REMINDER

Midterm Oct. 3, 3:10-4:03 PM

Closed Book, Closed Notes, Bring Calculator, Paper Provided

Last Name A-K 2040 Valley LSB; Last Name L-Z in 10 Evans

Old Exams Are Posted on Web

Review Session 5-6:30 Tu 2060 Valley LSB

EE 43 Labs Are Not Cancelled:

Students in Tu 6-8PM should go to a different section during 6th week.

Professor Neureuther will not be available Wed. Oct 3rd-Fri Oct. 5th due to a Conference

Lecture 10: October 1, 2001

Logic Implementation and Synthesis

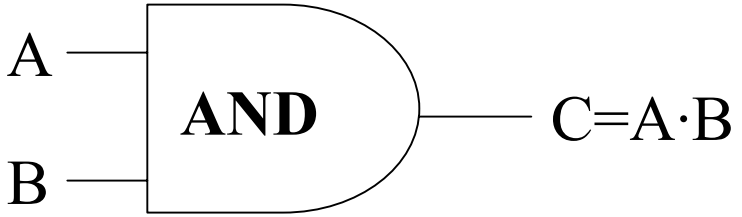
A) Logic Levels and Gate Circuits
B) Combination of Logic Functions
C) Synthesis from a Truth Table
D) NAND Gate Synthesis
E) XOR and Introduction to Timing

The following slides were derived from those prepared by Professor Oldham For EE 40 in Fall 01

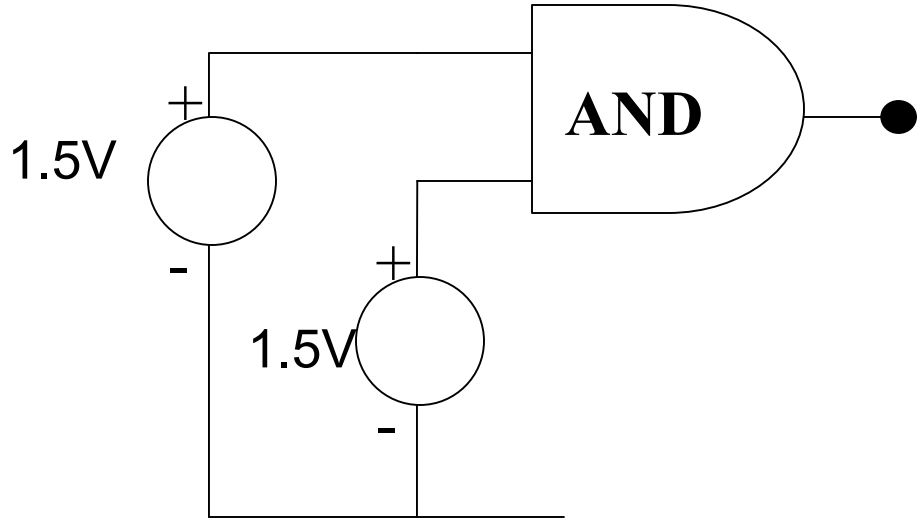
Reading:

Schwarz and Oldham 11.2-11.3 pp. 403-422

Logic Gates – How are they used in practice?



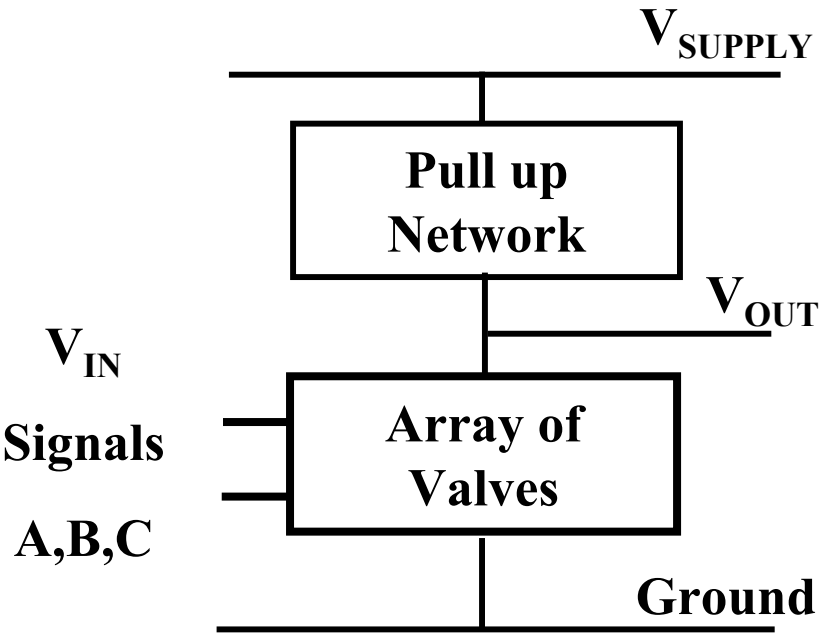
First of all we must agree on what is High (logical **1**) or low (logical **0**). Suppose 1.5 V is **1** and 0V is logical **0**.



