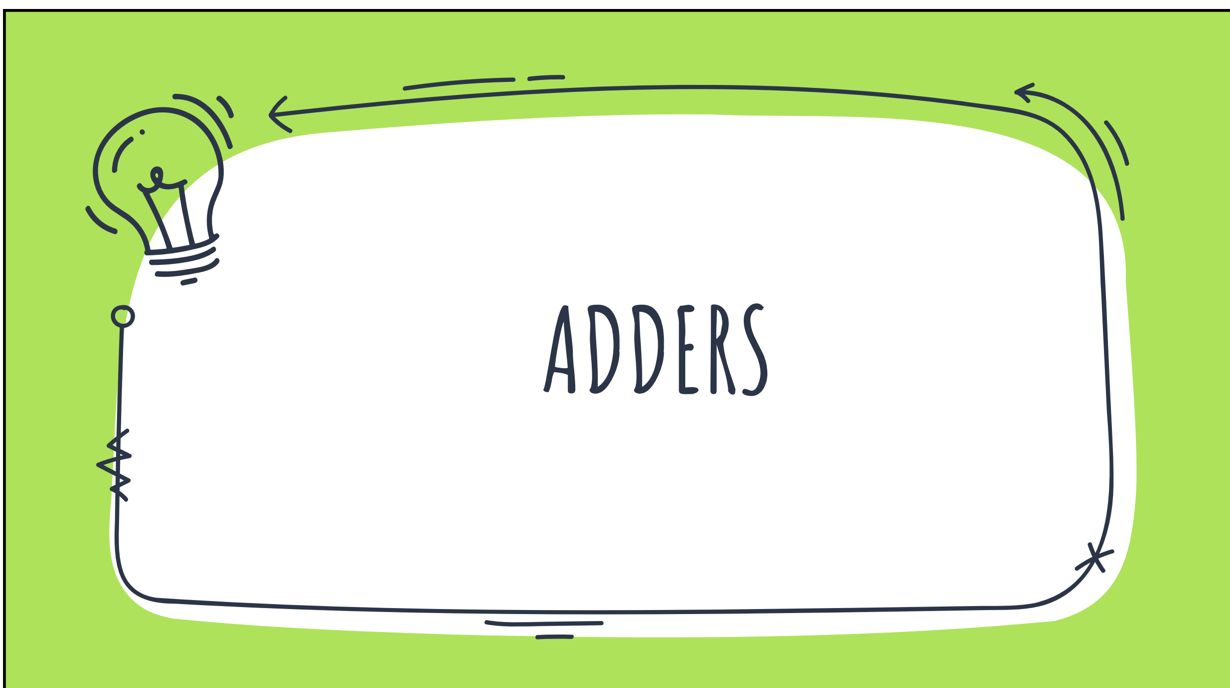


$$F = x'y' + x'z'$$



35

35



36

Binary Addition

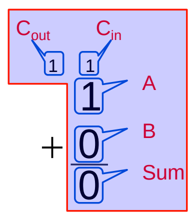
Single Bit Addition:

Carry

$$\begin{array}{r} 0 \\ +\,0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\,0 \\ \hline 1 \end{array} \qquad \begin{array}{r} {}^{1}\;\;1 \\ +\,1 \\ \hline 1\;0 \end{array}$$

Multiple Bit Addition:

$$\begin{array}{r} 6 \\ +\,3 \\ \hline 9 \end{array} \qquad \begin{array}{r} {}^{1}\;{}^{1}\;\;\;\;\; \\ 0\;1\;1\;0 \\ +\,0\;0\;1\;1 \\ \hline 1\;0\;0\;1 \end{array}$$



37

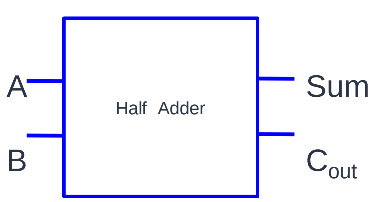
Two Types of Adders

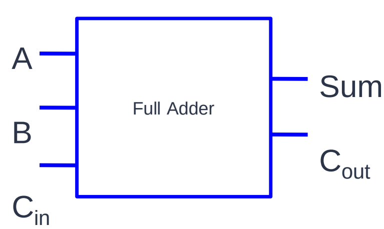
Half Adder

- 2 Inputs (A & B)
- 2 Outputs (Sum & C_{out})
- Used for LSB only

Full Adder

- 3 Inputs (A, B, C_{in})
- 2 Outputs (Sum & C_{out})
- Used for all other bits





38

38

Binary Adder-Subtractor

- Most Basic arithmetic function is Addition of two binary digits
  - 0+0=0, 1+0=1, 0+1=1, 1+1=10 (Carry)
  - Carry is added to the next higher order pair of significant values
- A combinational circuit that performs addition of two bit is called **Half Adder**
- A combinational circuit that performs addition of three bits is called **Full Adder** (Adding two half adder)

