Introduction to Computer Science & Engineering

Lecture 4: Gates and Circuits

Jeonghun Park

Gates and Circuits

- Gate
  - A device that operates a basic operation on electrical signals
- Circuits
  - Combined gates to perform more complicated tasks

jeonghun.park@knu.ac.kr

Descriptions

- Boolean operation
  - A mathematical notation expressing two valued-logic
- Logic diagram
  - A graphical representation of a circuit;
  - Each gate has its own symbol
- Truth table
  - A table showing all possible input values and the associated output values

jeonghun.park@knu.ac.kr

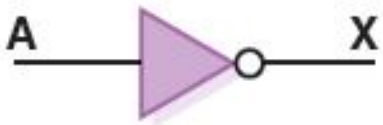
Gates

- Six types
  - **NOT**
  - **AND**
  - **OR**
  - **XOR**
  - **NAND**
  - **NOR**

jeonghun.park@knu.ac.kr

NOT

- Accepts single input
- Returns the complementary signal as output

Boolean Expression	Logic Diagram Symbol	Truth Table
X = A'		

A	X
0	1
1	0

FIGURE 4.1 Representations of a NOT gate

jeonghun.park@knu.ac.kr

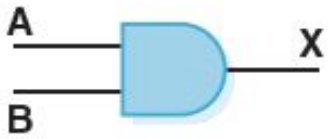
AND

- Accepts two inputs
- Returns 1 only if two inputs are both 1

Boolean Expression

$X = A \cdot B$

Logic Diagram Symbol



Truth Table

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

FIGURE 4.2 Representations of an AND gate

jeonghun.park@knu.ac.kr

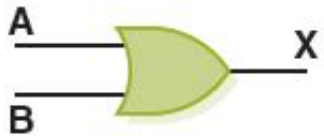
OR

- Accepts two inputs
- Returns 0 only if two inputs are both 0

Boolean Expression

$X = A + B$

Logic Diagram Symbol



Truth Table

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

FIGURE 4.3 Representations of an OR gate

jeonghun.park@knu.ac.kr

XOR

- Accepts two inputs
- Returns 0 if two inputs are same
- Returns 1 if two inputs are different

Boolean Expression

$X = A \oplus B$

Logic Diagram Symbol

Truth Table

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

FIGURE 4.4 Representations of an XOR gate

jeonghun.park@knu.ac.kr

Some Notes

- Note the difference between the XOR gate and the OR gate; they differ only in one input situation
  - When both input signals are 1, the OR gate produces a 1 and the XOR produces a 0
- XOR is called the *exclusive OR* because its output is 1 if (and only if):
  - *either* one input *or* the other is 1,
  - *excluding* the case that they both are

jeonghun.park@knu.ac.kr

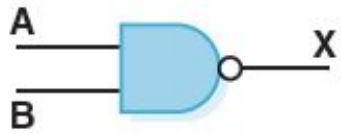
NAND

- Accepts two inputs
- Returns 0 only if two inputs are both 1

Boolean Expression

$X = (A \cdot B)'$

Logic Diagram Symbol



Truth Table

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

FIGURE 4.5 Representations of a NAND gate

jeonghun.park@knu.ac.kr

NOR

- Accepts two inputs
- Returns 1 only if two inputs are both 0

Boolean Expression

$X = (A + B)'$

Logic Diagram Symbol

Truth Table

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

FIGURE 4.6 Representations of a NOR gate

jeonghun.park@knu.ac.kr

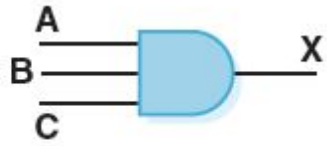
Gates with More Inputs

- Some gates can be generalized to accept three or more input values

Boolean Expression

$X = A \cdot B \cdot C$

Logic Diagram Symbol



Truth Table

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

FIGURE 4.7 Representations of a three-input AND gate

jeonghun.park@knu.ac.kr

Constructing Gates

- Transistor
  - A device that acts either as a wire that conducts electricity or as a resistor that blocks the flow of electricity, depending on the voltage level of an input signal
  - A transistor has no moving parts, yet acts like a switch
  - It is made of a semiconductor material, which is neither a particularly good conductor of electricity nor a particularly good insulator

jeonghun.park@knu.ac.kr

NOT Gate Example

- If the Base signal is low, the transistor acts like an open switch, so the output is same as the source
- If the Base signal is high, the transistor acts like an closed switch, so the Output is pulled low