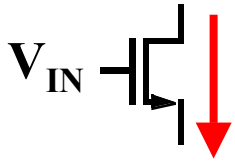
C would have the value of 1.5 V (logical 1).

But it would have the value of 0V logical 0 if either one of the inputs were held at zero V.

Logic Gates – How are they built in practice?

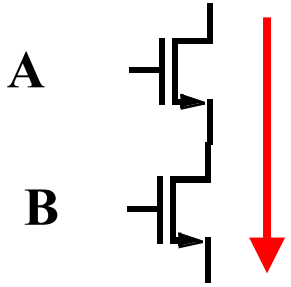


A Valve is a Transistor

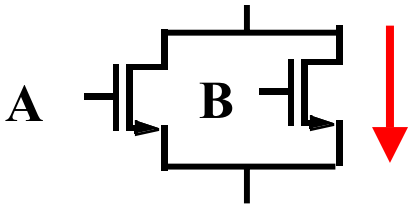


Current flows when V_{IN} is high

Valves in Series => NAND



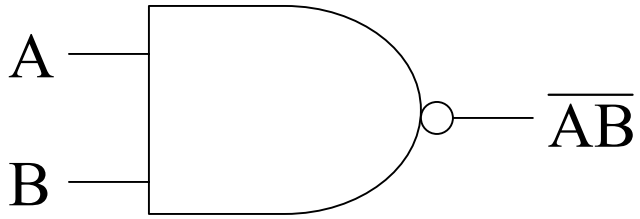
Valves in Parallel => NOR



(You can learn about building gates in EE 141.)

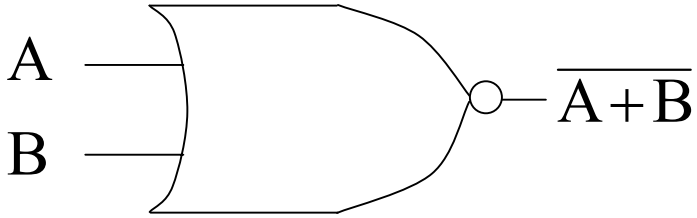
The most common basic gates are NAND and NOR?