39

39

Binary Adders

- Addition is important function in computer system
- What does an adder do?
  - Add binary digits
  - Generate carry if necessary
  - Consider carry from previous computation
- Binary adders operate bit-wise
  - A 16-bit adder uses 16 one-bit adders
- Binary adders come in two flavors
  - Half adder adds two bits and generates result and carry
  - Full adder considers carry input in addition to half adder
  - Two half adders make one full adder

40

Binary Half Adder

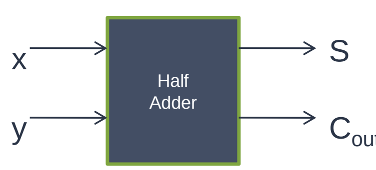
- **Specification**
  - Design a circuit that adds two bits and generates the sum and a carry
- **Input/Output**
  - Two inputs: *x*, *y*
  - Two output: *S* (sum), *C* (carry)
- **Functionality**

	0	0	1	1
+	0	1	0	1
sum	0	1	1	0
carry out	0	0	0	1

41

Half Adder

x	y	C_{out} (Carry)	S (Sum)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



S(A,B)	0	1
0	0	1
1	1	0

$S = x'y+xy'$

C	0	1
0	0	0
1	0	1

$C_{out} = xy$

42

42

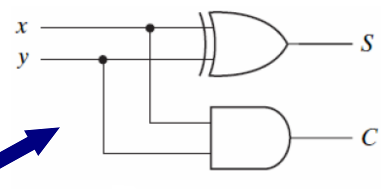
Binary Half Adder

	0	0	1	1
+	0	1	0	1
sum	0	1	1	0
carry out	0	0	0	1

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

C= $\sum(3)$=xy

S= $\sum(1,2)$= x'y+xy'

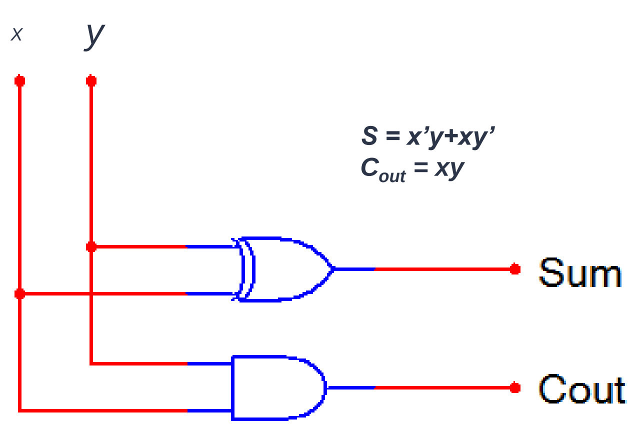


$S = x \oplus y$

$C = xy$

43

Half Adder



S = x'y+xy'

C_{out} = xy

44

44

Full Adder

- Half adder works only for a single bit
  - When multiple bits are involved, carry bits should be considered
  - Solution → Full adder
- **Specifications**
  - A circuit that adds three bits and generates sum and carry
- **Input/output**
  - Three inputs: x, y, C_{in}
  - Two outputs: S (Sum), C_{out} (Carry)
- **Truth table**

x	y	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

45

Full Adder

A	B	C_{in}	Sum	C_{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

46

46

Full Adder

A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

	$\overline{C_{in}}$	C_{in}
$\overline{A}\,\overline{B}$	0	0
$\overline{A}B$	0	1
AB	1	1
$A\overline{B}$	0	1

BC_{in}

AB

AC_{in}

$$C_{out} = AB + BC_{in} + AC_{in}$$

47

47

Full Adder

A	B	C_{in}	Sum	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

	$\overline{C_{in}}$	C_{in}
$\overline{A}\,\overline{B}$	0	1
$\overline{A}B$	1	0
AB	0	1
$A\overline{B}$	1	0

$$Sum = \overline{A}\,\overline{B}C_{in} + \overline{A}B\overline{C_{in}} + ABC_{in} + A\overline{B}\,\overline{C_{in}}$$

