부울대수 $\xrightarrow{\text{논리적}}$ 디지털회로

디지털설계의 중요한 이론

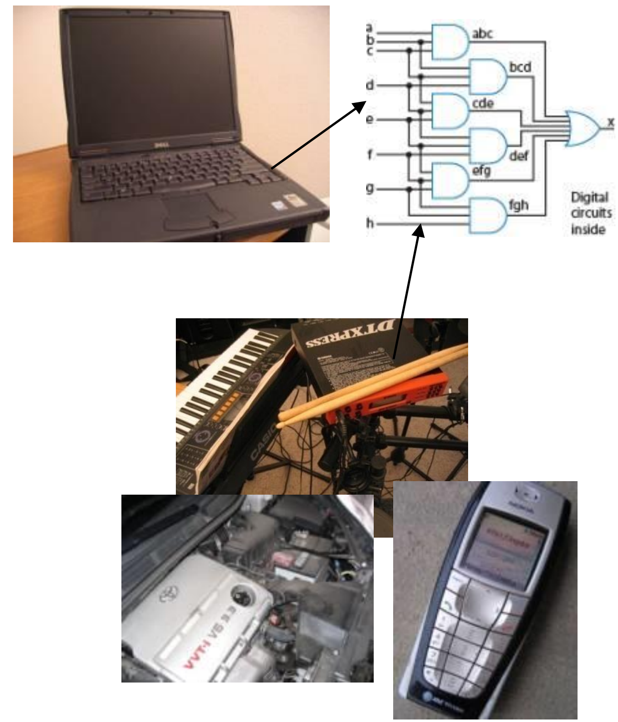
Boolean Algebra

- Base Design Theory for Digital Circuit -

Introduction to Digital Circuits

Why Study Digital Circuit Design?

- Look "inside"of computers.
  - Composed of several hardware components such as CPU, memory, GPU, etc.
  - Each hardware component in computer is implemented with digital circuits.
- Look "inside" of other electronic devices such as
  - sound recorders, cameras, navigation in cars, medical device, ...
  - They are also implemented with digital circuits.
- Why study digital circuit design?
  - We can exhaustively understand how to implement hardware in detail. 철두철미하게 이해
  - It provides confidence and insight into field of computer science.



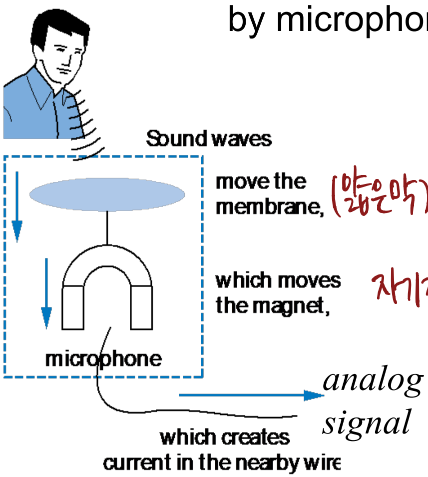
What Does "Digital" Mean?

- Analog (Continuous) signal
  - Infinite possible values
    - Ex: voltage on a wire created by microphone

↔

- Digital (Discrete) signal
  - Finite possible values
    - Ex: button pressed on a keypad

불연속적, 유한, 명확

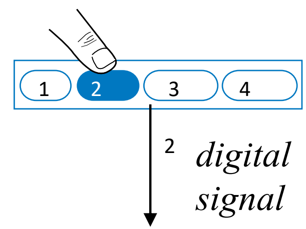


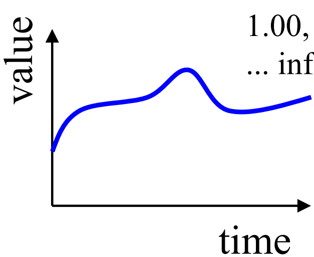
(얇은막)

자기장

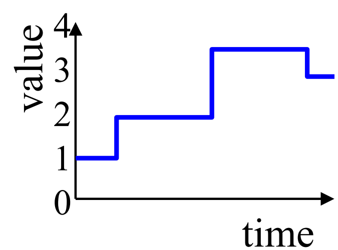
전자기유도

전류





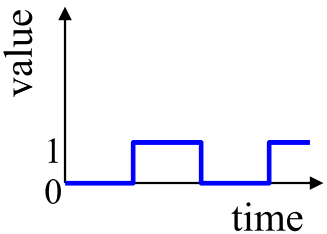
Possible values:
1.00, 1.01, 2.0000009,
... infinite possibilities



Possible values:
0, 1, 2, 3, or 4.
That's it.

