CSSE2010/CSSE7201
Lecture 2

Intro to Logic Gates

School of Information Technology and Electrical Engineering
The University of Queensland

Today…

- Introduction to Logic Gates
- Logic Diagrams
- Boolean Algebra and Logic Expressions
- There will be several polling questions
  - URL: responsewaresg.net
  - Session ID: csse2010s2

Learning Lab Sessions

- Slides used will be made available
  - After the last session that week
- Only attend the session you are signed-up to
  - Contact eait.mytimetable@uq.edu.au if you have sign-on issues
- If specific preparation is required, you'll get told, by default you should review previous lectures
- Make sure you attend and complete the learning labs for each week

Digital Logic

- **Digital circuits**
  - Only two logical levels present (i.e. binary)
    - Logic '0' – usually small voltage (e.g. around 0 volts)
    - Logic '1' – usually larger voltage (e.g. 0.8 to 5 volts, depending on the "logic family", i.e. type/size of transistors)
- **Logic gates**
  - are the building blocks of computers;
  - Each gate has
    - one or more inputs
    - exactly one output
  - perform logic operations (or functions)
    - 7 basic types: **NOT**, **AND**, **OR**, **NAND**, **NOR**, **XOR**, **XNOR**
    - Inputs & outputs can have only two states, 1 & 0 - can be called "true" & "false"
    - **Logic symbol**, **Truth table, Boolean expression, Timing diagram**

Recall – Levels of Abstraction

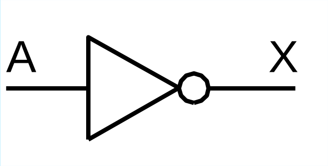
Level 5	**Problem-oriented language level**
Level 4	**Assembly language level**
Level 3	**Operating system machine level**
Level 2	**Instruction set architecture level**
Level 1	**Microarchitecture level**
Level 0	**Digital Logic Level**

Basic Logic Gates

Logic Symbol ⇨

Truth table ⇨

NOT gate

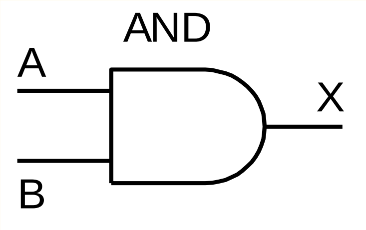


A	X
0	1
1	0

Inverts the input

Also called "inverter"

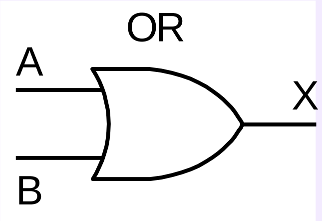
AND gate



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Output is HIGH when all the inputs are HIGH

OR gate



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

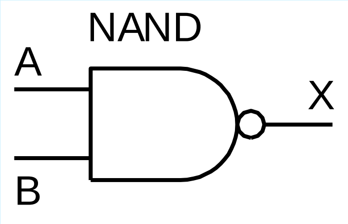
Output is HIGH when at least one input is HIGH

Basic Logic Gates (cont…)

Logic Symbol ⇨

Truth table ⇨

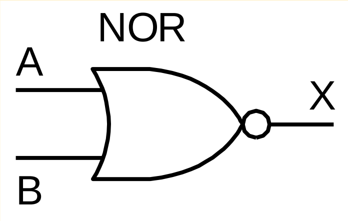
NAND gate



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

Output is HIGH when at least one input is LOW

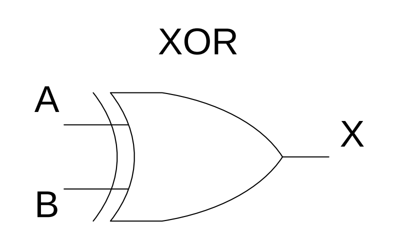
NOR gate



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Output is HIGH when all the inputs are LOW

XOR gate



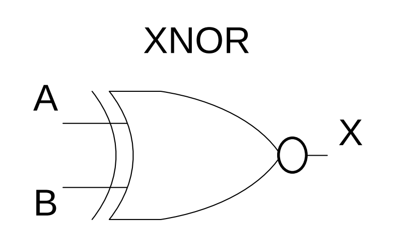
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Output is HIGH when exactly one input is HIGH

Basic Logic Gates (cont…)

XNOR gate

Logic Symbol



Truth table

A	B	X
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

Output is HIGH when the inputs are the same

NOT

AND

OR

NAND

NOR

XOR

XNOR

Useful to remember:

XOR is the odd function and XNOR is the even function

What’s the truth table for a 3-input NAND gate

1.