48

48

Full Adder

$$\text{Sum} = \overline{A}\,\overline{B}\,C_{IN} + \overline{A}\,B\,\overline{C_{IN}} + A\,B\,C_{IN} + A\,\overline{B}\,\overline{C_{IN}}$$

$$\text{Sum} = \overline{A}\left(\overline{B}\,C_{IN} + B\,\overline{C_{IN}}\right) + A\left(B\,C_{IN} + \overline{B}\,\overline{C_{IN}}\right)$$

$$\text{Sum} = \overline{A}\left(B \oplus C_{IN}\right) + A\left(\overline{B \oplus C_{IN}}\right)$$

Let $K = B \oplus C_{IN}$ and substitute

$$\text{Sum} = \overline{A}\left(K\right) + A\left(\overline{K}\right)$$

$$\text{Sum} = A \oplus K$$

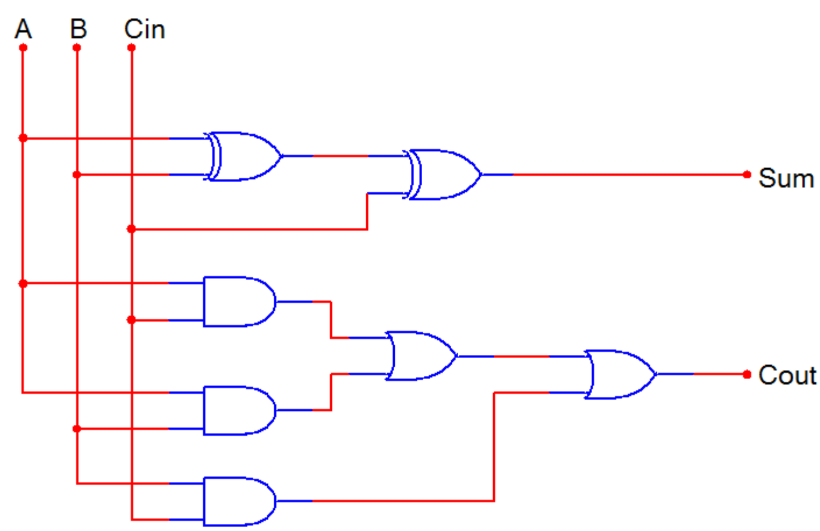
Replacing $B \oplus C_{IN}$ for K

$$\text{Sum} = A \oplus B \oplus C_{IN}$$

49

49

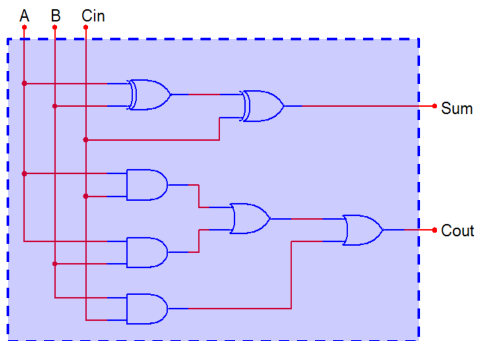
Full Adder



50

50

Full Adder



SSI - Full Adder

51

51

Full Adder

$$S = x'y'C_{in} + x'yC'_{in} + xy'C'_{in} + xyC_{in} = x \oplus y \oplus C_{in}$$
$$C_{out} = xy + xC_{in} + yC_{in} = (x \oplus y)C_{in} + xy$$

52

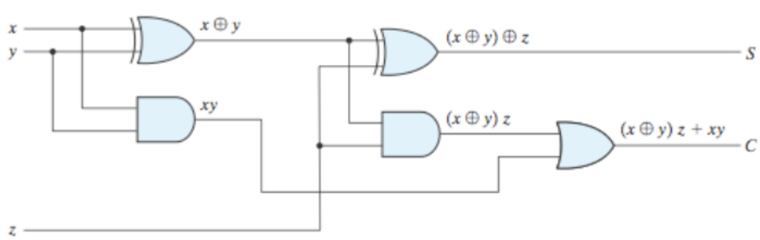
52

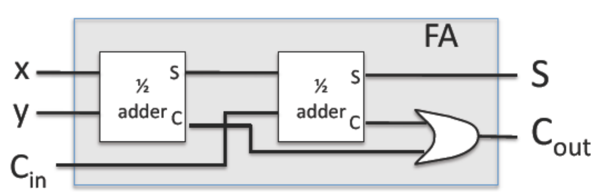
Full Adder from Half Adders

- How can two half adders make a full adder?
- Observations
  - Three inputs x, y, z can be added in two steps
    - *x+y+z = (x+y) + z*
  - What about the carry?
    - Carry can occur when adding *x+y* and when adding z
- Full adder: $S = x \oplus y \oplus z$, $C = xy + (x \oplus y)z$

