Digital Logic

COMP375 Computer Architecture and Organization

“Contrariwise, if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic.”

Lewis Carroll

Schedule

- There will be a quiz in class today
- There will be an exam next week, Wednesday, February 12
- We will review for the exam on Monday

Sum of Product Form

Sum of Product	(A*B) + (C*D)
Product of Sums	(A+B) * (C + D)

Truth Table to Function

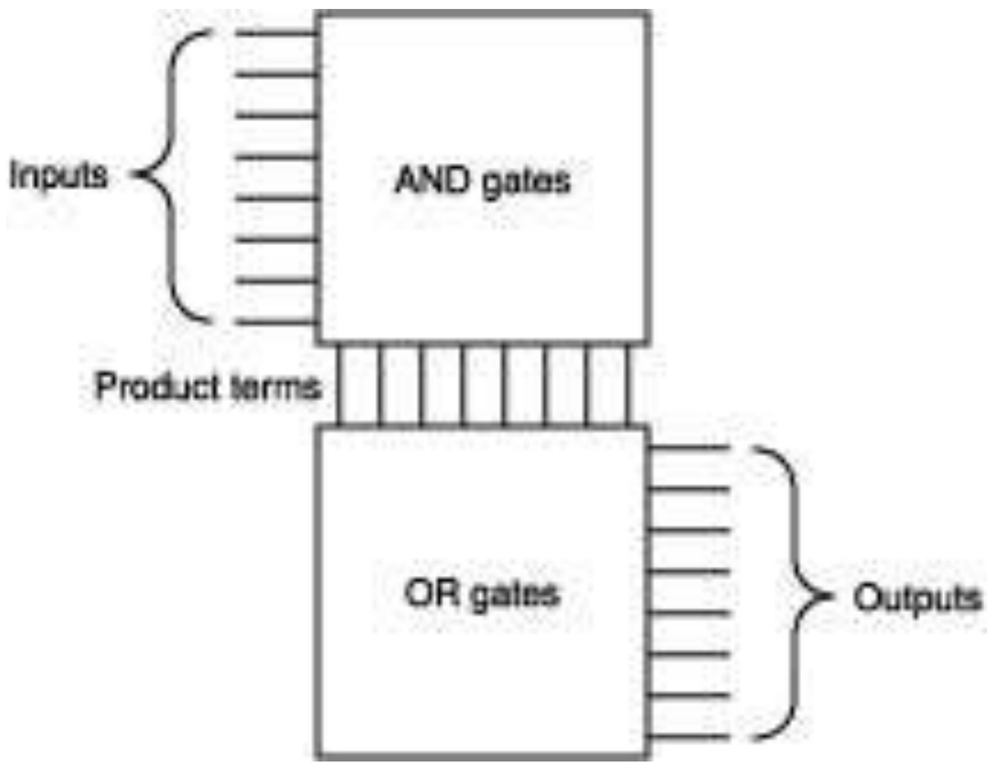
- A sum of products solution can be written by ORing the lines of the truth table that are true

$$F = \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C}$$

A	B	C	**F**
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Programmable Logic Array

- PLAs implement the sum of products



Logic Simplification

- It is frequently possible to simplify a logical expression. This makes it easier to understand and requires fewer gates to implement
- There are several simplification techniques including Boolean algebra and Karnough maps

Karnough maps

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Order of the Columns and Rows

- If you consider the column and row labels and binary logic, they are in Gray Code order

00 01 11 10

or

$\bar{A}\bar{B}$ $\bar{A}B$ AB $A\bar{B}$

or

0 1 3 2

0	1	3	2
4	5	7	6
12	13	15	14
8	9	11	10

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Single Variable Group of 8

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0000	0001	0011	0010
$\overline{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\overline{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0000	0001	0011	0010
$\overline{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\overline{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Two Variable Group of 4