A	B	C	X
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

2.

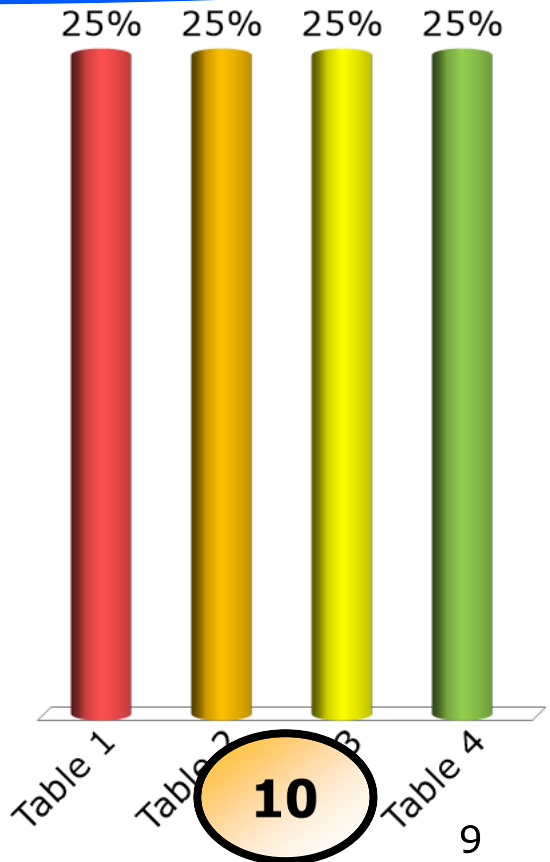
A	B	C	X
0	0	0	**1**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**0**

3.

A	B	C	X
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

4.

A	B	C	X
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**



What's the truth table for a 3-input XOR gate

1.

A	B	C	X
0	0	0	**1**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

2.

A	B	C	X
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

3.

A	B	C	X
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

4.

A	B	C	X
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

9

Boolean Logic Functions

- Logic functions can be expressed as expressions involving:
  - variables (literals), e.g. $A \; B \; X$
  - functions, e.g. $+ \;\; . \;\; \oplus \;\; \overline{}$
- Rules about how this works called **Boolean algebra**
- Variables and functions can only take on values **0** or **1**

