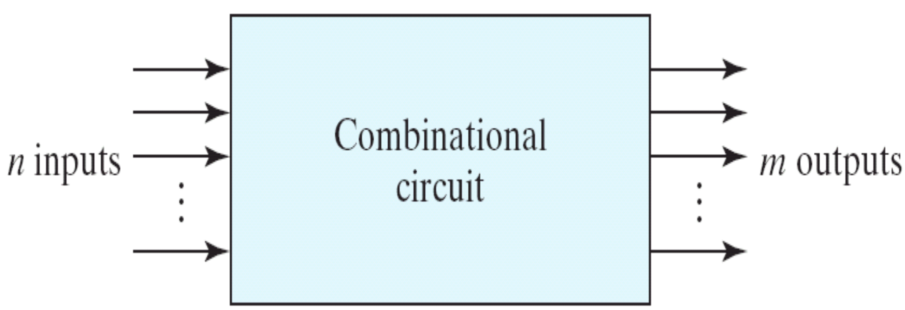
Lecture – 7 - Chapter 4

Outline

- Analysis of Combinational Logic (CL)
- Design of CL
- Classical CL circuits
- Adder

Combinational Logic (CL)

- CL consists of input variables, logic gates, and output variables.
- CL is a logic that has no feedback path from output to input and no memory elements.



- For n input variable, 2^n binary input combinations.
- Specified with a truth table, list output values for each combination of input variables.
- Described by m Boolean functions

COMBINATIONAL LOGIC (CL)

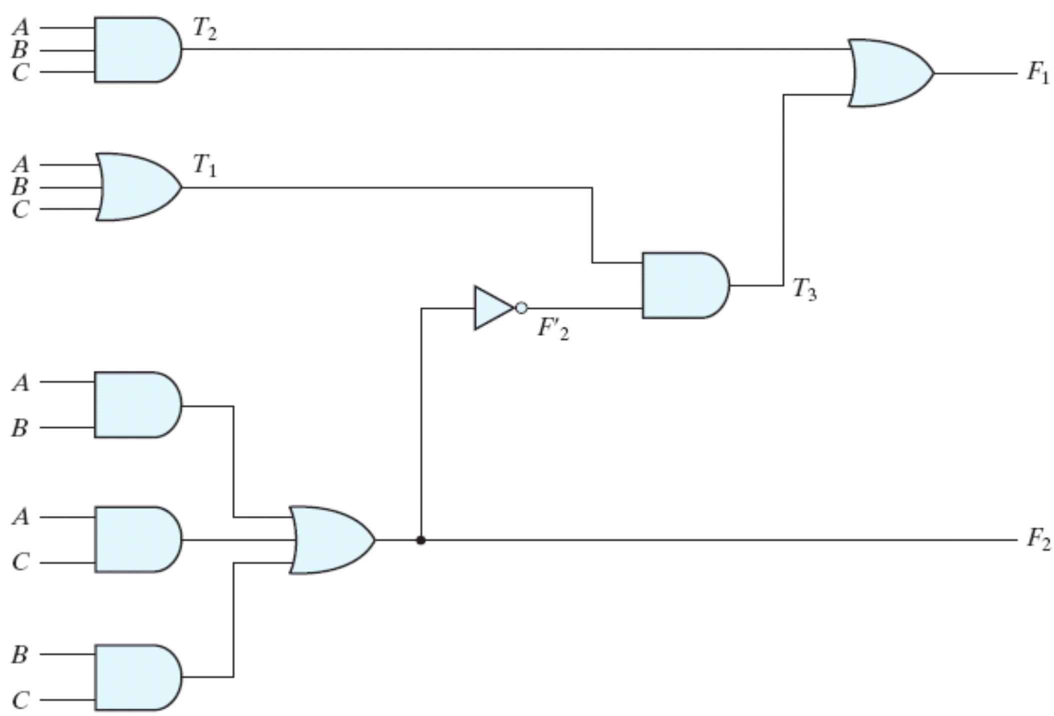
- Tasks:

(1) Analysis of CL: Analyze the behavior of a given logic circuit

(2) Design of CL: Synthesize a circuit that will have a given behavior

Analysis of CL

- Analysis steps
  - Given a CL circuit
  - Obtain a function for each output
  - Derive a truth table for each output
  - Interpret the operation



Analysis of CL

- F2=AB+AC+BC
- F1=A'BC'+A'B'C+AB'C'+ABC
- Interpretation

A	B	C	F_2	F_1
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Design of CL

- Design steps
  - Given the specification of a circuit
  - Determine the inputs and the outputs
  - Derive a truth table for each output
  - Obtain a function for each output
  - Draw the circuit

- Code Conversion
  - BCD to Excess-3

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
	1010	0101	0000	0001
Unused	1011	0110	0001	0010
bit	1100	0111	0010	0011
combi-	1101	1000	1101	1100
nations	1110	1001	1110	1101
	1111	1010	1111	1110

Design of CL

- Code Conversion
  - BCD to Excess-3

Design steps
- **Given the specification of a circuit**
- **Determine the inputs and the outputs**
- **Derive a truth table for each output**
- **Obtain a function for each output**
- **Draw the circuit**

Truth Table for Code-Conversion Example

Input BCD				Output Excess-3 Code			
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

Design of CL

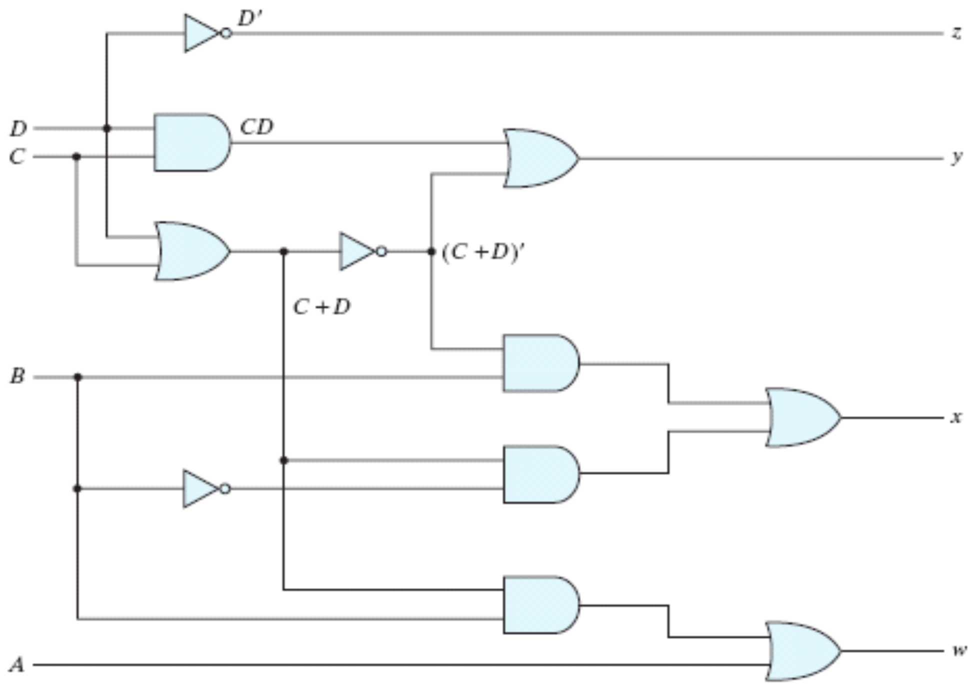
Design steps
- **Given the specification of a circuit**
- **Determine the inputs and the outputs**
- **Derive a truth table for each output**
- **Obtain a function for each output**
- **Draw the circuit**

K-map

- w = A+BC+BD
- x = B'C+B'D+BC'D'
- y = CD+C'D'
- z = D'

Multi-level

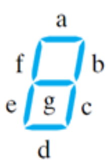
- w = A+B(C+D)
- x = B'(C+D)+B(C+D)'
- y = CD+(C+D)'
- z = D'



Design of CL

- Build a circuit to display a decimal digit(BCD) in a 7-segment component.

