



CONTENTS

학습목표

- 컴퓨터에서 자료를 표현하는 방법을 설명할 수 있다.
- 컴퓨터에서 사용되는 진수와 수에 대해 설명할 수 있다.
- 컴퓨터에서 이용하는 정보의 종류를 설명할 수 있다.
- 컴퓨터에서 논리연산 방법과 논리회로설계 방법을 설명할 수 있다.

▶▶▶▶▶
들어가기

LEARNING

지난 주차 복습

▶▶▶▶▶
복습하기

01주차 학습내용. 정보의 표현

- 1 컴퓨터의 이해
- 2 컴퓨터의 기원
- 3 컴퓨터의 역사
- 4 컴퓨터의 종류

지난 주차 **복습**

LEARNING
복습하기

컴퓨터의 이해

- 컴퓨터란 데이터의 처리를 담당하는 전자계산기임
- 컴퓨터의 구성요소
 - 하드웨어와 소프트웨어로 구분

컴퓨터의 기원

- 계산도구의 기원은 주판이라 할 수 있음
- 최초의 컴퓨터는 에니악(ENIAC)임

지난 주차 **복습**

LEARNING
복습하기

컴퓨터의 역사

- 컴퓨터의 세대별 분류 : 논리소자
 - 1세대 : 진공관
 - 2세대 : 트랜지스터
 - 3세대 : 집적회로(IC)
 - 4세대 : 고밀도 집적회로
 - 5세대 : 초고밀도 집적회로

지난 주차 **복습**

복습하기

컴퓨터의 종류

- ◆ 크기와 처리 능력에 따른 구분
 - 슈퍼 컴퓨터, 메인 프레임, 미니 컴퓨터, 워크스테이션
 - 마이크로 컴퓨터, 휴대용 컴퓨터, 개인용 디지털 보조기
- ◆ 사용목적에 따른 구분
 - 범용 컴퓨터, 전용 컴퓨터
- ◆ 모바일 컴퓨팅
- ◆ 유비쿼터스 컴퓨팅
- ◆ 클라우드 컴퓨팅

1

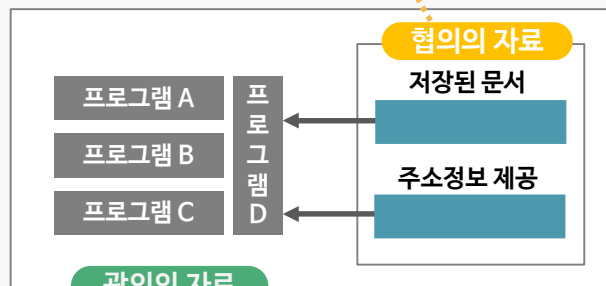
컴퓨터의 자료 표현

1] 자료 표현 원리

학습하기

◇ 자료의 정의

프로그램 실행을 위해 필요한 것



프로그램과 그 프로그램을 이용해 원하는 결과를 얻기 위해 사용하는 모든 재료

1] 자료 표현 원리

학습하기

◇ 0과 1로 문장 표현

문자 'a'

01100001

정수 20

0000000000010100

Here are some words

H	0100	1000
e	0110	0101
r	0111	0010
e	0110	0101
	0010	0000
a	0110	0001
r	0111	0010
e	0110	0101
	0010	0000
s	0111	0011
o	0110	1111
m	0110	1101
e	0110	0101
	0010	0000
w	0111	0111
o	0110	1111
r	0111	0010
d	0110	0100
s	0111	0011
.	0010	0001

0
1
2
3
4
5
6
7
8
9
10
11
:
:
98
99
100
101

Ten different symbols in the decimal system.

Numbers above 9 use more than one digit.

