

Homework

- Reading
 - Tokheim, Chapter 3, 4, and 6.1 - 6.3
 - Free Digital Logic Simulator website
- Machine Projects
 - Continue on mp3
- Labs
 - Continue in labs with your assigned section

Digital Logic

- Two types of digital logic design
 - Combinational Logic – has no memory elements
 - Sequential Logic – contains memory elements
- Combinational logic design is sometimes called Boolean algebra after George Boole
 - Based on binary logic
 - Uses AND, OR, XOR, NOT, etc.
 - Develop truth tables and implement a design

Boolean Algebra

- Variables / Expressions

Two Values Only (0 or 1)

- Basic Operators

AND *

OR +

XOR ⊕

NOT Bar over the variable or expression

Or # after variable name

Boolean Algebra

- Precedence of Operators

NOT (like unary minus)

AND (like multiply)

Division None

OR/XOR (like add)

Subtraction None

- Parentheses to force precedence

$A * (B + C)$ is not the same as $A * B + C$

Boolean Algebra

- Multiplicative Identities

$$A * 0 = 0$$

$$A * \underline{A} = A \quad (\text{Note: Not } A \text{ squared})$$

$$A * \bar{A} = 0$$

- Additive Identities

$$A + 1 = 1$$

$$A + \underline{A} = A \quad (\text{Note: Not } 2A)$$

$$A + \bar{A} = 1$$

Boolean Algebra

- Negative Identity

$$\overline{\overline{A}} = A$$

- Commutative Property

$$A * B = B * A$$

$$A + B = B + A$$

- Distributive Property

$$A (B + C) = A * B + A * C$$

Boolean Algebra

- Common Reductions of Sums of Products (Also called a “Minterm” expression)

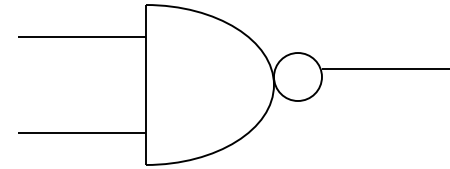
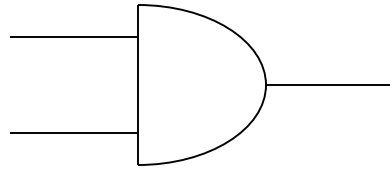
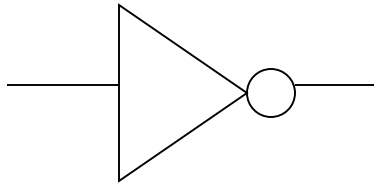
$$A * \overline{B} + \overline{A} * B + A * B = A + B \quad \text{OR}$$

$$A * \overline{B} + \overline{A} * B = A \oplus B \quad \text{XOR}$$

- Common Reductions of Products of Sums (Also called a “Maxterm” expression)

$$(A + B) * (\overline{A} + B) * (A + \overline{B}) = A * B \quad \text{AND}$$

Binary Logic Symbols/Tables

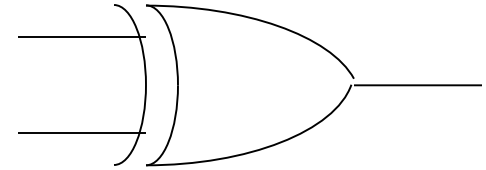
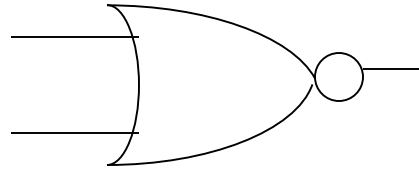
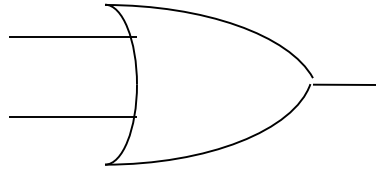