Design steps
- **Given the specification of a circuit**
- **Determine the inputs and the outputs**
- **Derive a truth table for each output**
- **Obtain a function for each output**
- **Draw the circuit**



(a) Segment designation


(b) Numeric designation for display

BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0

a=A'C+A'BD+A'B'D' +AB'C'

b = ?

c = ?

. .

. .

. .

g = ?

Outline

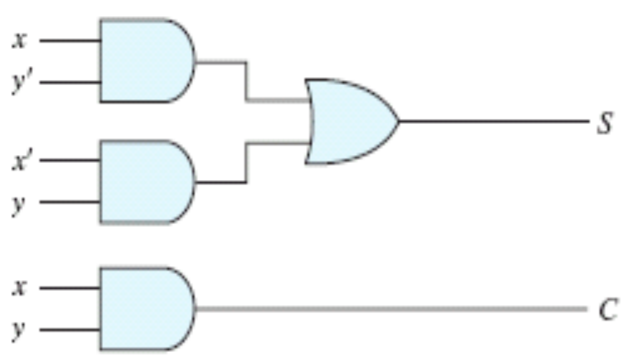
- Analysis of Combinational Logic (CL)
- Design of CL
- Classical CL circuits
- Adder

Half Adder

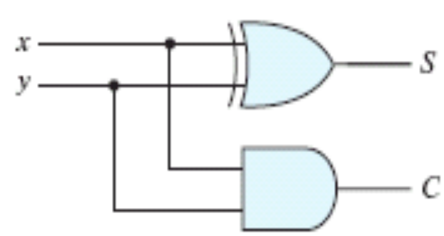
- Addition of two binary bits
- x+y = ?
  - $S=x'y+xy'=x\oplus y$
  - $C=xy$

Half Adder

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



(a) $S = xy' + x'y$
$C = xy$



(b) $S = x \oplus y$
$C = xy$

Full Adder

- Addition of three bits (two significant bits and a previous carry)
- x+y+z= ?