Digital Signals with Only Two Values: Binary

- ***Binary*** digital signal -- only *two* possible values
  - Typically represented as **0** and **1**
  - One *b*inary dig*it* is a ***bit***
  - We'll only consider *binary* digital signals
    - **Why?**



Benefit of Binary Digitization

전자기의 특징 때문에 .→ 신호의 왜곡

- **Analog signal (e.g., audio) may lose quality**
  – Voltage levels not saved/copied/transmitted perfectly

Let bit encoding be: 1 V: "01" 2 V: "10" 3 V: "11"

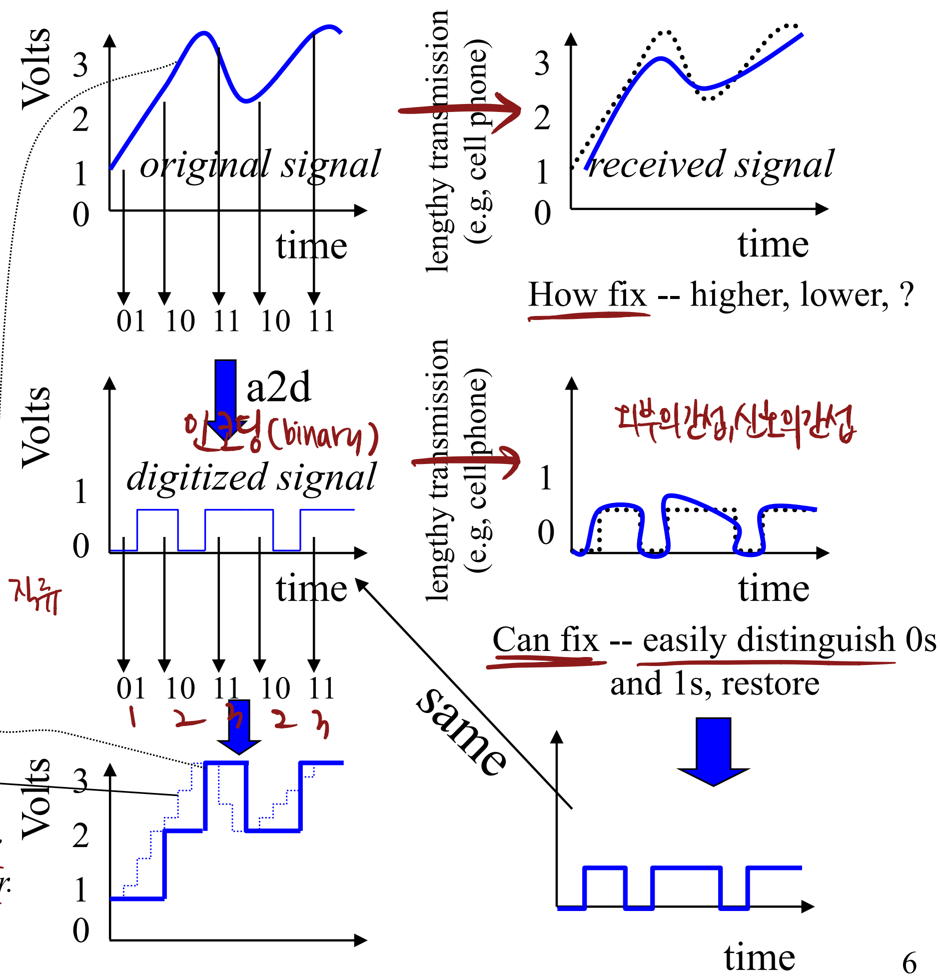
- **Binary digitization enables near-perfect save/cpy/trn.**
  – Voltage levels still not kept perfectly.
  – But we can distinguish 0s from 1s.