53

Full Adder from Half Adders



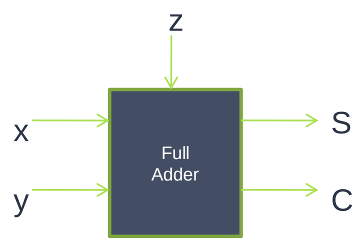


54

Full Adder – Implementation with Two Half Adders

Inputs			Outputs	
x	*y*	*z*	*S*	*C*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = z \oplus (x \oplus y)$$



x \ yz	00	01	11	10
0			1	
1		1	1	1

$$C = xy + xz + yz$$

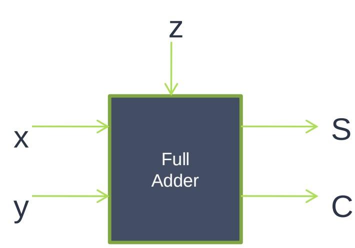
55

55

Full Adder – Implementation with Two Half Adders

Inputs			Outputs	
x	*y*	*z*	*S*	*C*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = z \oplus (x \oplus y)$$



x \ yz	00	01	11	10
0			1	
1		1	$\boxed{1}$	$\boxed{1}$

$$C = xy + xy'z + x'yz$$

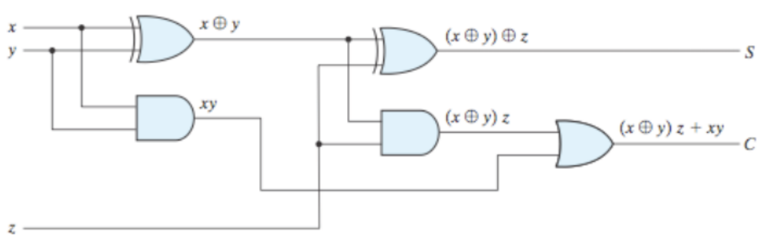
$$C = xy + (x \oplus y)z$$

56

56

Full Adder – Implementation with Two Half Adders

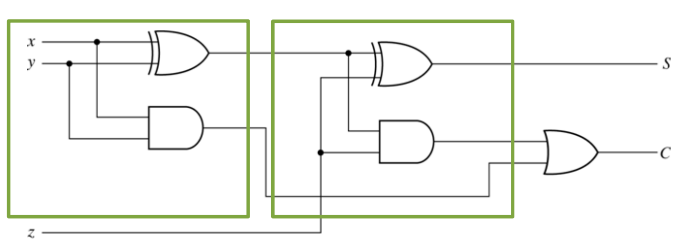
$$C = xy + xy'z + x'yz$$
$$C = xy + (x \oplus y)z$$



57

57

Full Adder – Implementation with Two Half Adders



$$S = z \oplus (x \oplus y) \qquad C = xy + (x \oplus y)z$$

58

58

Binary Adder

- Binary Adder is a circuit that produces sum of two binary numbers
- It can be constructed with full adders (FA) connected in cascade, with output carry from one connected to the input carry of the next full adder

Cascading Adders – Four bits

$$\begin{array}{r} 0\ 1\ 1\ 0 \\ +\ 0\ 0\ 1\ 1 \\ \hline 1\ 0\ 0\ 1 \end{array}$$

Example: 6 + 3 = 9

General Form

$$\begin{array}{ccccc} & C_{out_3} & C_{out_2} & C_{out_1} & C_{out_0} \\ & A_3 & A_2 & A_1 & A_0 \\ + & B_3 & B_2 & B_1 & B_0 \\ \hline & S_3 & S_2 & S_1 & S_0 \end{array}$$

Binary n-bit Adder

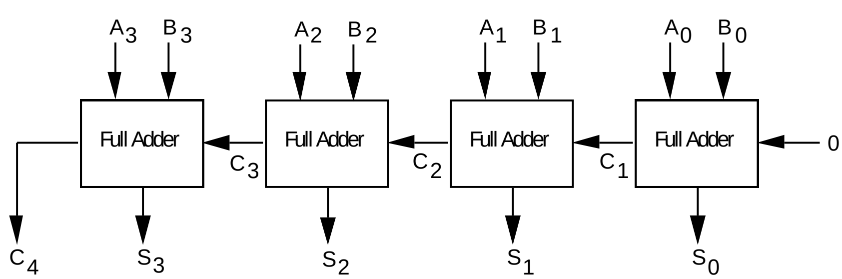
- How can we build an *n*-bit adder
- from full adders?
  - One adder for each bit (*n* total)
  - Connect carry to next adder's input
  - Output: sequence of sums and a final carry
- 4-bit adder circuit (Ripple Carry Adder)