Full Adder

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

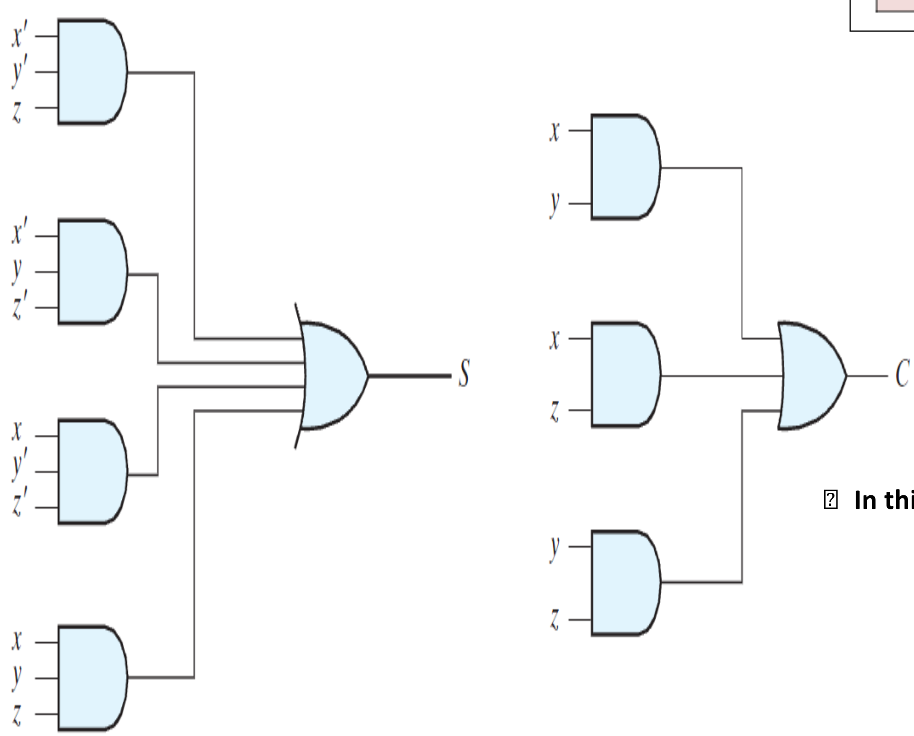
Full Adder

Adds three bits (two significant bits and a previous carry)

- Note that the <u>Sum</u> alternates between an XOR and XNOR depending upon $Carry_{in}$
- Note that the $Carry_{out}$ alternates between an AND and an OR depending upon $Carry_{in}$

- S=x'y'z+x'yz'+xy'z'+xyz
- C=xy+xz+yz

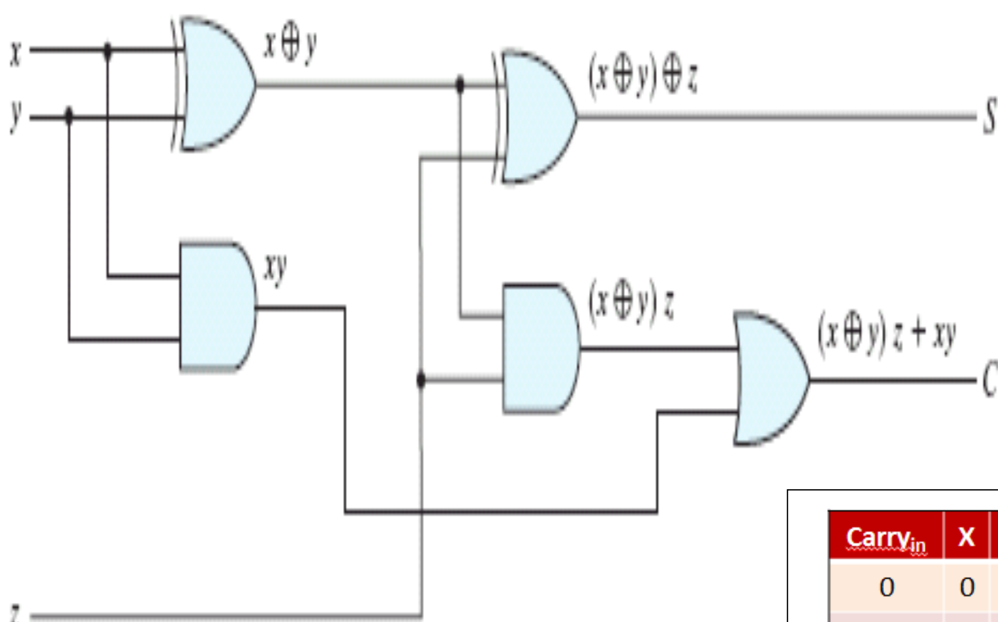
$Carry_{in}$	X	Y	Sum	$Carry_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



In this figure, Z = $Carry_{in}$ and C = $Carry_{out}$

Full Adder

- Implemented with half adder ?
  - S=x'y'z+x'yz'+xy'z'+xyz=z⊕ (x⊕y)
  - C=xy+xz+yz=z(xy'+x'y)+xy



$Carry_{in}$	X	Y	Sum	$Carry_{out}$
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

Recall: For a half – adder:

$S = X'Y + XY' = X \oplus Y$

$C_{in} = XY$

$C_{out} = XY + XC_{in} + YC_{in}$

$= XY(C_{in} + C_{in}') + XC_{in}(Y + Y') + YC_{in}(Z + Z')$

$= C_{in}(X \oplus Y) + XY$

For a Full – Adder:

$S = X'Y'C_{in} + X'YC_{in}' + XY'C_{in}' + XYC_{in}$

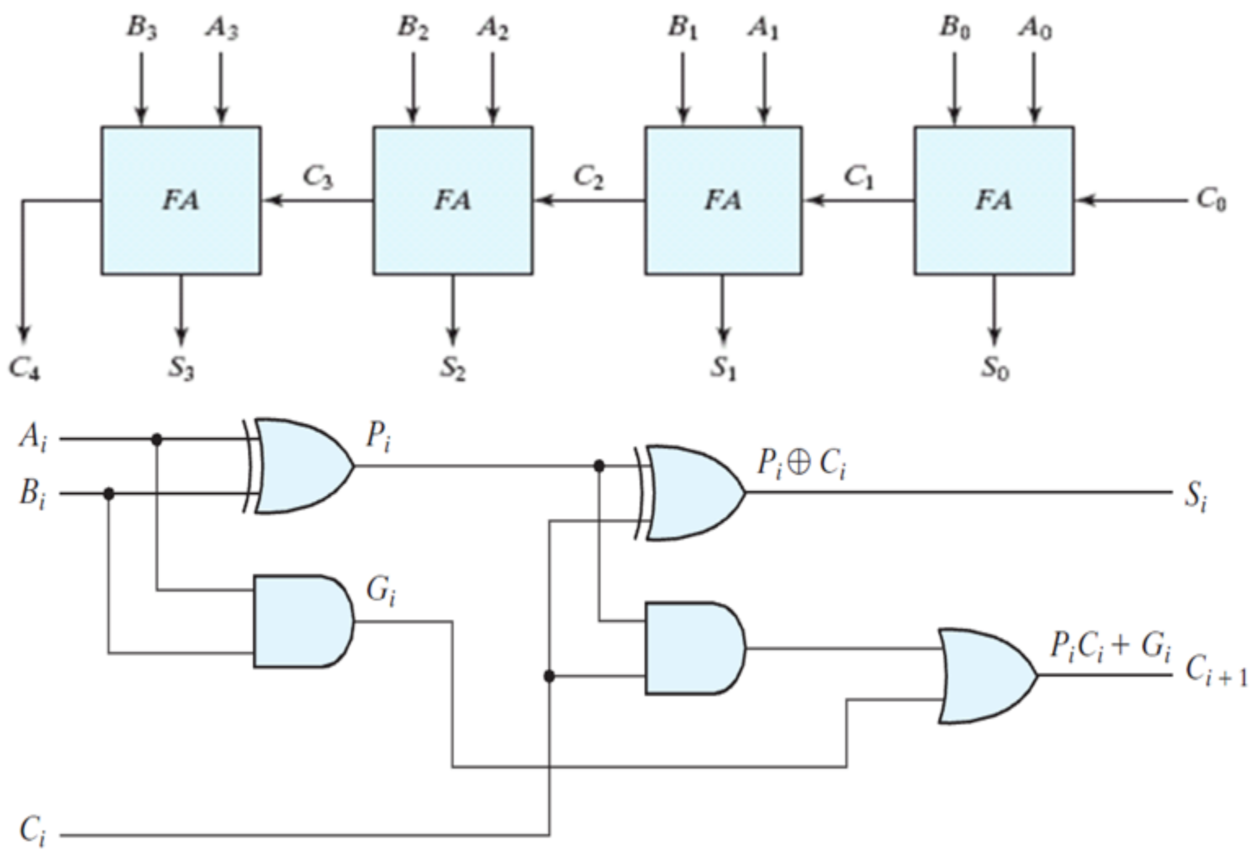
$= (X'Y' + XY)C_{in} + (X'Y + XY')C_{in}'$

$= (X'Y' + XY)C_{in} + (X \oplus Y)C_{in}'$

$= (X \oplus Y) \oplus C_{in}$

Binary Adder

- Arithmetic sum of two binary numbers
- A chain of full adders : ripple (serial) adder



Application of Adder

- Binary multiplier
  - B3B2B1B0 x A2A1A0