복원의 정확성

사람이 알아차리지 못할 만큼.

감각기관의 임계치↑

Digitized signal not perfect re-creation, but higher sampling rate and more bits per encoding brings closer.



Basis and Theory for Digital Circuits

디지털회로설계

Basis and Theory for Digital Circuits

- Before learning to design digital circuits, two questions…

반도체 field를 MOS가 평정

디지털회로의 basis.

  – Q1: What's '*fundamental basis*' of digital circuit?
    - CMOS(Complementary Metal Oxide Semiconductor)

    상보형 금속산화 반도체 : 트랜지스터소자 → 0/1로 스위칭이 일어나는주체

  – Q2: What's '*base design theory*' for implementing digital circuit ?
    - Boolean Algebra

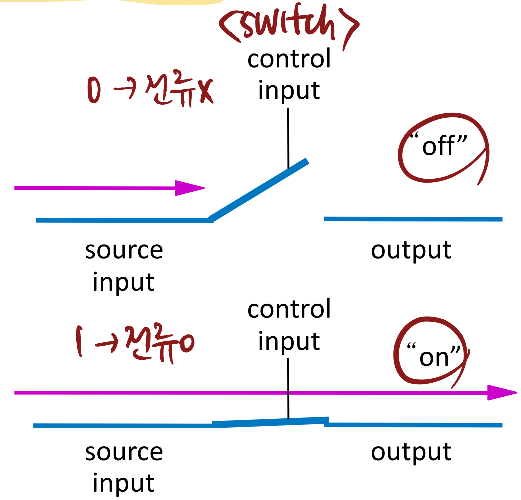
    ↳ 그자체가 base design theory

    부울대수만으로 디지털회로설계

Fundamental Basis

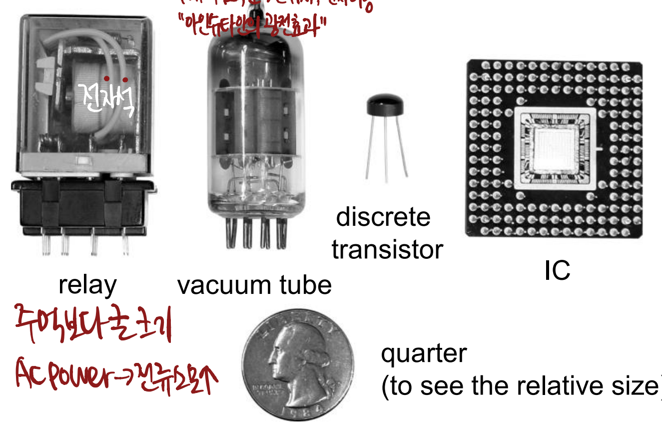
전자스위치

- Electronic switches are the basis of binary digital circuits
- A switch has three parts
  - Source input, and output
    - Current wants to flow from source input to output.
  - Control input
    - Controls whether that current can flow.



- The amazing shrinking switch
  - 1930s: Relays 전기스위치
  - 1940s: Vacuum tubes 진공관-스위칭↑
  - 1950s: Discrete transistor (반도체)
  - 1960s: Integrated circuits (ICs)
    집적회로
    - Initially just a few transistors on IC
    - Then tens, hundreds, thousands of transistors

(b)



Fundamental Basis

현미경.

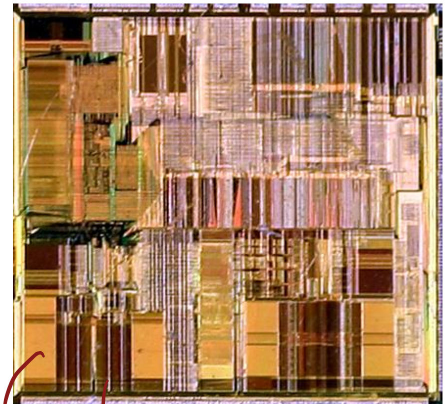
- IC(Integrated Circuit) capacity doubling about every 18 months for several decades
  - Known as "Moore's Law" after Gordon Moore, co-founder of Intel
    - Predicted in 1965 predicted that components per IC would double roughly every year or so.

      18개월

      실제는 훨씬 빠르게 집적도가 향상됨.