1's digit
10's digit
100's digit

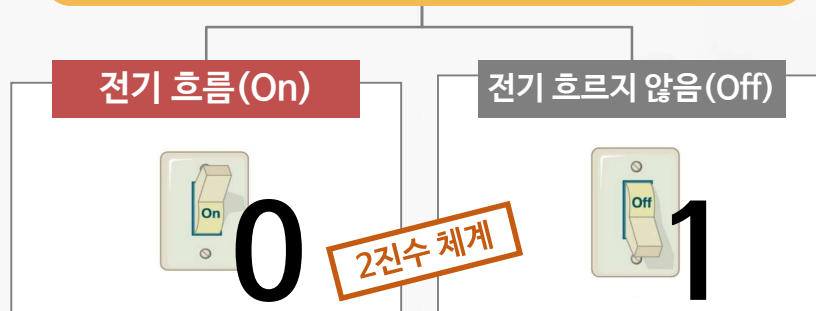
1] 자료 표현 원리

학습하기

◇ 두 가지 자료 표현

“ 컴퓨터 내부에서는 두 가지 전기신호만을 표현할 수 있는 ”
트랜지스터를 이용하여 자료를 처리하고 저장함

트랜지스터의 전기적 스위치 0과 1의 이진 표현

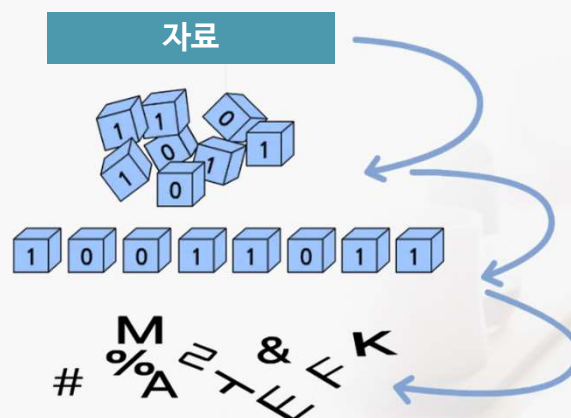


1] 자료 표현 원리

학습하기

◇ 두 가지 자료 표현

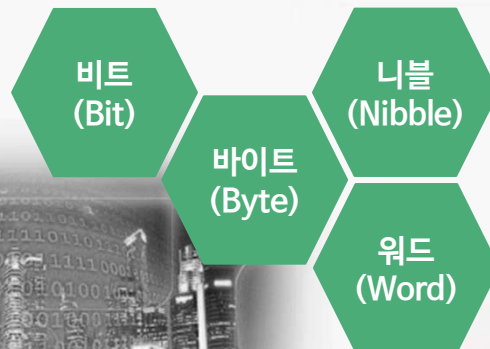
자료가 컴퓨터를 거쳐 사람에게 보여질 때까지



1] 자료 표현 원리

학습하기

◇ 비트, 바이트, 니블, 워드



1] 자료 표현 원리

학습하기

◇ 비트, 바이트, 니블, 워드

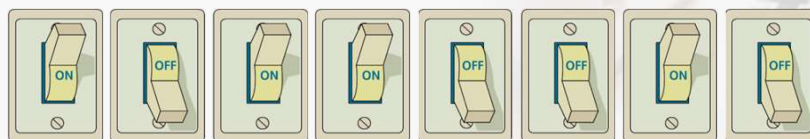
1 비트 (Bit)

- 컴퓨터의 정보 처리 단위 중에서 가장 작은 정보 단위(Binary digit)

2 바이트 (Byte)

- 비트가 연속적으로 8개 모인 정보 단위
- 1바이트 = 8개 비트 조합 : 총 256(2^8)가지 종류의 정보 저장 가능

트랜지스터의 전기적 스위치로 구성된 8비트 10110010의 표현



1] 자료 표현 원리

학습하기

◇ 비트, 바이트, 니블, 워드

3 니블(Nibble)

- 바이트의 1/2 크기인 4비트

4 워드(word)