Not-AND = NAND



A	B	AB	$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Not-OR = NOR

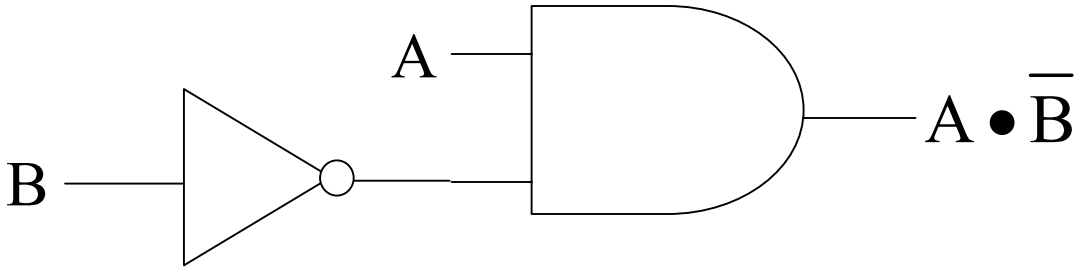


A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

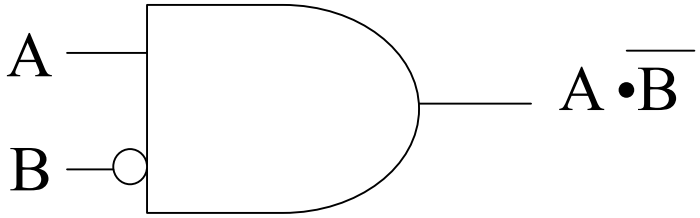
How to Combine Gate to Produce a Desired Logic Function? (More basic Logical Synthesis)

Example

$F = A \bullet \overline{B}$



Again a little shorthand is useful



How to Combine Gate to Produce a Desired Logic Function? (More basic Logical Synthesis)

Suppose we are given a truth table (all logic statements can be represented by a truth table). How can we implement the function?

Answer: There are lots of ways, but one simple way is implementation from "sum of products" formulation.

How to do this: 1) Write sum of products expression from truth table and 2) Implement using standard gates.

(Warning this is probably inefficient – we need to minimize, or simplify the expression. You will learn this in CS 150.)

How to Combine Gate to Produce a Desired Logic Function? (More basic Logical Synthesis)

Example:

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

Clearly: F= 1 if

$$\overline{A}\,\overline{B}\,C = 1$$

or

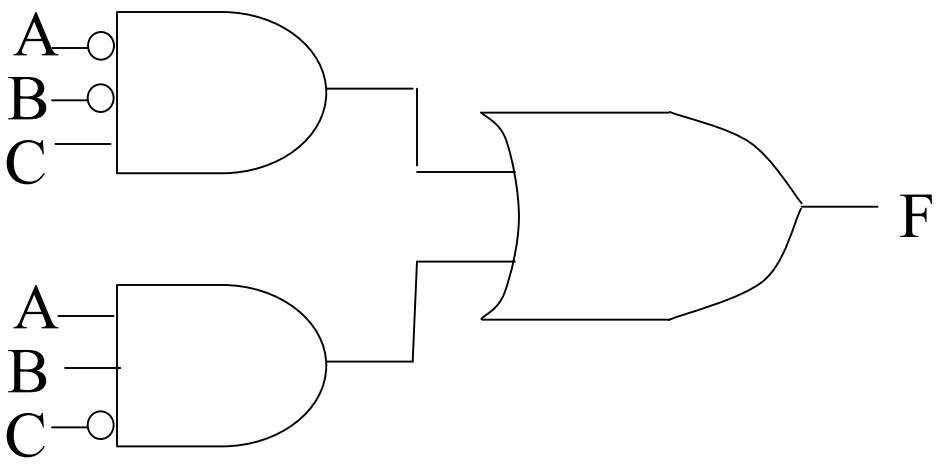
$$AB\overline{C} = 1$$

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

How to Combine Gate to Produce a Desired Logic Function? (More basic Logical Synthesis)

Example:

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



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

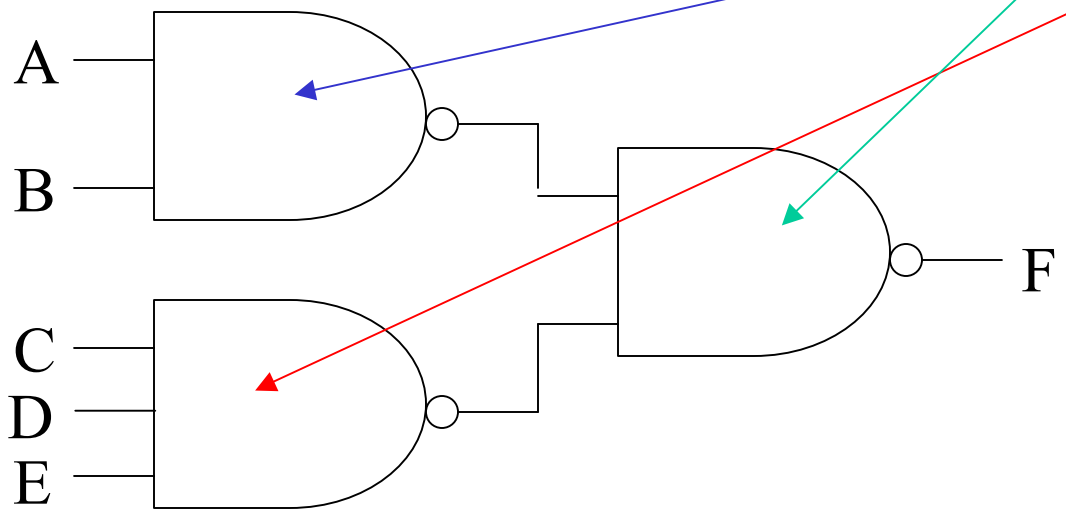
Logical Synthesis
Guided by DeMorgan's Theorem