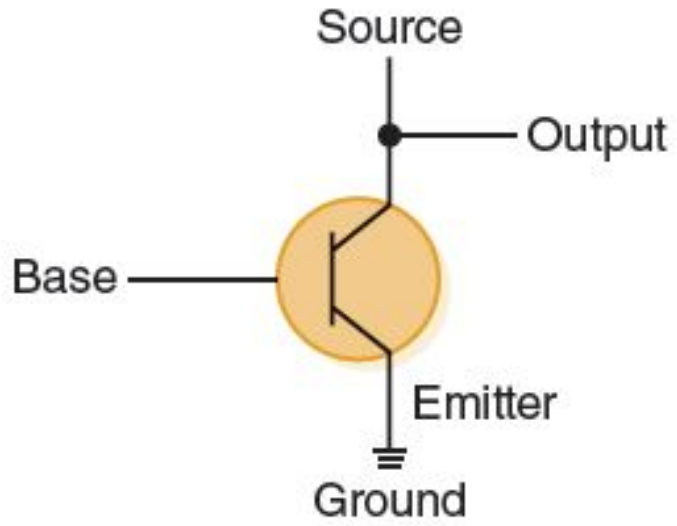
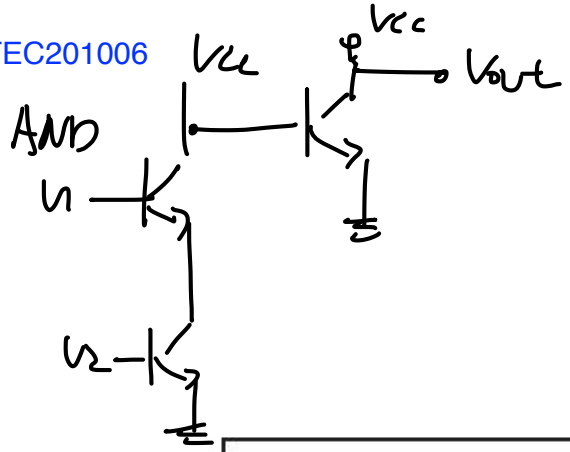