- 4개의 바이트 모음
- 시스템마다 크기가 다를 수 있음

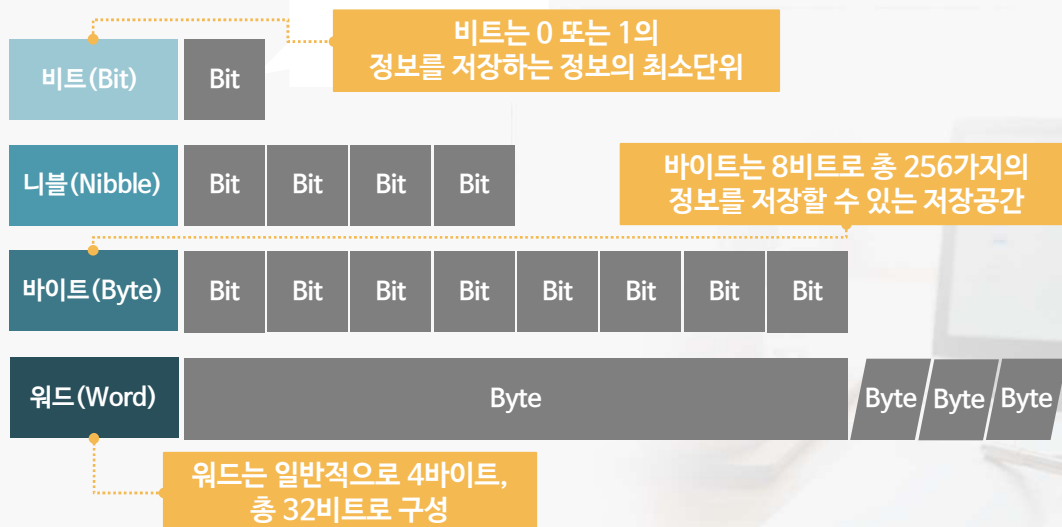
윈도우 시스템 : 32비트 = 1워드

유닉스 시스템 : 64비트 = 1워드

1] 자료 표현 원리

학습하기

◇ 비트, 바이트, 니블, 워드



2] 저장 용량

학습하기

◇ 저장 용량의 단위

◆ 바이트가 저장 용량의 기본 단위



◆ 기억용량 비교

A4 용지 텍스트 1장	0.002MB	CD 오디오 수준의 음악 1분	10MB
이미지(정지화상) 1장	1MB	비디오(동화상) 1초	27MB

2] 저장 용량

학습하기

◇ 정보의 단위와 MP3 파일 저장 용량

구분	B	KB	MB	GB	TB	PB	EB	ZB	YB
단위	바이트	킬로 바이트	메가 바이트	기가 바이트	테라 바이트	페타 바이트	엑사 바이트	제타 바이트	요타 바이트
하위 단위 환산	1Byte	1024B	1024K	1024 MB	1024 GB	1024 TB	1024 PB	1024 EB	1024 ZB
바이트 환산	2 ⁰ B	2 ¹⁰ B	2 ²⁰ B	2 ³⁰ B	2 ⁴⁰ B	2 ⁵⁰ B	2 ⁶⁰ B	2 ⁷⁰ B	2 ⁸⁰ B
MP3(4MB) 저장 곡 수	-	-	0.25	256	26만 2000	2억 7000만	2748억 8000만	281조 5000억	28경 8000조

2

진수와 수의 표현

1] 진수의 종류

 학습하기

10진수

- 0~9의 10가지의 수를 한 자리(Digit)의 기본단위로 사용
- 10진수
 - 각 자리는 오른쪽부터 1(10^0)자리, 10(10^1)자리, 100(10^2)자리 순
- N진수
 - 각 자릿수는 0에서 N-1까지의 정수를 이용함
 - 오른쪽부터 n번째 자리의 크기는 N^{n-1}

2진수

- 0과 1의 **2가지 표현**으로 각 자릿수를 표시

8진수

- 0부터 7까지의 **8가지의 수**를 이용하여 숫자를 표시

16진수

- 0에서 9까지 그리고 A에서 F까지 총 **16개의 숫자나 문자**를 사용하여 표시

1] 진수의 종류

학습하기

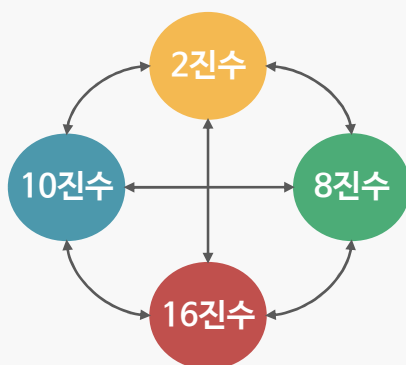
◇ 2진수의 의미



2] 진수의 변환

학습하기

◇ 진법 변환 관계



10진수	2진수	2진 표현
0	0	000
1	1	001
2	10	010
3	11	011
4	100	100
5	101	101
6	110	110
7	111	111