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

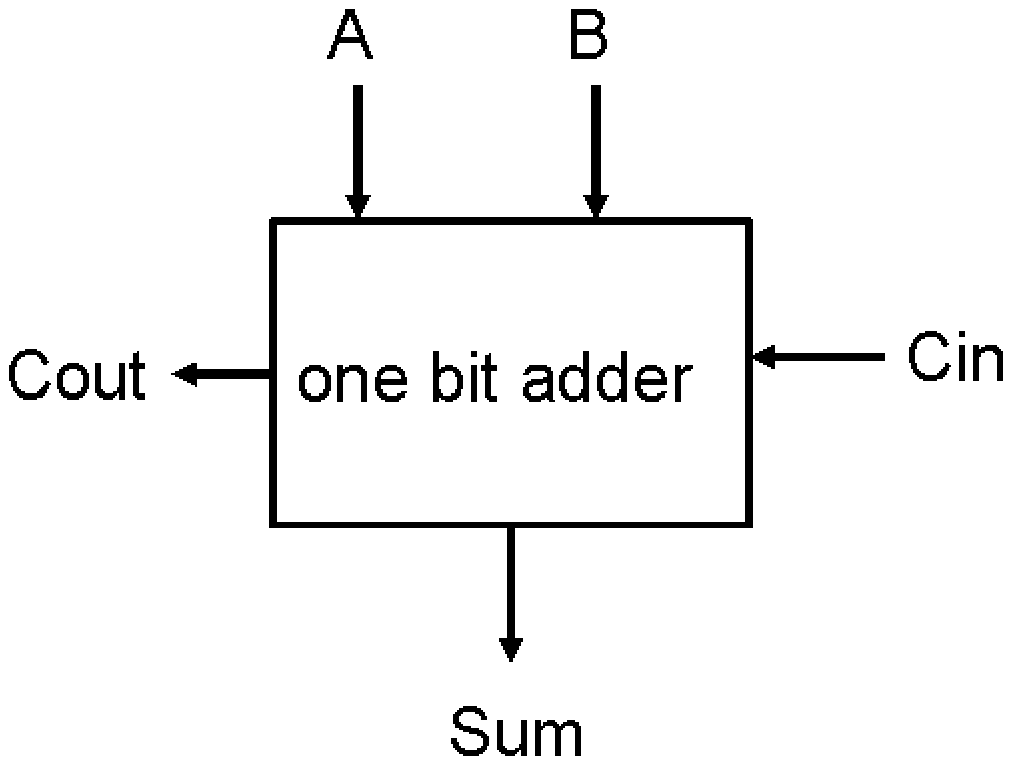
Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Three Variable Pairs

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0000	0001	0011	0010
$\bar{A}B$	0100	0101	0111	0110
AB	1100	1101	1111	1110
$A\bar{B}$	1000	1001	1011	1010

Addition



Cin	A	B	Cout	Sum
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

Addition Sum Karnough Map

	$\bar{X}\bar{Y}$	$X\bar{Y}$	XY	$\bar{X}Y$
$\overline{Cin}$	0	1	0	1
Cin	1	0	1	0

$$Sum = \bar{X}\bar{Y}C_{in} + \bar{X}Y\overline{C}_{in} + XYC_{in} + \bar{X}Y\overline{C}_{in}$$