      → 어떻게?

      한국의 전자공학자
    - Enables incredibly powerful computation in incredibly tiny devices
  - Today's ICs hold *billions* of transistors
    - The first Pentium processor (early 1990s) needed only 3 million.



메모리

로직

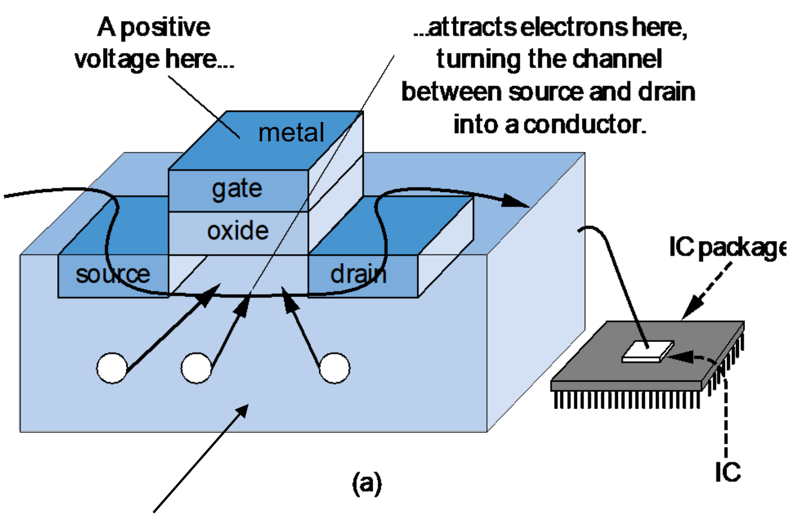
An Intel Pentium processor IC having millions of transistors

다이포토?

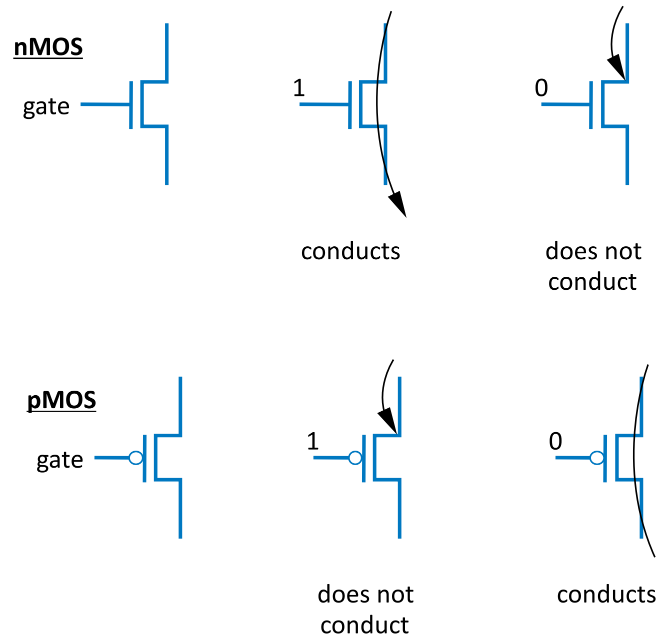
dye photo?

Fundamental Basis

- CMOS(Complementary Metal Oxide Semiconductor) transistor
  - Basic switch in modern ICs



Silicon -- not quite a conductor or insulator:
Semiconductor



Fundamental Basis

- Dr. Dae Won Kang (1931-1992)
  - an Korean electronic engineer/ physicist
  - B.S. from Seoul National Univ.
  - Ph.D from Ohio state Univ.
  - He developed the world's first MOS 1960.
  - When, as a 29-year-old engineer at AT&T Bell Lab.
  - He is also credited with inventing nonvolatile memory that is a basis of flash memory.
  - He has been the most important one who is leading IT Revolution.

Base Design Theory

- Claude Shannon (1916-2001)
  - an American mathematician, electronic engineer, cryptographer
  - Famous for having founded information theory : Known as “the father of information theory”
  - He is also credited with founding both **digital computer** and **digital circuit** design theory in 1937.
  - When, as a 21-year-old master’s student at MIT.
  - He wrote a thesis about **how the Boolean algebra could be used to design digital circuits.**
    - The circuit in electronic devices have inputs, each of which is either 0 or 1 and produce outputs that are also 0 or 1.
  - **The most important master’s thesis of all time.**

Base Design Theory

- Boolean Algebra
  - as developed in 1854 by George Boole in his book '*An Investigation of the Laws of Thought*'.

