Version Date 9/25/01

Lecture 9: September 26th, 2001

Digital Signals and Basic Logic Blocks

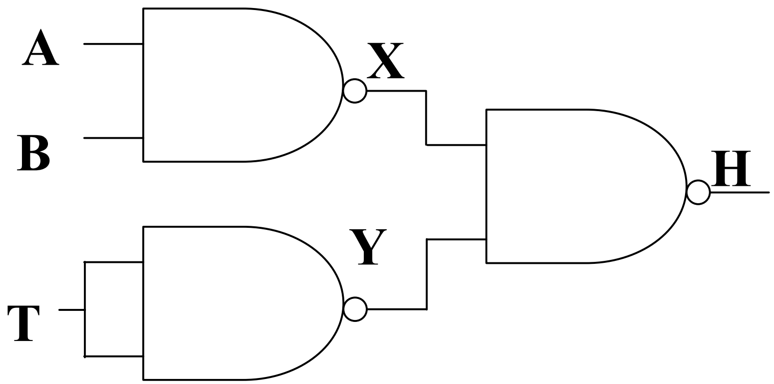
A) Advantages of Digital
B) Goals: Gates ⇔ Logical Functions
C) Truth Tables and Logical Functions
D) Boolean Operations and Gates

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

Reading:
Schwarz and Oldham 11.1, 11.2 pp. 92-402

GOAL FOR LECTURE 9

Given Gates



Find the Truth Table and the Boolean Logic Function

A	B	T		H
0	0	0		0
0	0	1		1
0	1	0		0
0	1	1		1
1	0	0		
1	0	1		
1	1	0		
1	1	1		

$H = (A \cdot B) + T$

Evaluation of Logical Expressions with "Truth Tables"

Suppose we have a heater which we want to operate any time anytime switch T is "on" or if both switches A and B are "on" .

We would say **H** is true if **T** is True or **A** and **B** are both true.

Or

We could say **H** is 1 if either **T** is 1 or **A** and **B** are both 1.

A "Truth Table" is a simple table listing all possible combinations of **A, B, T** and the resulting value of **H**.

A	B	T		H
0	0	0		?
etc				
etc				

Evaluation of Logical Expressions with “Truth Tables”

H is **1** if either **T** is **1** or **A** and **B** are both **1**.

Truth Table for Heater Algorithm

A	B	T	H
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Logical Expressions to express Truth Tables

We need a notation for logic expressions.

Standard logic notation and logic gates:

AND: "dot" Examples: $\mathbf{X = A \cdot B}$; $\mathbf{Y = A \cdot B \cdot C}$

OR : "**+** sign" Examples: $\mathbf{W = A+B}$; $\mathbf{Z = A+B+C}$

NOT: "bar over symbol for complement" Example: $\mathbf{Z = \overline{A}}$

With these basic operations we can construct any logical expression.

Digital Heater Control Example (cont.)

Logical Expression : To create logical values we will define a closed switch as "True", ie Boolean **1** (and thus an open switch as **0**).

Heater is on (**H**=**1**) if (**A and B) or T** is **1**

- Logical Statement: **H = 1** if **A and B** are **1 or T** is 1.

- Remember we use "dot" to designate logical "and" and "**+**" to designate logical or in switching algebra. So how can we express this as a Boolean Expression?

- Boolean Expression: $H = (A \cdot B) + T$

The Important Logical Functions

The most frequent (i.e. important) logical functions are implemented as electronic "building blocks" or "gates".

We already know about **AND** , **OR** and **NOT** What are some others:

Combination of above: inverted AND = **NAND,**
inverted OR **= NOR**

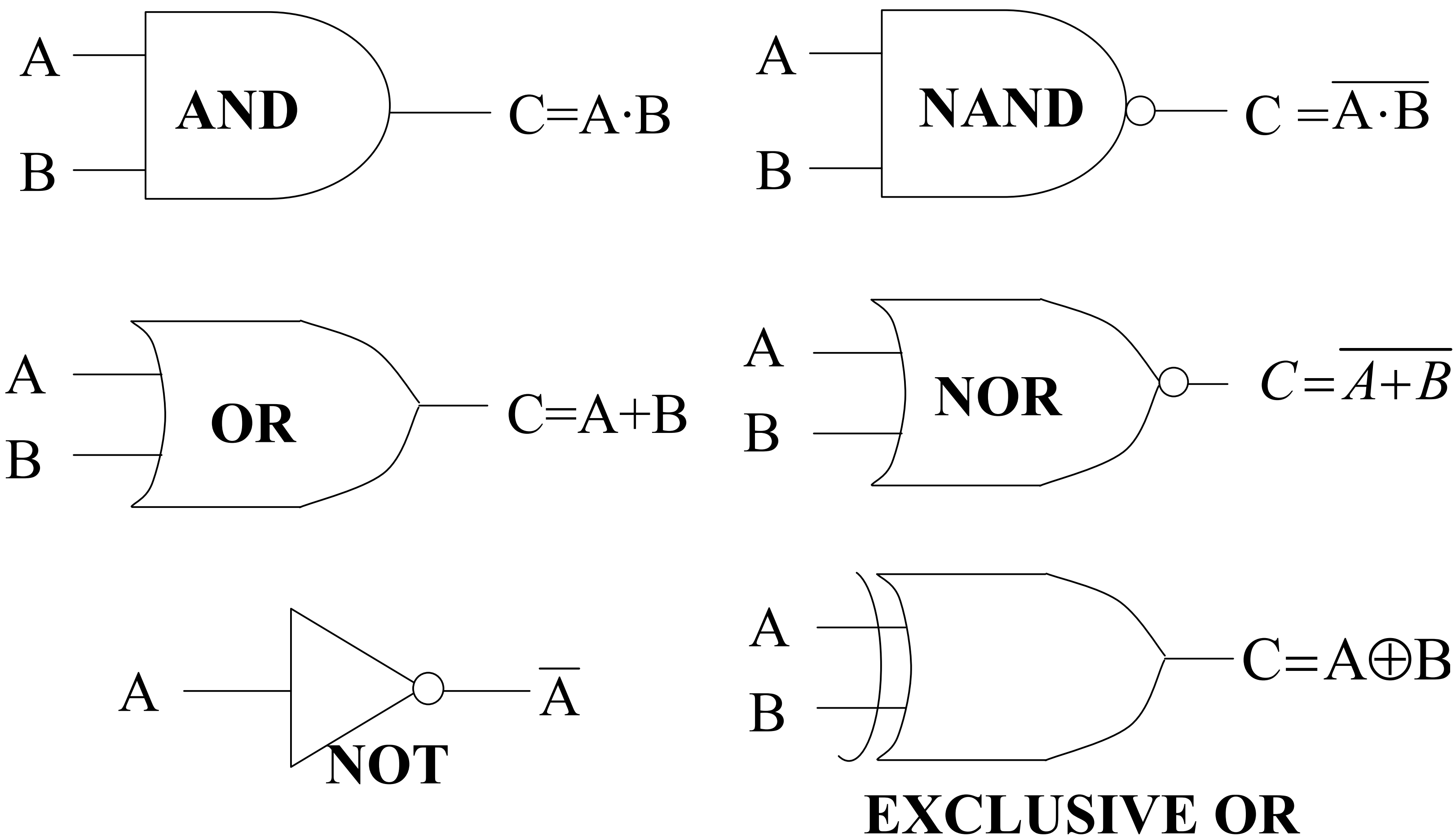
And one other basic function is often used: the "EXCLUSIVE OR" … which logically is "or except not and"

An EXCLUSIVE OR circuit is need in adding two binary bits to decide if the result bit is a one.

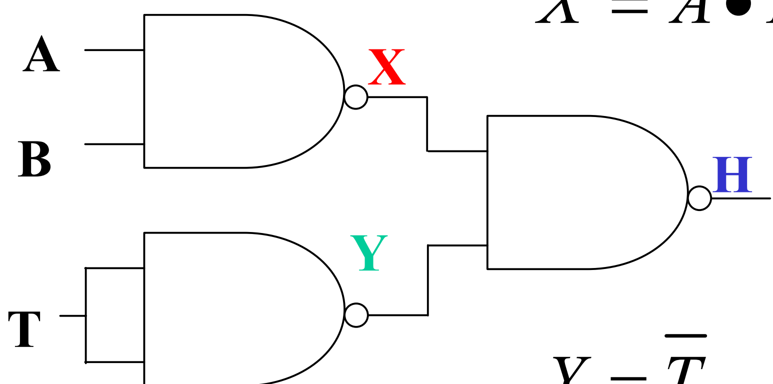
Some Important Logical Functions

- **"AND"** $\mathrm{A \cdot B} \quad (\mathrm{or\ A \cdot B \cdot C})$
- **"OR"** $\mathrm{A+B} \quad (\mathrm{o}r\ A+B+C+D\ldots)$
- **"INVERT" or "NOT"** not A or $\overline{\mathrm{A}}$
- **"not AND" = NAND** $\overline{\mathrm{AB}} \quad (\mathrm{o}n\mathrm{ly\ 0\ when\ } A \mathrm{\ and\ } B = 1)$
- **"not OR" = NOR** $\overline{\mathrm{A+B}} \quad (\mathrm{only\ 1\ when\ A = B = 0})$
- **exclusive OR = XOR** $\mathrm{A \oplus B}$ (only 1 when A, B differ)
  i.e., $\mathrm{A+B}$ **except** $\mathrm{A \cdot B}$

SYMBOLS FOR LOGIC GATES



EXAMPLE: LOGIC GATES TO LOGIC FUNCTION



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

$$Y = \overline{T}$$

$$H = \overline{X \bullet Y}$$

De Morgan's Law

$$H = \overline{X} + \overline{Y}$$

$$H = \overline{\overline{A \bullet B}} + \overline{\overline{T}}$$

$$H = A \bullet B + T$$

A	B	T	X	Y	H
0	0	0	1	1	0
0	0	1	1	0	1
0	1	0	1	1	0
0	1	1	1	0	1

There are 4 more cases