Addition Carry Karnough Maps

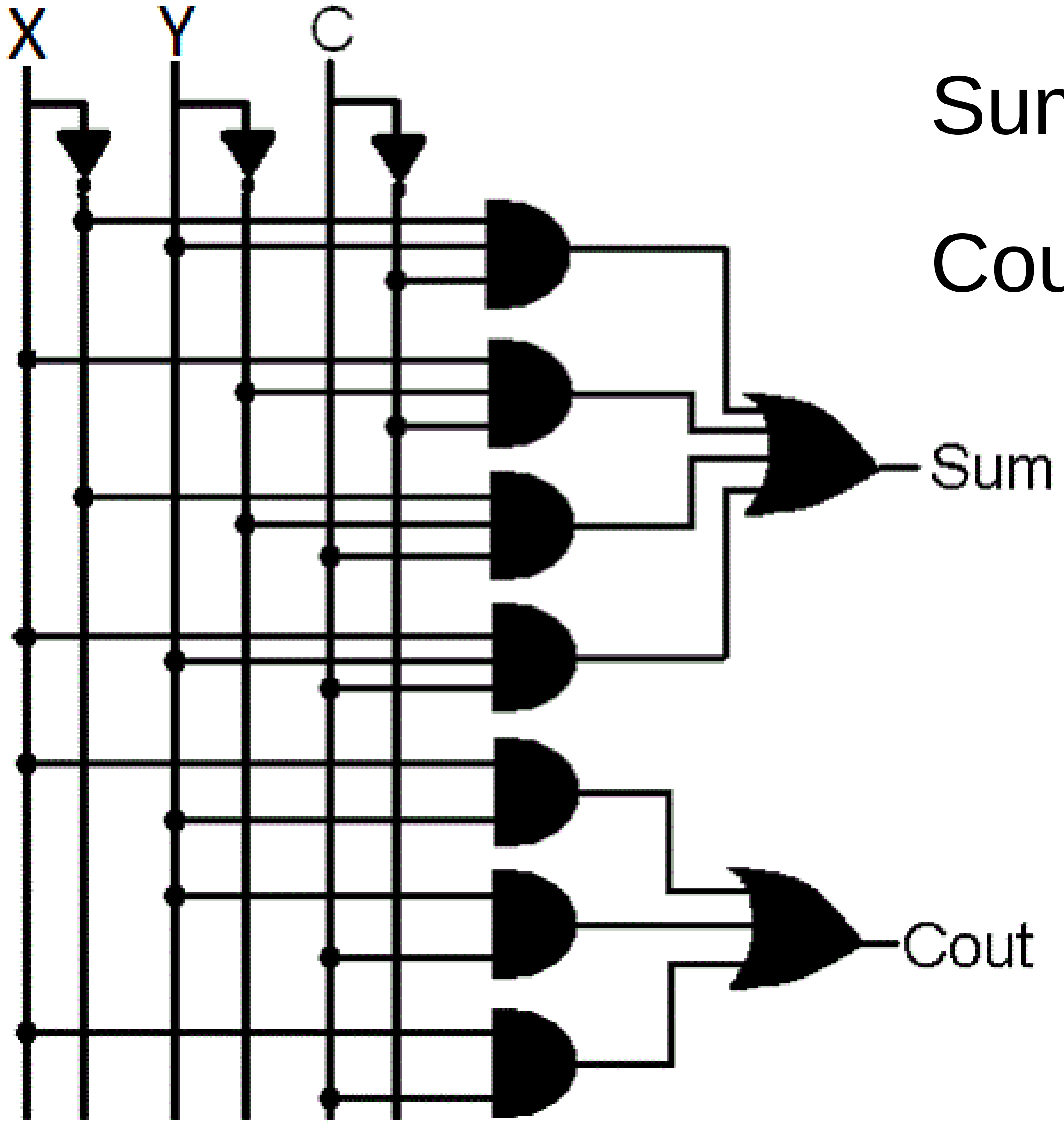
	$\bar{X}\bar{Y}$	$\bar{X}Y$	XY	$X\bar{Y}$
$\overline{C_{in}}$	0	0	1	0
C_{in}	0	1	1	1

$$Cout = YC_{in} + XY + XC_{in}$$

Drawing the Gates

- You can easily convert a logical expression to a gate diagram that will implement the logic
- For Sum of Product form equations, you AND the clauses and then OR them together
- It helps if you start by drawing lines for all the inputs and their complement