  - is a variant of ordinary elementary algebra
    - Instead of the usual algebra of numbers, Boolean algebra is the algebra of truth values 0 and 1.
    - Variables represent 0 or 1 only.
    - Operators return 0 or 1 only.

  - Basic operators
    - AND (·) : $a \cdot b$ returns 1 only when both a=1 and b=1
    - OR (+) : $a + b$ returns 1 if either (or both) a=1 or b=1
    - NOT (') : a' returns the opposite of a (1 if a=0, 0 if a=1)

Base Design Theory

- Boolean Algebra
  - Boolean Identities

$x + 0 = x$ $x \cdot 1 = x$	Identity Law	$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Associative Law
$x + 1 = 1$ $x \cdot 0 = 0$	Domination law	$x(y + z) = xy + xz$ $x + yz = (x + y)(x + z)$	Distributive Law
$x + x = x$ $x \cdot x = x$	Idempotent Law	$(x\,y)' = x' + y'$ $(x + y)' = x'\,y'$	De Morgan's Law
$(x')' = x$	Complementation Law	$x + xy = x$ $x\,(x + y) = x$	Absorption Law
$x + y = y + x$ $xy = yx$	Commutative Law	$x + x' = 1$ $x\,x' = 0$	Complement Law

Base Design Theory

- Boolean Algebra
  - Comparison between boolean identities and set identities

Set Identities

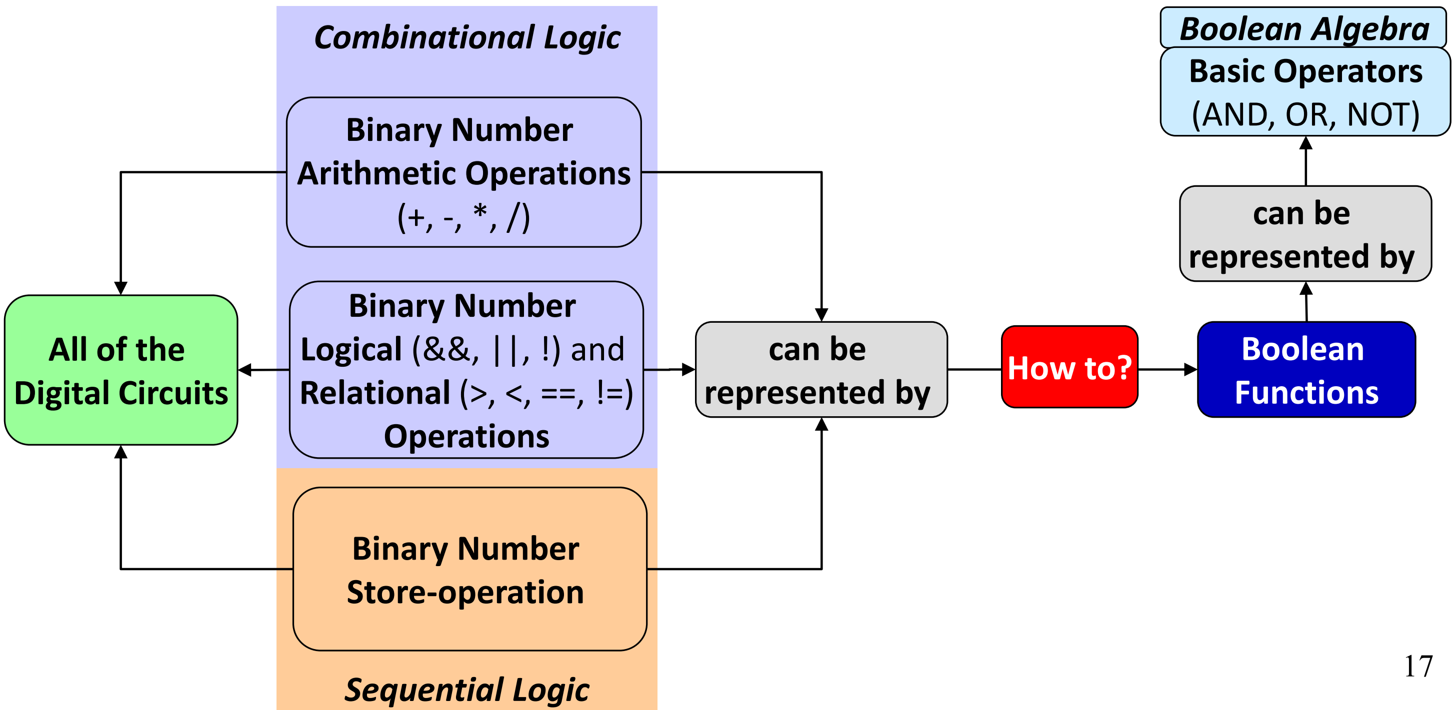
$A \cup \varnothing = A$ $A \cap U = A$	Identity Law	$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative Law
$A \cup U = U$ $A \cap \varnothing = \varnothing$	Domination law	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Distributive Law
$A \cup A = A$ $A \cap A = A$	Idempotent Law	$(A \cup B)^c = A^c \cap B^c$ $(A \cap B)^c = A^c \cup B^c$	De Morgan's Law
$(A^c)^c = A$	Complementation Law	$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption Law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative Law	$A \cup A^c = U$ $A \cap A^c = \varnothing$	Complement Law