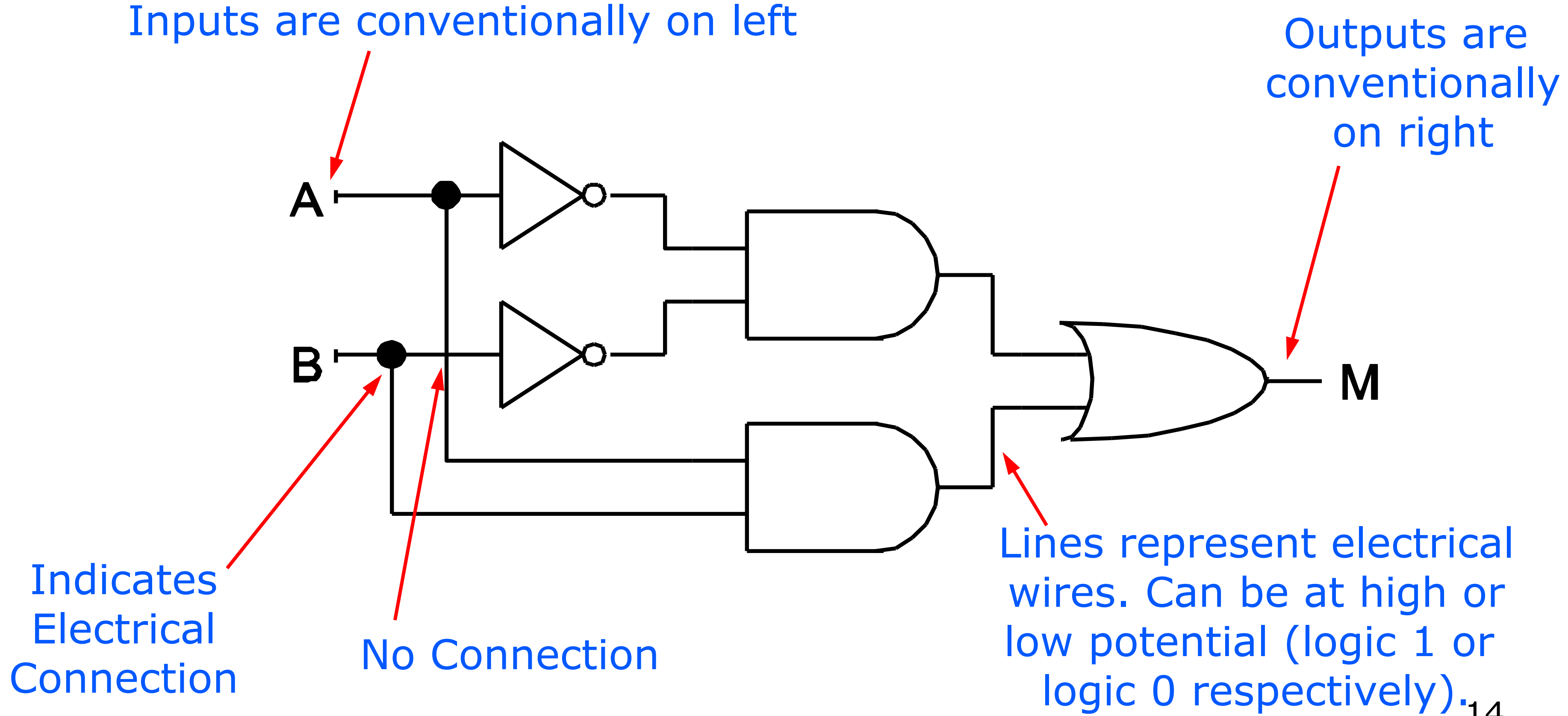
Boolean Algebra conventions

- Conventions we'll use:
  - **Inversion**: ‾ (overline)
    - e.g. **NOT(A) = $\bar{A}$ (pronounced as A bar)**
  - **AND**: dot(.) or implied (by adjacency)
    - e.g. **AND(A,B) = AB = A.B**
  - **OR**: plus sign
    - e.g. **OR(A,B,C) = A+B+C**
- Other examples:
  - XOR(A,B) = $A \oplus B = \overline{A}B + A\overline{B}$
  - NAND(A,B,C) = $\overline{ABC}$
  - NOR(A,B) = $\overline{A + B}$