2] 진수의 변환

학습하기

◇ 10진수를 2진수로 변환

정수 변환

2	26	
2	130
2	61
2	30
	11

↑

소수를 2진수로 변환

	0.6875
x	2
1.3750	
x	2
0.7500	
x	2
1.5000	
x	2
1.0000	

↓

2] 진수의 변환

학습하기

◇ 2진수, 8진수, 16진수 간 상호관계

수식 $8 = 2^3$, $16 = 2^4$ 이 만족하므로 2진수로 표현된 수를 각각 8진수, 16진수로 표현하는데 쉽게 변환이 가능

	왼쪽 ←	.	→ 오른쪽				
8진수	4	5	.	5	4	→	(45.54) ₈
	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">↖</div> <div style="border: 1px solid black; padding: 2px;">↗</div> <div style="border: 1px solid black; padding: 2px;">↖</div> <div style="border: 1px solid black; padding: 2px;">↗</div> </div>						
2진수	001	001	01	.	101	100	
	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">↖</div> <div style="border: 1px solid black; padding: 2px;">↗</div> <div style="border: 1px solid black; padding: 2px;">↖</div> <div style="border: 1px solid black; padding: 2px;">↗</div> </div>						
16진수	2	5	.	B		→	(25.B) ₁₆

2] 진수의 변환

학습하기

◇ 2진수, 8진수, 16진수 간 상호관계

2진법	8진법	16진법
$125_{(10)} \rightarrow 1111101_{(2)}$ <div> <div>2</div><div> </div><div>125</div><div></div> <div>2</div><div> </div><div>62</div><div>...1</div> <div>2</div><div> </div><div>31</div><div>...0</div> <div>2</div><div> </div><div>15</div><div>...1</div> <div>2</div><div> </div><div>7</div><div>...1</div> <div>2</div><div> </div><div>3</div><div>...1</div> <div></div><div> </div><div>1</div><div>...1</div> </div>	$160_{(10)} \rightarrow 240_{(8)}$ <div> <div>8</div><div> </div><div>160</div><div></div> <div>8</div><div> </div><div>20</div><div>...0</div> <div></div><div> </div><div>2</div><div>...4</div> </div>	$175_{(10)} \rightarrow AF_{(16)}$ <div> <div>16</div><div> </div><div>175</div><div></div> <div></div><div> </div><div>10</div><div>...15</div> <div></div><div> </div><div>(A)</div><div>(F)</div> </div>

3] 2진수의 음수 표현

학습하기

◇ 보수

보수

- 두 수의 합이 해당 진법의 수가 되게 하는 수
- 원래의 수와 보수를 합치면 해당 진법의 수가 되는 것

예

- 10진수 3의 보수 $\rightarrow 7$
- 10진수 6의 보수 $\rightarrow 4$

3] 2진수의 음수 표현

학습하기

◇ 보수

보수는 왜 존재하는가?

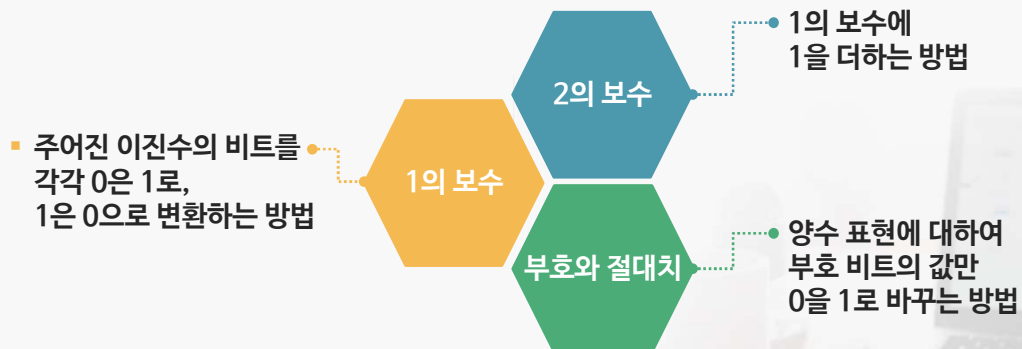
컴퓨터에서 **음의 정수**를 표현하기 위해서임

컴퓨터의 CPU에는 가산기(덧셈기)만 존재하기 때문

3] 2진수의 음수 표현

학습하기

◇ 컴퓨터에서 음수 표현 방법



3

컴퓨터의 정보 종류

1] 정수

 학습하기

부호가 있는 정수 표현

- 양수와 음수를 모두 다루는 정수를 **부호가 있는(Signed) 정수**
- 컴퓨터는 정수의 양수와 음수를 표현하는데, 주로 **2진수와 2의 보수 방법**을 이용
- 8비트의 메모리로는 **2⁸가지(256)**의 정보를 표현($2^7 \sim 2^7-1$)