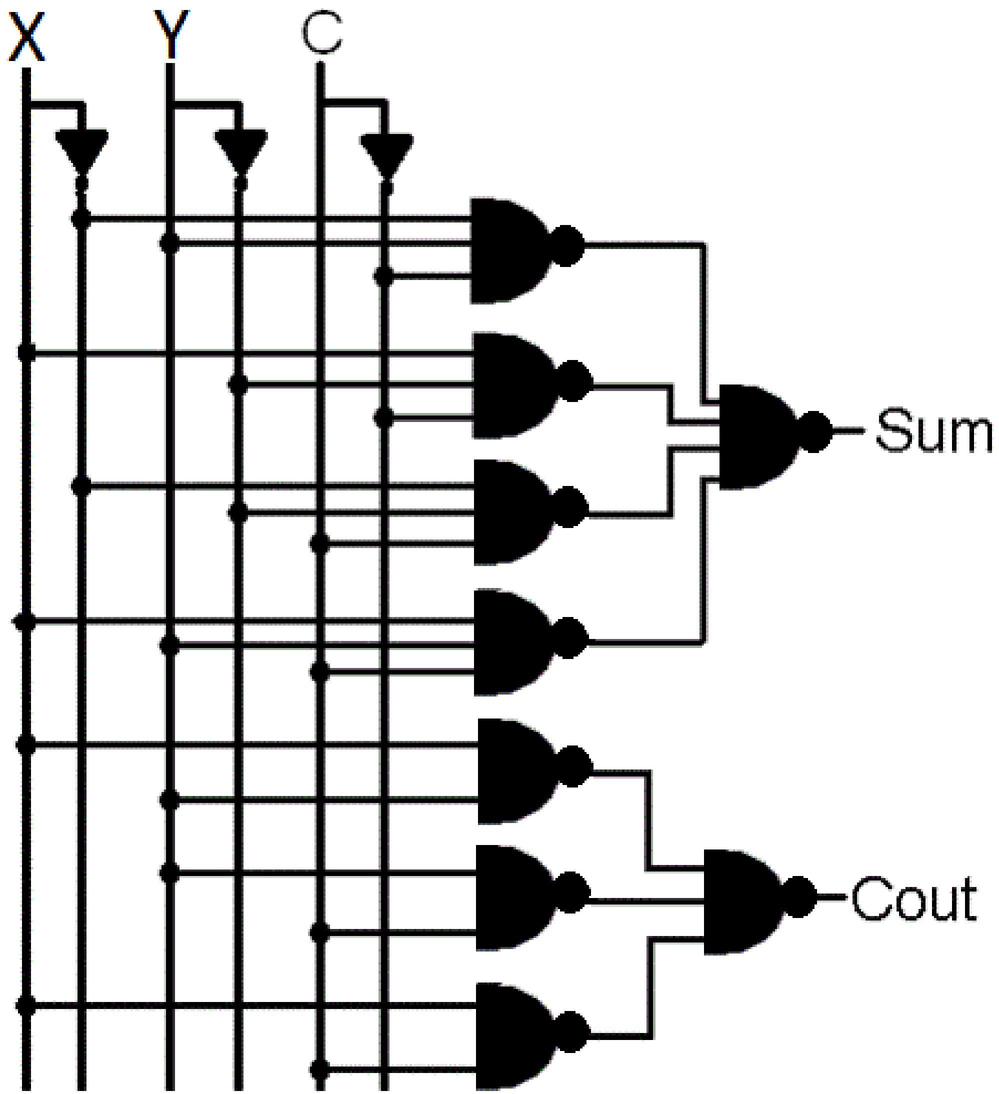
Addition Circuit



$$Sum = \bar{X}\bar{Y}C_{in} + \bar{X}Y\overline{C_{in}} + XYC_{in} + \bar{X}Y\overline{C_{in}}$$

$$Cout = YC_{in} + XY + XC_{in}$$

NAND Addition Circuit



- Instead of AND and OR, you can just use NAND
- DeMorgan's Theorem

$$X + Y = \overline{(\overline{X} * \overline{Y})}$$

- The not of the output of the AND gates goes to a NAND instead or an OR

Karnaugh Maps for Programming

```
if (((S <6)&&(L>10)) ||
    ((S>=6)&&(N==G)&&(L>10)) ||
    ((N==G)&&(L<=10)) ||
    ((S>=6)&&(N==G)) ||
    ((S>=6)&&(N!=G)&&(L>10)) )  {

        print “OK”;
}
```

Fill Table with IF Clauses

S is (S>=6) L is (L>10) N is (N==G)

S' is (S<6) L' is (L<=10) N' is (N!=G)

	S'N'	S' N	S N	S N'
L'	0	1	1	0
L	1	1	1	1

```
if ((N==G) || (L>10))
```

Schedule

- There will be a quiz in class today
- There will be an exam next week, Wednesday, February 12
- We will review for the exam on Monday