NOT	
0	1
1	0

AND	0	1
0	0	0
1	0	1

NAND	0	1
0	1	1
1	1	0

Binary Logic Symbols/Tables



OR	0	1
0	0	1
1	1	1

NOR	0	1
0	1	0
1	0	0

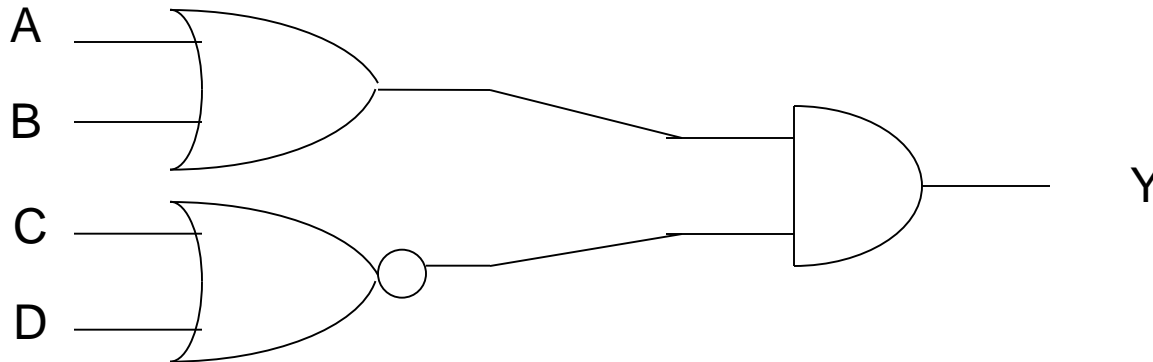
XOR	0	1
0	0	1
1	1	0

Boolean Algebra / Logic Diagram

- Sample Boolean Equation

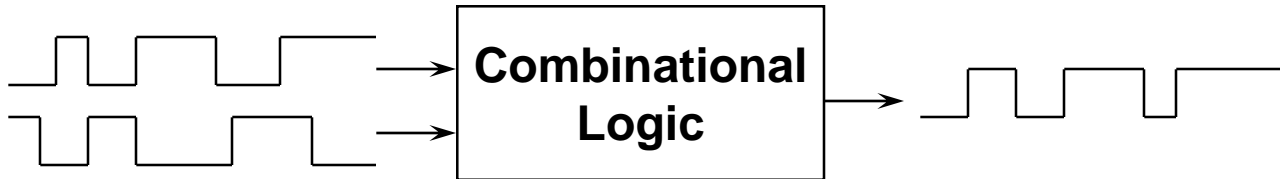
$$Y = (A + B) * \overline{(C + D)}$$

- Equivalent Logic Diagram



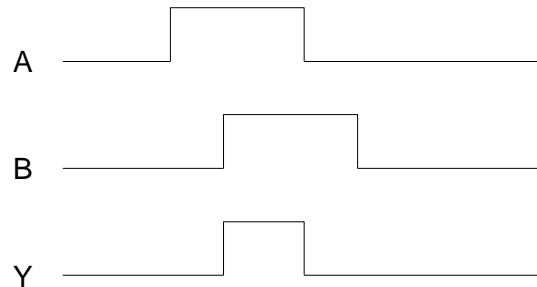
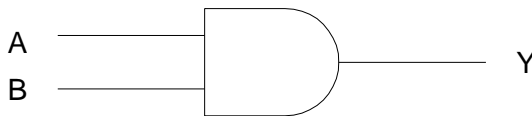
Timing Considerations

- Can look at pulse trains through combinational logic over time rather than just constant inputs



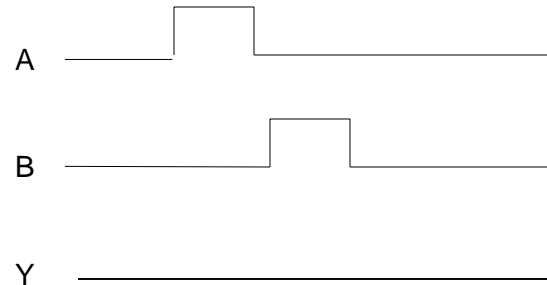
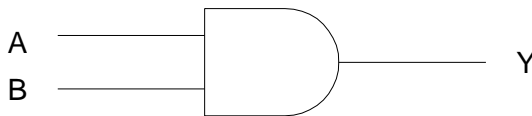
Timing Considerations

- Things don't happen instantaneously
 - Each signal arrives at some unique time
 - Gate logic takes some time to react
 - Changes in the output appear some time after changes to the inputs (at nanosecond level)
- Example, an AND gate



Timing Considerations

- Things don't happen instantaneously
 - As clock speed increases, the “skew” due to differential delays narrows output pulses
 - Beyond the designed maximum clock speed, the circuit may fail
- Example, an AND gate



Free Logic Design Tool

- Practical software design tools such as VHDL are now being used in industry instead of building and testing physical “breadboards”
- VHDL is complex and has a learning curve
- Logic Gate Simulator is a simple design tool
- Allows us to design and test simple hardware logic via software building and simulation
- Download:
<https://www.kolls.net/gatesim/>

Free Logic Design Tool

The screenshot displays the Logic Gate Simulator software interface. The title bar reads "[Untitled] (Modified) - Logic Gate Simulator". The interface includes a toolbar with various icons for file operations and simulation. On the left, a component palette is visible, containing sections for "Compound Gates" and "I/O Gates". The main workspace shows a logic circuit with five horizontal rows of components:

- Row 1: A red square input connected to a NOT gate, which is connected to a yellow circle output.
- Row 2: Two red square inputs connected to an AND gate, which is connected to a yellow circle output.
- Row 3: A red square input and a yellow square input connected to an OR gate, which is connected to a red circle output.
- Row 4: Two red square inputs connected to an XOR gate, which is connected to a red circle output.
- Row 5: A red square input and a yellow square input connected to an OR gate, which is connected to a red circle output.

The status bar at the bottom of the window contains the text: "Logic Gate Simulator 1.4.4182.31395 Copyright © 2011 Steve Kollmansberger et al. <http://www.kolls.net/gatesim>". The Windows taskbar at the very bottom shows the search bar and system tray icons, including the time "5:39 PM" and date "5/23/2016".

Free Logic Design Tool

- Demonstration in Class
 - Button
 - Indicator
 - Inverter
 - And Gate
 - Or Gate
 - Exclusive-Or Gate