부호가 없는 정수 표현

- 양수만을 다루는 정수를 **부호가 없는(Unsigned) 정수**

2] 문자와 논리



◇ 문자와 코드표

- 영문자는 7개의 비트의 조합으로 표현

아스키코드
(ASCII)

엡시딕
(EBCDIC)

유니코드
(Unicode)

2] 문자와 논리



◇ 문자와 코드표

1 아스키코드 (ASCII)

- American Standard Code for Information Interchange
- 미국 표준협회에서 국제적인 표준으로 정한 문자 코드 체계
 - 7비트를 사용하여 128개의 문자, 숫자, 특수문자 코드를 규정
- 7비트를 이용하지만 실제로 한 문자는 8비트인 1바이트에 저장
 - 한 문자의 시작은 0으로, 나머지는 코드 값으로 구성

2] 문자와 논리

학습하기

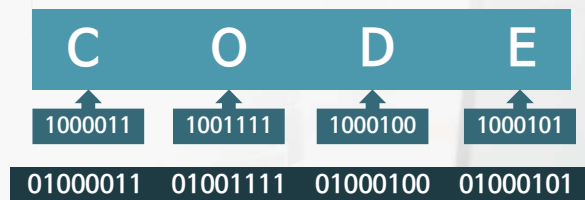
◆ 문자와 코드표

1 아스키코드 (ASCII)

아스키코드 (ASCII) 코드의 예

Code	Character
00110000	0
00110001	1
00110010	2
00110011	3
00110100	4
00110101	5
01000001	A
01000010	B
01000011	C
01000100	D
01000101	E

“code” 문자열이 저장된 1바이트의 메모리 내부



2] 문자와 논리

학습하기

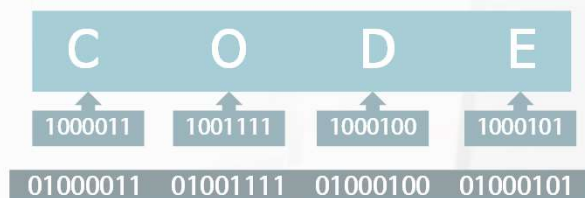
◆ 문자와 코드표

1 아스키코드 (ASCII)

아스키코드 (ASCII) 코드의 예

Code	Character
00110000	0
00110001	1
00110010	2
00110011	3
00110100	4
00110101	5
01000001	A
01000010	B
01000011	C
01000100	D
01000101	E

“code” 문자열이 저장된 1바이트의 메모리 내부



원래 7비트이지만, 실제 한 문자는 8비트인 1바이트에 저장되므로
한 문자의 시작은 0으로 함

2] 문자와 논리

학습하기

◇ 문자와 코드표

2 앱시딕(EBCDIC)

- Extended Binary Coded Decimal Interchange Code
- 8비트**를 사용하여 문자를 표현하는 코드 표준으로 IBM에서 제정한 코드

Zone bits(2bit)	Zone bits(2bit)	Numeric bits
11 : 대문자, 소문자	00 : A ~ Z	
10 : 소문자	01 : J ~ R	
01 : 특수문자	10 : S ~ Z	
00 : 사용하지 않음	11 : 숫자	

To. 교수님
빈칸으로 두는 것이 맞을지
확인 부탁드립니다.

2] 문자와 논리

학습하기

◇ 문자와 코드표

3 유니코드(Unicode)

- 전 세계 모든 언어를 하나의 코드 체계 안으로 통합하기 위하여 만들어진 코드
- 전 세계의 문자를 모두 표현하기 위한 **2바이트인 16비트로 확장된** 코드 체계

2] 문자와 논리



◇ 논리

논리

참(True)과 거짓(False)을 의미하는 두 가지 정보를 표현한 논리 값

하나의 비트 정보

0 1

거짓

참

4

컴퓨터의 연산

1] 정수 연산

학습하기

정수의 연산에서 주의할 것은
정해진 비트를 넘어서는 올림은 무시

◇ 정수의 덧셈

예

$$\begin{array}{r}
 17:00010001 \\
 + 23:00010111 \\
 \hline
 \text{결과 } 40:00101000
 \end{array}$$