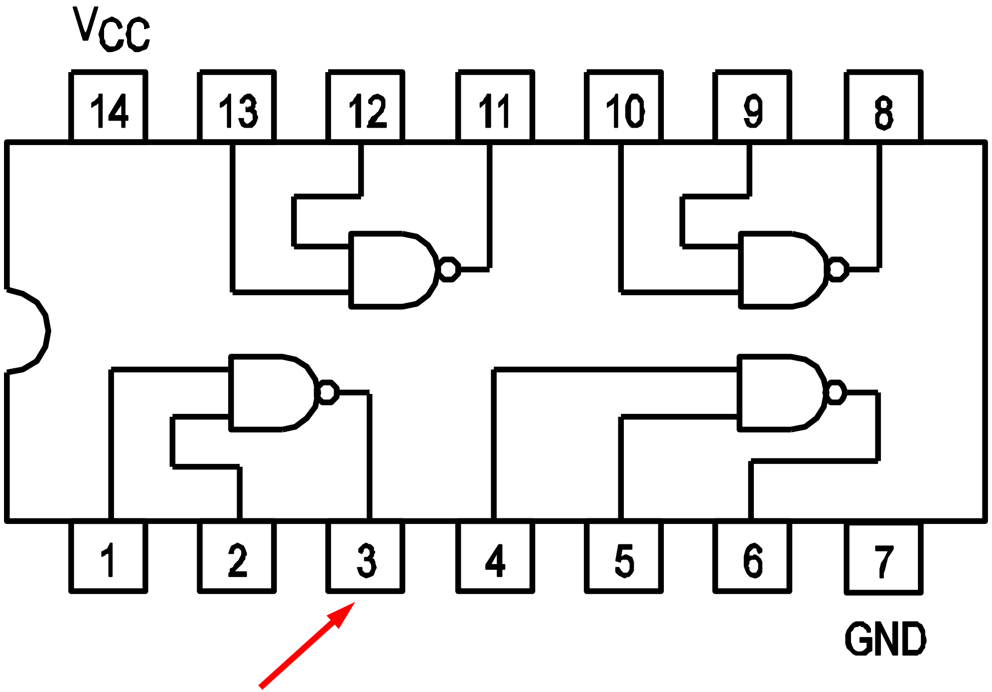
Summary of Logic Function Representations

- There are four representations of logic functions (assume function of n inputs)
  - **Truth table**
    - Lists output for all 2^n combinations of inputs
      - Best to list inputs in a systematic way
  - **Boolean function** (or **equation**)
    - Describes the conditions under which the function output is 1
  - **Logic Diagram**
    - Combination of logic symbols joined by wires
  - **Timing Diagram**

Logic Diagram Conventions



Gates on Integrated Circuits (ICs)

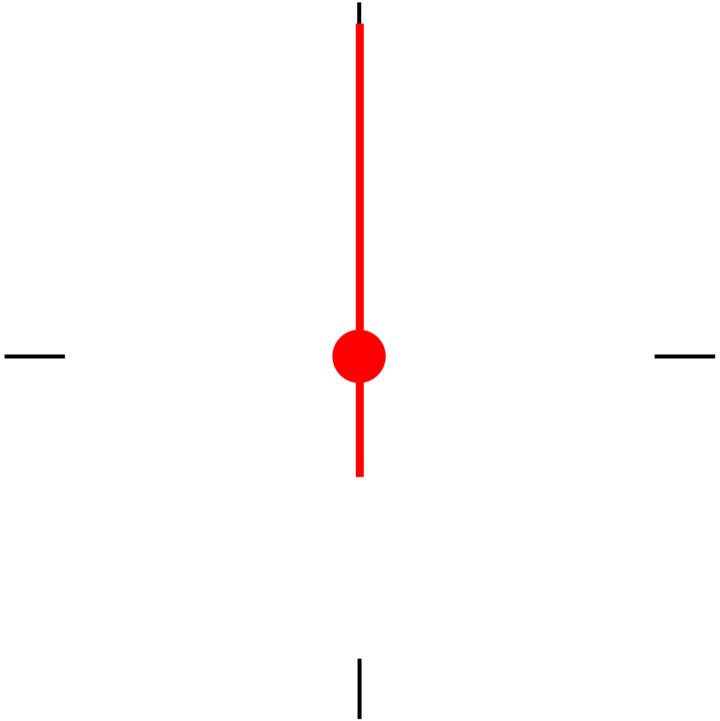


Pin spacing is 0.1” x 0.3”; chip is about 15mm long

- 74HCT00 – has four 2-input NAND gates
- **Vcc** = Power (e.g. 5V), **GND** = Ground (0V)

Short Break

- Stand up and stretch



Logic Function Implementation

- Any logic function can be implemented as the OR of AND combinations of the inputs
  - Called **sum of products**
- Example:
  - Consider truth table
  - For each '1' in the output column, write down the AND combination of inputs that give that 1
  - OR these together