FIGURE 4.8 The connections of a transistor

jeonghun.park@knu.ac.kr



More Examples

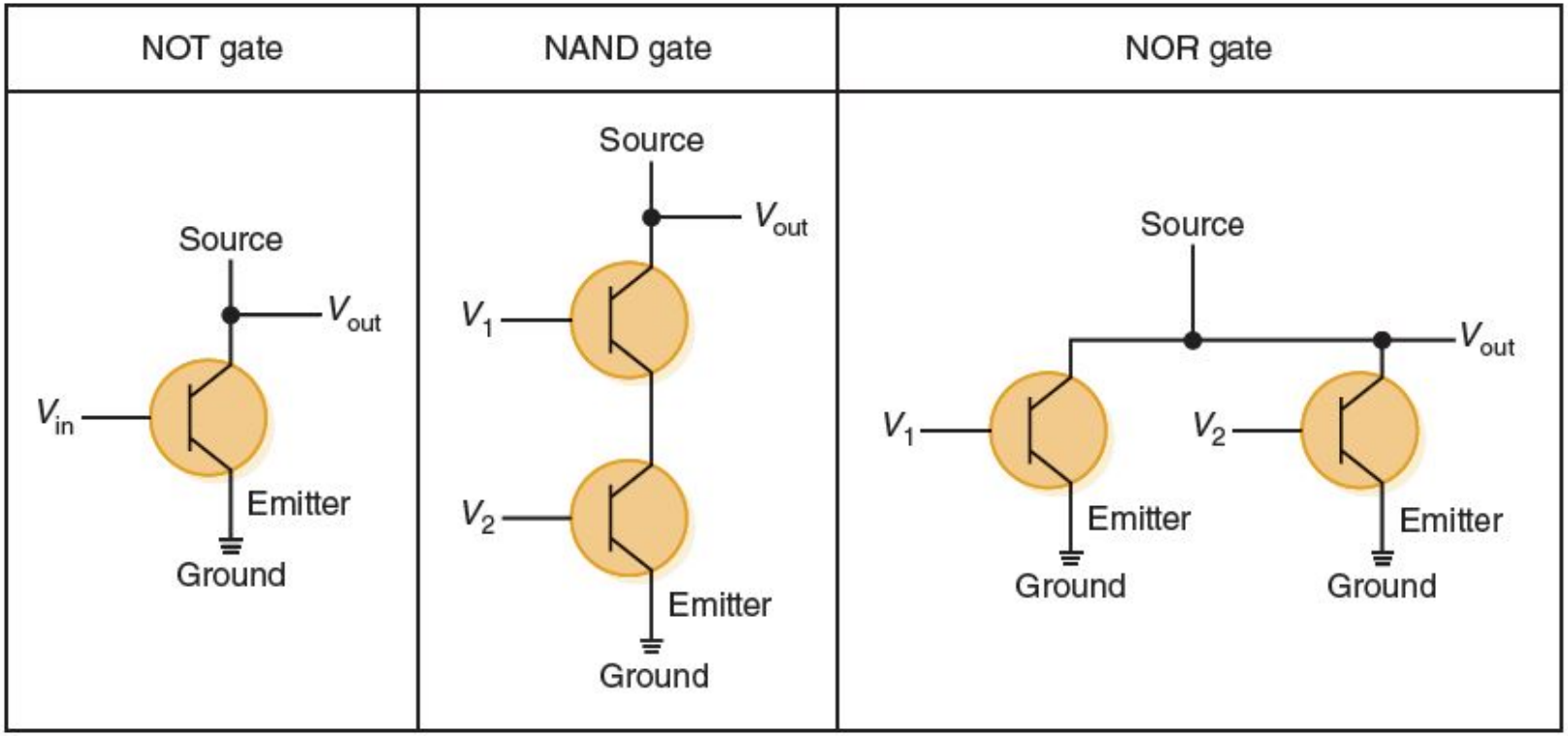
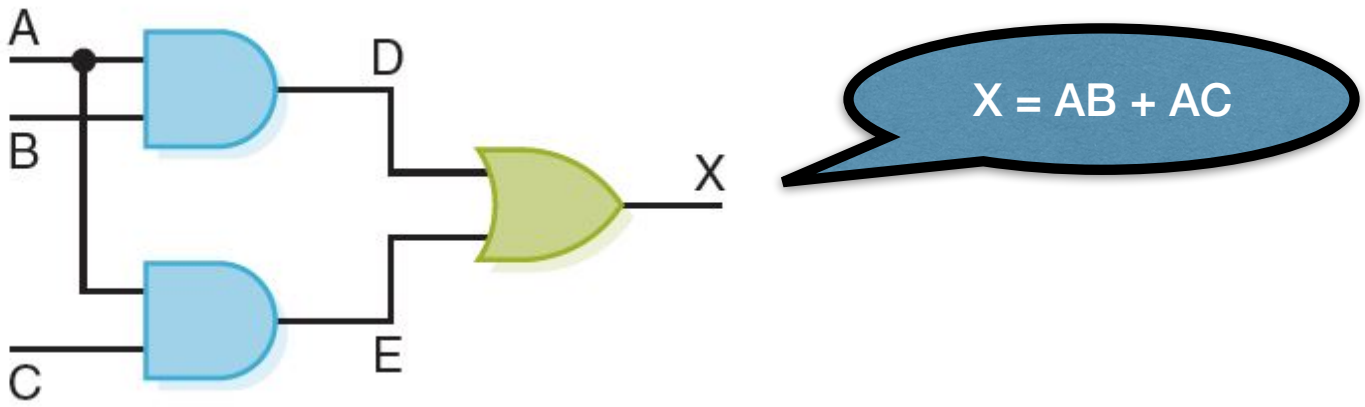


FIGURE 4.9 Constructing gates using transistors

jeonghun.park@knu.ac.kr

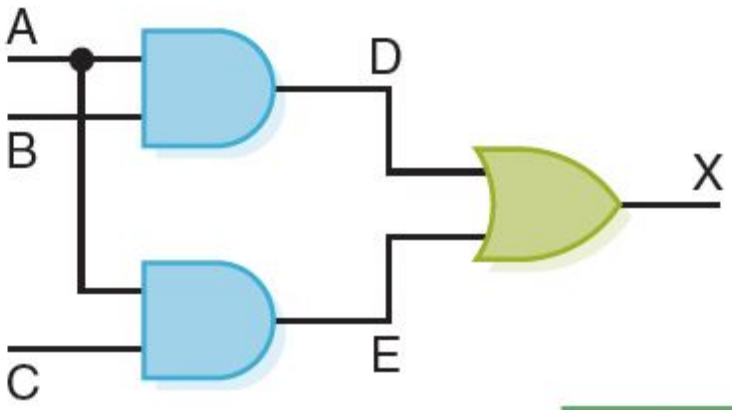
Circuits

- Gates are combined into circuits by using the output of one gate as the input for another