1] 정수 연산

학습하기

◇ 정수의 뺄셈

- ✓ 2의 보수를 이용한 음수의 표현 방법을 이용
- ✓ $a - b$ 의 연산은 $a + (-b)$ 로 변환하여 덧셈을 수행

예 $4 - 4 \rightarrow 4 + (-4)$

$$\begin{array}{r}
 0100 \quad \rightarrow \text{4의 4비트 2진수} \\
 + 1100 \quad \rightarrow \text{-4의 4비트 2진수} \\
 \hline
 0000 \quad \rightarrow \text{결과가 0임을 쉽게 알 수 있음,} \\
 \quad \quad \quad \text{4비트 계산이므로 5비트에 1올림은 무시하고} \\
 \quad \quad \quad \text{계산함}
 \end{array}$$

2] 부동소수 연산

학습하기

1 부동소수의 덧셈과 뺄셈은 지수 부분을 두 지수 중에서 큰 수의 지수로 같게 조정하여 가수 부분을 덧셈 또는 뺄셈

2 메모리에 결과 값을 저장하기 전에 다시 정규화하여 저장

◇ 부동소수의 덧셈

예

$$\begin{array}{r}
 2^5 \times 1.1011 \\
 + 2^5 \times 0.01011 \\
 \hline
 2^5 \times 10.00001
 \end{array}
 \xrightarrow{\text{정규화}}
 2^6 \times 1.000001$$

2] 부동소수 연산

학습하기

◇ 부동소수의 뺄셈

✓ 가수의 뺄셈에서 큰 것에서 작은 것을 빼고 큰 것의 부호를 부호부에 이용하면 연산이 간편

예

$$\begin{array}{r}
 2^5 \times 0.01011 \\
 - 2^5 \times 1.1011 \\
 \hline
 -2^5 \times 1.01011
 \end{array}
 \xrightarrow{\text{정규화}}
 \begin{array}{r}
 1.1011 \\
 - 0.01011 \\
 \hline
 -1.01011 \\
 -2^5 \times 1.01011
 \end{array}$$

3] 논리 연산

학습하기

◇ 논리 연산자

단항 연산자

- NOT 연산자
- 하나의 항만이 연산에 참여

이항 연산자

- AND, OR 연산자
- 두 개의 항이 연산에 참여

연산자 NAND

AND 연산의
결과와 반대

연산자 NOR

OR 연산의
결과와 반대

연산자 XOR

항이
서로 다르면 1이고,
같으면 0인 결과

3] 논리 연산

학습하기

◇ 진리표 1 : 논리 연산자 AND, OR, NOT

AND 연산자

X_1	X_2	$X_1 \cdot X_2$
0	0	0
0	1	0
1	0	0
1	1	1

OR 연산자

X_1	X_2	$X_1 + X_2$
0	0	0
0	1	1
1	0	1
1	1	1

NOT 연산자

X_1	X_1'
0	1
1	0

3] 논리 연산



◆ 진리표 2 : 논리 연산자 NAND, NOR, XOR

NAND 연산자			NOR 연산자			XOR 연산자		
X_1	X_2	결과	X_1	X_2	결과	X_1	X_2	결과
0	0	1	0	0	1	0	0	0
0	1	1	0	1	0	0	1	1
1	0	1	1	0	0	1	0	1
1	1	0	1	1	0	1	1	0

3] 논리 연산



◆ 부울 대수

논리회로를 수학적으로 해석하기 위해
영국의 수학자 부울(Boole)이 제창한 기호 논리학의 한 분야



컴퓨터가 정보를 처리하는 방식에 대하여
이론적 배경을 제공함

3] 논리 연산

학습하기

◇ 부울 대수

0과 1의 법칙	$0 + X = X$	$0 \cdot X = 0$	$1 + X = 1$	$1 \cdot X = X$
항등 법칙	$X + X = X$	$X \cdot X = X$		
보수 법칙	$X + X' = 1$	$X \cdot X' = 0$		
부정의 부정	$(X')' = X$			
흡수 법칙	$X + (X \cdot Y) = X$		$X \cdot (X + Y) = X$	
교환 법칙	$X \cdot Y = Y \cdot X$		$X + Y = Y + X$	
결합 법칙	$X + (Y + Z) = (X + Y) + Z$		$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	
분배 법칙	$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$		$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$	
드모르강의 법칙	$(X + Y)' = X' \cdot Y'$		$(X \cdot Y)' = X' + Y'$	