Boolean Identities

$x + 0 = x$ $x \cdot 1 = x$	Identity Law	$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Associative Law
$x + 1 = 1$ $x \cdot 0 = 0$	Domination law	$x(y + z) = xy + xz$ $x + yz = (x + y)(x + z)$	Distributive Law
$x + x = x$ $x \cdot x = x$	Idempotent Law	$(x\ y)' = x' + y'$ $(x + y)' = x'\ y'$	De Morgan's Law
$(x')' = x$	Complementation Law	$x + xy = x$ $x(x + y) = x$	Absorption Law
$x + y = y + x$ $xy = yx$	Commutative Law	$x + x' = 1$ $x\ x' = 0$	Complement Law

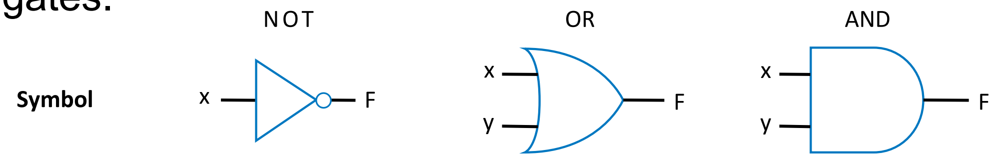
Base Design Theory

- Boolean Algebra and its Relation to Digital Circuits
  - Claude Shannon's observation
    - all of the digital circuits can be implemented by only using 3 operations in boolean algebra no matter how the circuits are very complex.



Base Design Theory

- Boolean Algebra and its Relation to Digital Circuits
  - Claude Shannon observed that all of the digital circuits can be implemented by only using 3 operations in boolean algebra no matter how the circuits are very complex.
  - In this reason, he introduced Boolean Logic Gates that are basic building blocks for digital circuits.
  - Boolean logic gates corresponds to 3 basic operators in boolean algebra.
    - AND, OR, NOT
  - Engineers can easily analyze and design digital circuits by using boolean logic gates.



a	NOT
0	1
1	0

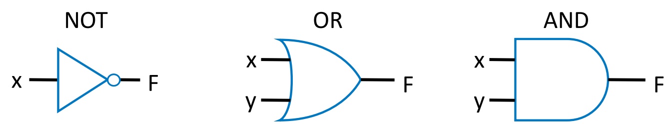
a	b	OR
0	0	0
0	1	1
1	0	1
1	1	1

a	b	AND
0	0	0
0	1	0
1	0	0
1	1	1

Base Design Theory

- Boolean Logic Gates
  - are eventually composed of CMOS transistor.