jeonghun.park@knu.ac.kr

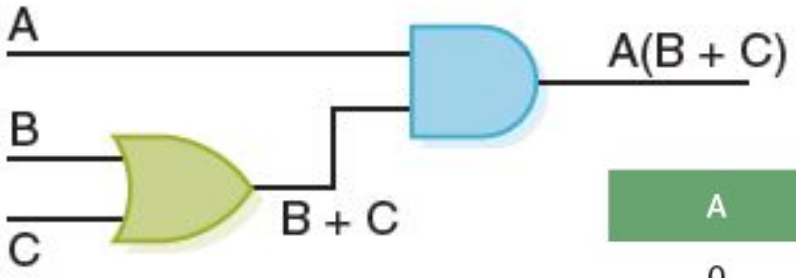
Truth Table



A	B	C	D	E	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

jeonghun.park@knu.ac.kr

Circuits



A	B	C	B + C	A(B + C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Different One?

jeonghun.park@knu.ac.kr

Circuits

- Two example circuits are equivalent
- Boolean algebra
  - Distributive law is satisfied in Boolean algebra
  - AB + AC = A(B+C)

jeonghun.park@knu.ac.kr

Other Properties

PROPERTY	AND	OR
Commutative	AB = BA	A + B = B + A
Associative	(AB) C = A (BC)	(A + B) + C = A + (B + C)
Distributive	A (B + C) = (AB) + (AC)	A + (BC) = (A + B) (A + C)
Identity	A1 = A	A + 0 = A
Complement	A(A′) = 0	A + (A′) = 1
De Morgan's law	(AB)′ = A′ OR B′	(A + B)′ = A′B′

jeonghun.park@knu.ac.kr

Adders

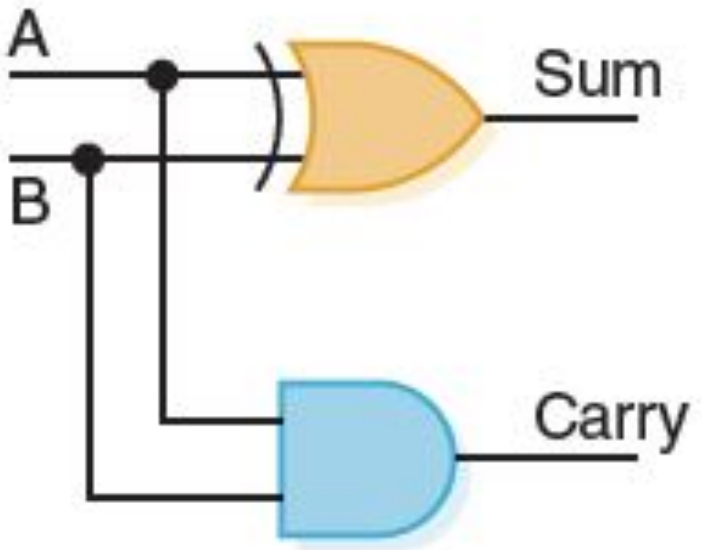
- How we implement the addition in binary with the gates?
- Two things we care:
  - Sum
  - Carry