3] 논리 연산

학습하기

◇ 논리 함수(부울 함수)

- 논리적인 문제를 해결하기 위한 수학적 방법
- 이진 논리(Binary logic) 사용

참	거짓
0	1

논리 연산

논리곱(AND)

논리합(OR)

논리부정(NOT)

3] 논리 연산

학습하기

◆ 논리 함수(부울 함수)

◆ 게이트(Gate)

게이트

신호를 나타내는 데 사용되는 논리회로

교환 법칙


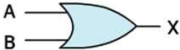
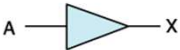
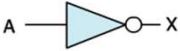
결합 법칙

분배 법칙

4] 논리 회로 설계

학습하기


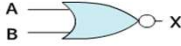
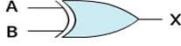

<1/2>

게이트	논리기호	논리함수	진리표															
AND		$X=AB$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$X=A+B$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Buffer		$X=A$	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	X	0	0	1	1									
A	X																	
0	0																	
1	1																	
NOT (Inverter)		$X=A'$	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0									
A	X																	
0	1																	
1	0																	

4] 논리 회로 설계


 학습하기

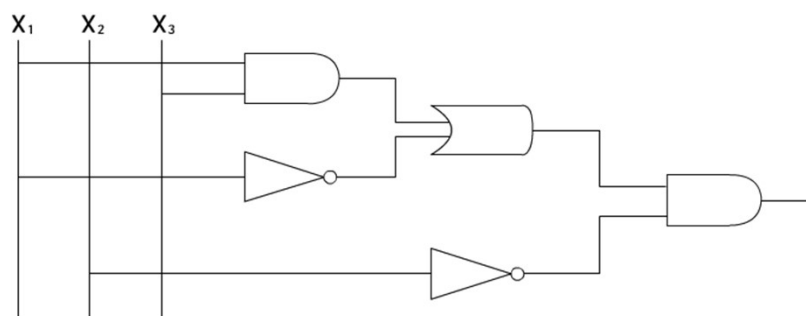
(2/2)

게이트	논리기호	논리함수	진리표															
NAND		$X=A'+B'$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$X=A'B'$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR (Exclusive-OR)		$X=A\oplus B=A'B+AB'$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR (Exclusive-NOR)		$X=A\odot B=A'B'+AB$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

4] 논리 회로 설계


 학습하기

◇ 논리 회로 표현

예 $X_2' \cdot (X_1' + X_1 \cdot X_3)$


4] 논리 회로 설계

학습하기

◇ 부울 함수의 간소화

예1	예2	예3
$F = AB' + B$ $= B + AB'$ $= (B+A)(B+B')$ $= (B+A) \cdot 1$ $= A+B$	$F = A(A'+B)$ $= AA' + AB$ $= 0 + AB$ $= AB$	$F = AB + AC + AB'C'$ $= A(B+C+B'C')$ $= A((B+C)+(B+C)')$ $= A \cdot 1$ $= A$

4] 논리 회로 설계

학습하기

◇ 카노맵(Karnaugh Map)을 이용한 부울 함수의 간소화

		B	0	1
A	0			1
	1	1		1

SUMMARY 정리하기

정리하기

컴퓨터의 자료 표현

- 컴퓨터의 정보 처리 단위 중에서 가장 작은 정보 단위는 Bit

진수와 수의 표현

- 컴퓨터와 관련된 진수
 - 10진수, 2진수, 8진수, 16진수

SUMMARY 정리하기

정리하기

컴퓨터의 정보 종류

- 정수, 부동소수, 문자와 논리 등

컴퓨터의 연산

- 정수의 뺄셈에 보수를 이용함
- 부동소수 연산은 지수를 같게 맞추고 가수부로 연산함
- 논리연산자들은 논리 게이트로 논리회로를 설계할 수 있음

차시예고

정리하기

1주차 2주차 3주차 4주차 5주차 6주차 중간고사

컴퓨터 구조

- 수고하셨습니다.