1	0	1	1	
	1	0	0	1
	1	0	1	1
	0	1	0	0

C4	C3	C2	C1	C0
	A3	A2	A1	A0
	B3	B2	B1	B0
	S3	S2	S1	S0

61

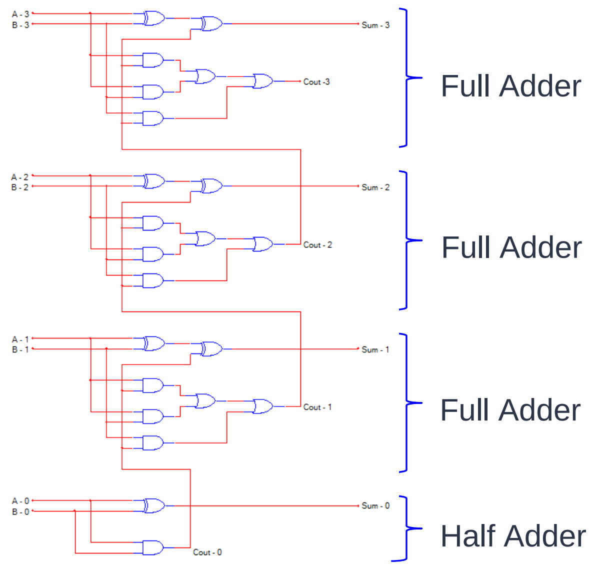
4-Bit Binary Adder



C 1 1 1 0
A 0 1 0 1
B 0 1 1 1
S 1 1 0 0

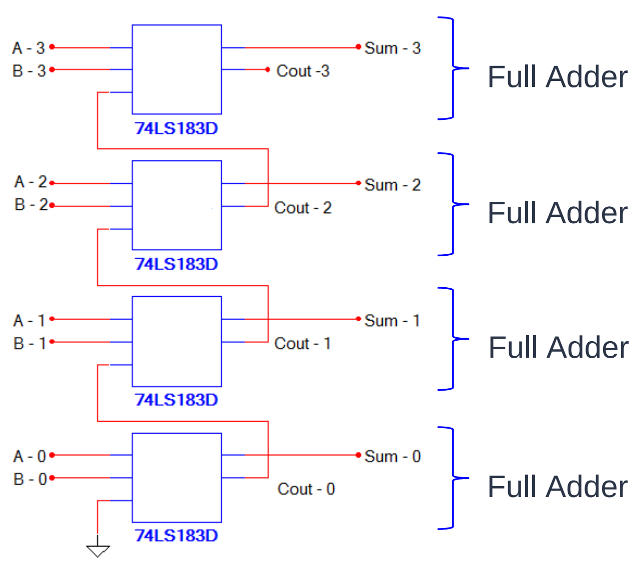
62

62


A - 3
B - 3
Sum - 3
Cout -3
Full Adder
A - 2
B - 2
Sum - 2
Cout - 2
Full Adder
A - 1
B - 1
Sum - 1
Cout - 1
Full Adder
A - 0
B - 0
Sum - 0
Cout - 0
Half Adder

63

63

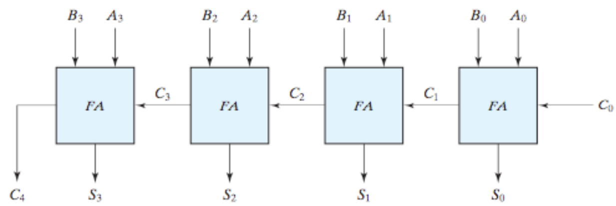

A - 3
B - 3
Sum - 3
Cout -3
74LS183D
Full Adder
A - 2
B - 2
Sum - 2
Cout - 2
74LS183D
Full Adder
A - 1
B - 1
Sum - 1
Cout - 1
74LS183D
Full Adder
A - 0
B - 0
Sum - 0
Cout - 0
74LS183D
Full Adder

64

64

Ripple Carry Adder

- How long does it take to complete an addition?
  - Carry needs to propagate through circuit – a problem



- Speed of addition is critical
  - Fundamental arithmetic operation
- How can we speed up addition?
  - Determine carries ahead of time (carry look-ahead)

65

References

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

66