DeMorgan's Theorem :

$$A + B + C = \overline{\left[\overline{A}\,\overline{B}\,\overline{C}\right]} \quad \text{or} \quad \overline{A} + \overline{B} + \overline{C} = \overline{\left[A\,B\,C\right]}$$

Example of Using DeMorgan's Theorem:

$$F = A \bullet B + C \bullet D \bullet E = \overline{\overline{AB} \bullet \overline{CDE}}$$



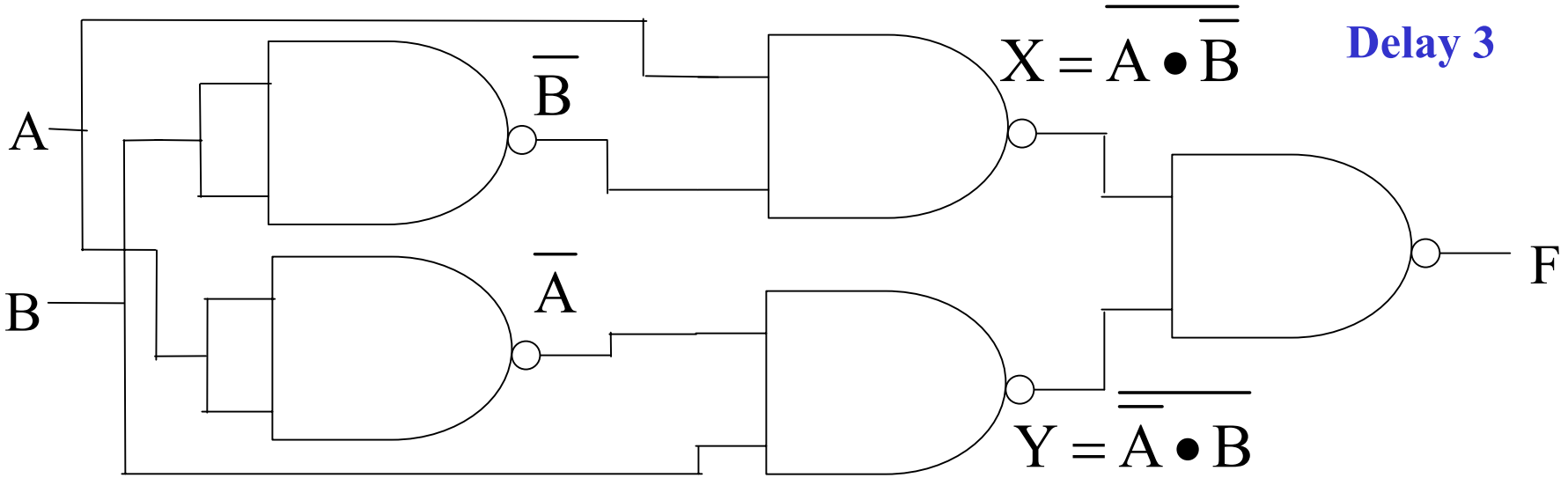
Thus any sum of products expression can be immediately synthesized from NAND gates alone

Logical Synthesis of XOR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

$$F = A \bullet \overline{B} + \overline{A} \bullet B$$

We Need a Timing Diagram!



Timing Diagram for Delays in Logic