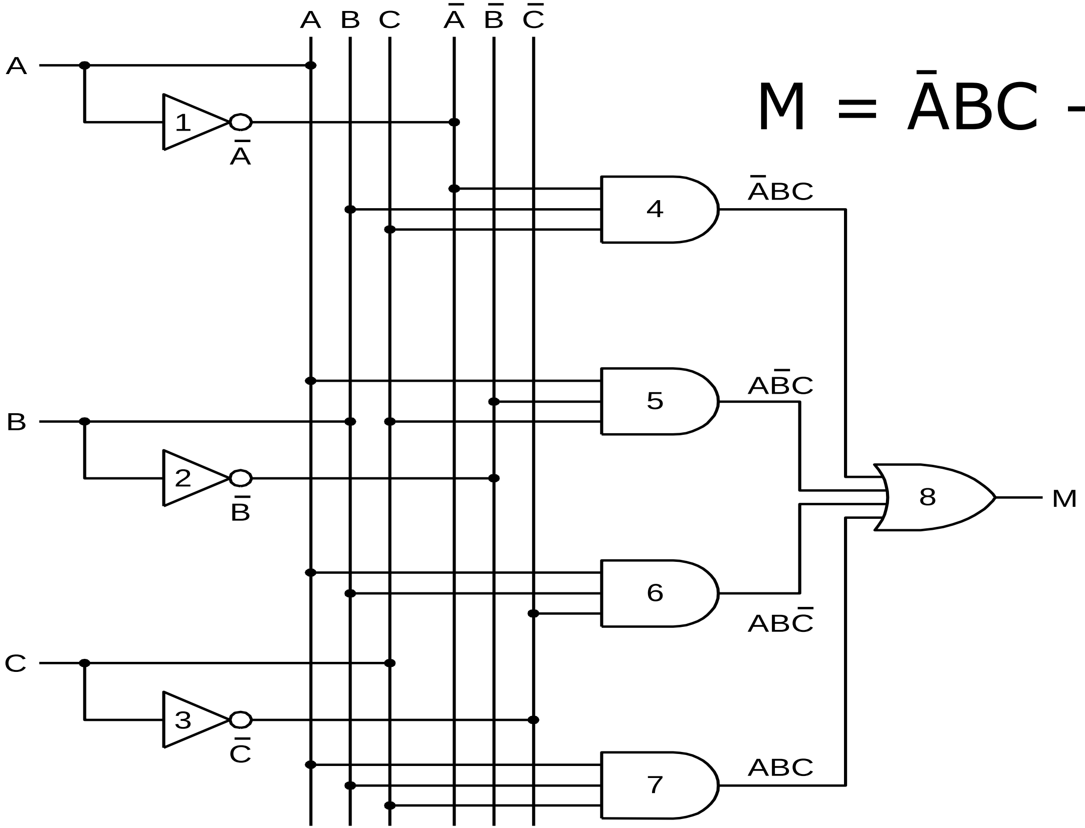
A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Logic Function Implementation

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

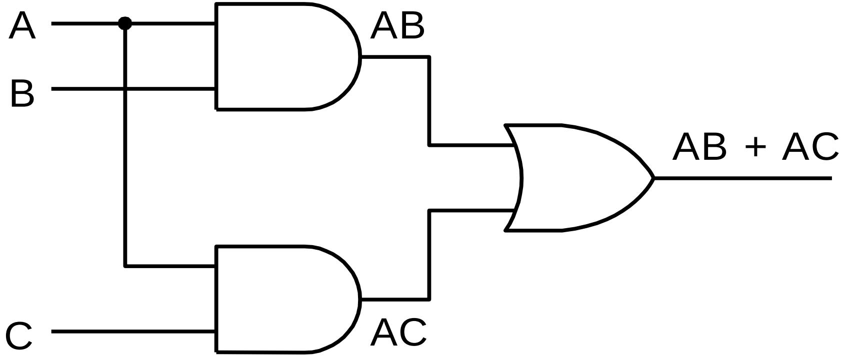
Example (cont.) Equivalent Logic Diagram