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

jeonghun.park@knu.ac.kr

Implementation

- Boolean expression
  - Sum = A⊕B
  - Carry = AB



jeonghun.park@knu.ac.kr

Full Adders

Logic Diagram

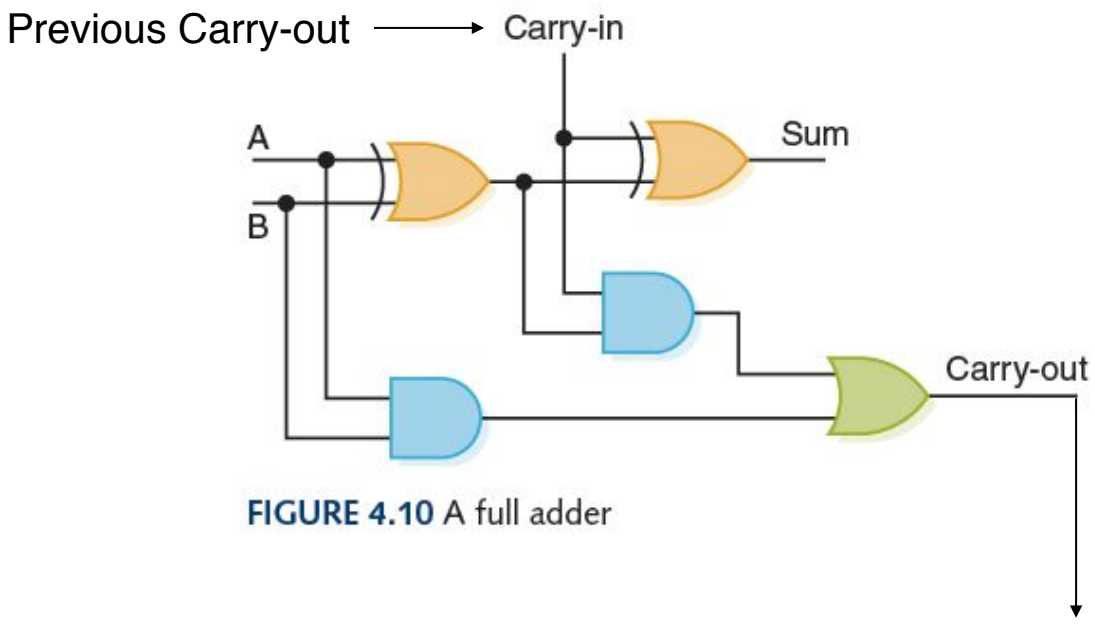


FIGURE 4.10 A full adder

Truth Table

A	B	Carry-in	Sum	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

jeonghun.park@knu.ac.kr

Multiplexers

- The control lines S0, S1, and S2 determine which of eight other input lines (D0 … D7) are routed to the output (F)

S0	S1	S2	F
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

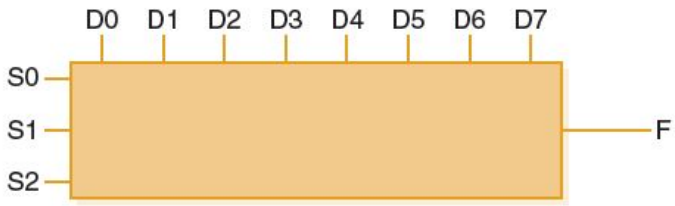


FIGURE 4.11 A block diagram of a multiplexer with three select control lines

jeonghun.park@knu.ac.kr

Circuits as Memory

- Digital circuits can be used to store information
- These circuits form a sequential circuit, because the output of the circuit is also used as input to the circuit

jeonghun.park@knu.ac.kr

S-R Latch

- SR Latch with NAND gates

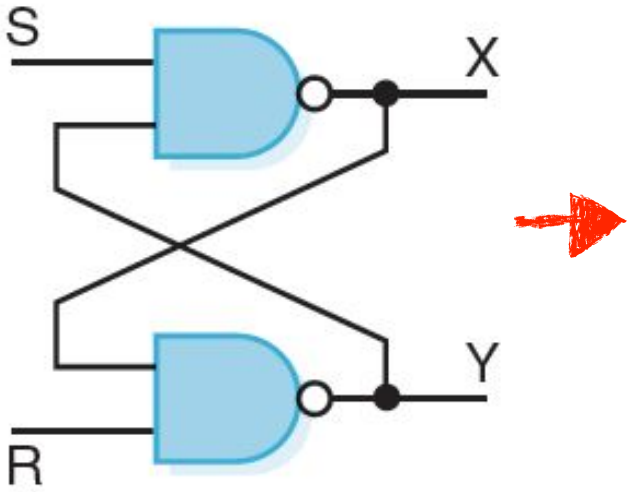
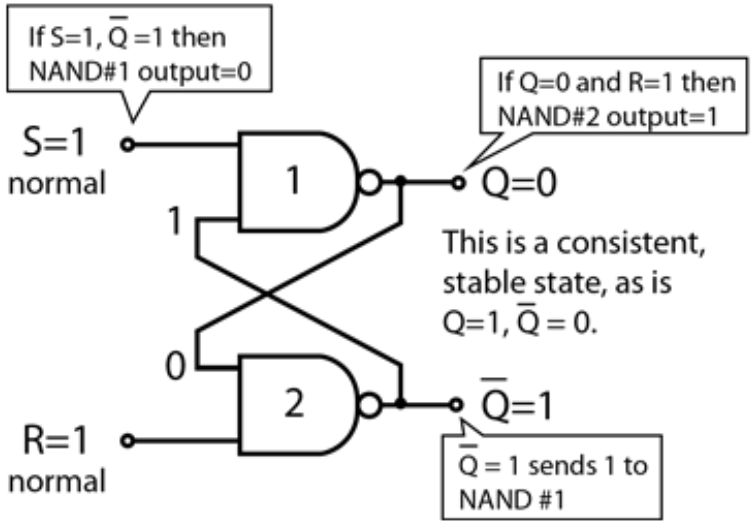


FIGURE 4.12 An S-R latch



Truth Table

Set	Reset	Output
1	1	No change*
0	1	Q=1
1	0	Q=0
0	0	Invalid state

* can be used for data storage

How do we store the information?

jeonghun.park@knu.ac.kr