Symbol



Truth table

NOT

x	F
0	1
1	0

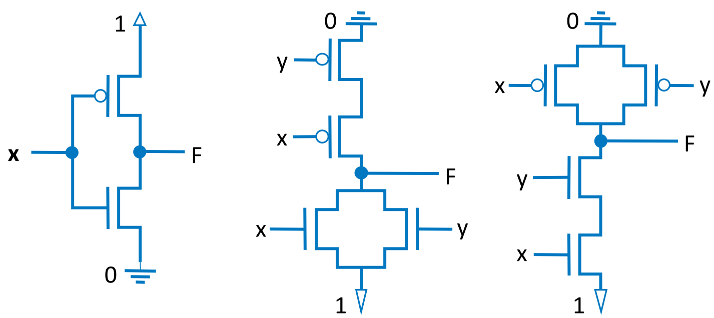
OR

x	y	F
0	0	0
0	1	1
1	0	1
1	1	1

AND

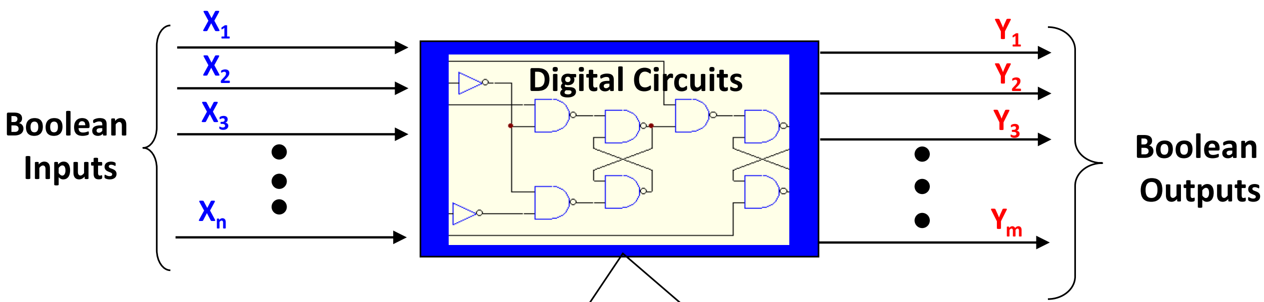
x	y	F
0	0	0
0	1	0
1	0	0
1	1	1

CMOS Transistor circuit



Base Design Theory

- When we design digital circuits
  - Specification



The digital circuits are physically implemented by connecting boolean inputs/outputs with several boolean logic gates.

It means that the digital circuits are logically implemented by boolean functions.

Base Design Theory

- Representations of Boolean Functions
  - The Below shows seven representations of the same functions F(a,b) using four different methods.

English 1: F outputs 1 when a is 0 and b is 0, or when a is 0 and b is 1.

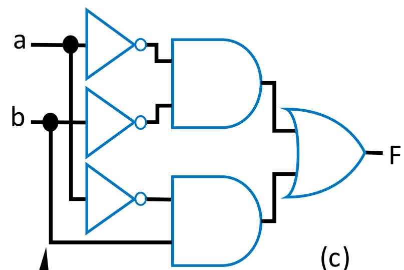
English 2: F outputs 1 when a is 0, regardless of b's value

(a)

Equation 1: F(a,b) = a'b' + a'b

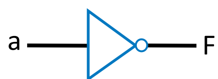
Equation 2: F(a,b) = a'

(b)



Circuit 1

(c)



Circuit 2

a	b	F
0	0	1
0	1	1
1	0	0
1	1	0

Truth table

(d)

The function F

Base Design Theory

- Four methods for representing boolean functions
  - English Sentences
    - Too verbose for numerous inputs
  - Truth tables
    - Too big for numerous inputs
  - Drawing circuits with logic gates
    - Too complex for numerous inputs
  - Boolean equations
    - precise and explicit representation even though numerous inputs.

Base Design Theory

- Boolean equations
  - It can be represented using boolean variables and operators.
  - Examples of boolean equations
    - 0, 1, boolean variable itself (x_1, x_2, …, x_n)
    - If E_1 and E_2 are Boolean equations,

      then E_1', $E_1 \cdot E_2$ and $E_1 + E_2$ are also Boolean equations.