$$M = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

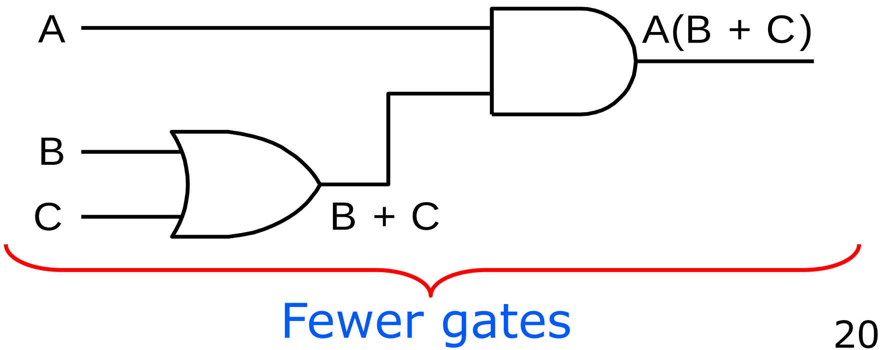


Equivalent Functions

- Sum of products does not necessarily produce circuit with minimum number of gates
- Can *manipulate* Boolean function to give an equivalent function
  - Use rules of Boolean algebra (next slide)
- Example: Z = AB + AC = A(B+C)



Simplified version



Boolean Identities

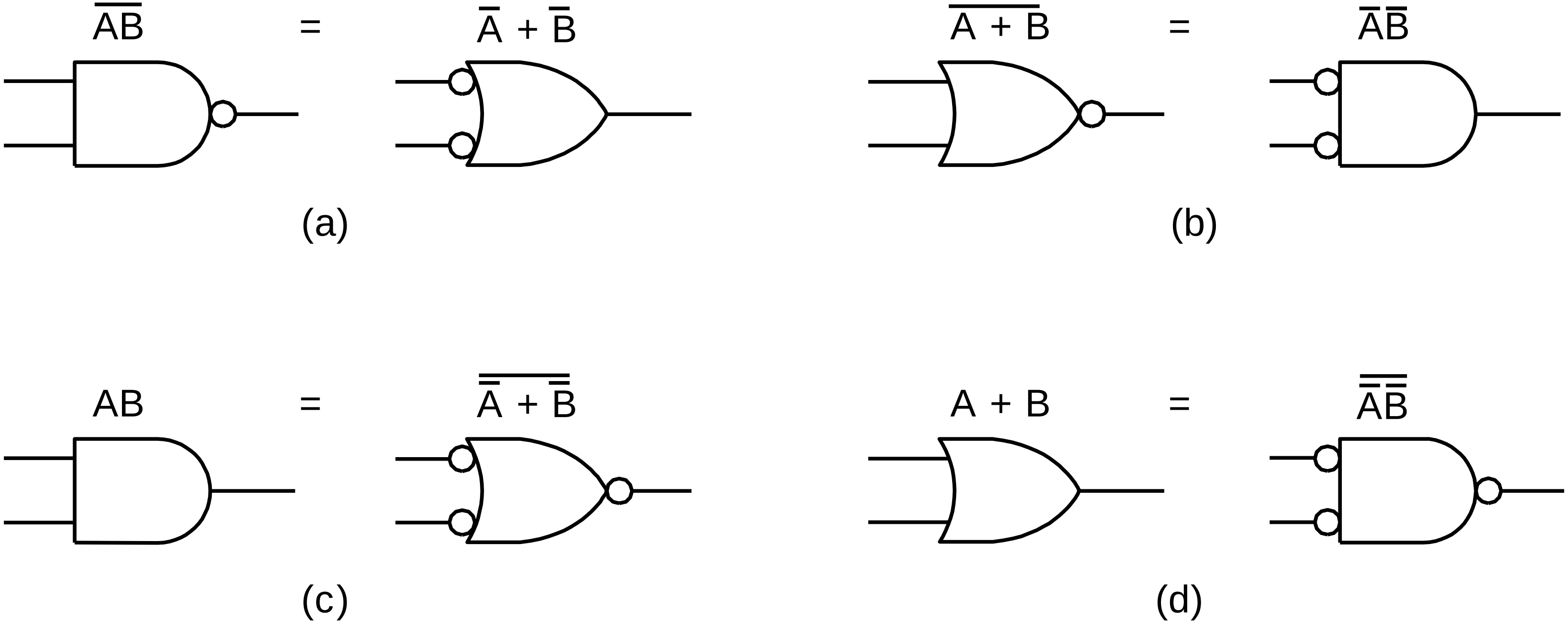
Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributative law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

Example

- Express $Z = \overline{A(B+C(\overline{A} + \overline{B}))}$ as a sum of products

De Morgan Law/Equivalents

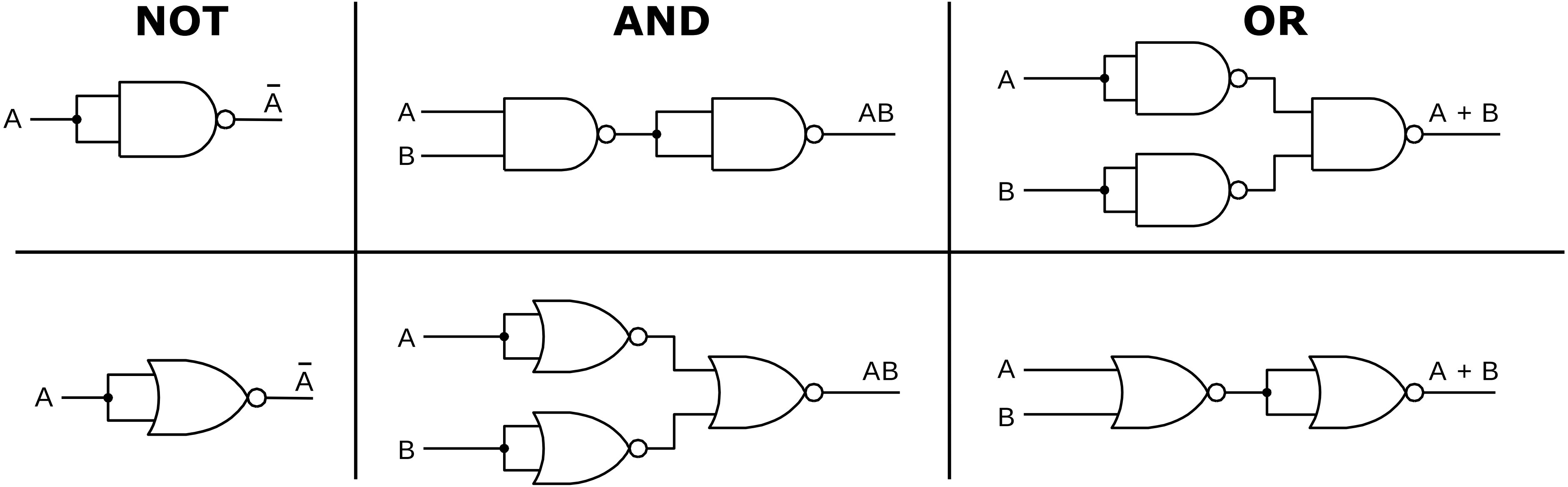
- AND/OR can be interchanged if you invert the inputs and outputs



- Homework: Use truth tables to convince yourself that these are valid

Equivalent Circuits

- All circuits can be constructed from NAND or NOR gates
  - These are called **complete** gates
- Examples:



- Reason: Easier to build NAND and NOR gates from transistors

Reminders

- Quiz 1 due next week Friday
- Attend Learning Lab sessions for the second half of this week
  - Only attend the session you're signed up to
  - Internal (IN) mode students should collect a kit in their face-to-face prac sessions.
  - External (EX) mode students, you do not need your hardware until week 6. But start acquiring your hardware items now.