

소프트웨어 프로젝트 1

정보의 표현과 컴퓨터 하드웨어

2022학년도 1학기

국민대학교 소프트웨어학부

지난 수업에는... 과목 소개

- 전공에 대한 흥미 유발
- 필수 배경 지식 습득



- ✓ 전산 개론
- ✓ Linux
- ✓ Raspberry PI
- ✓ GIT/github



컴퓨터란?

compute: 계산 → “전자계산기” 와 같은 것? 다른 것?



VS



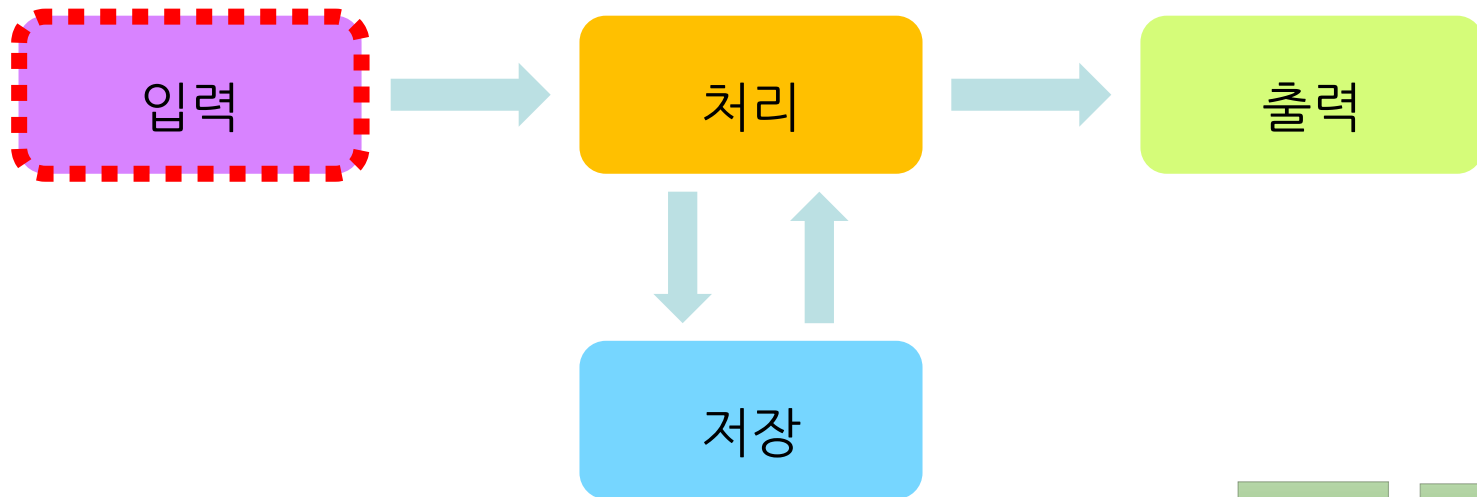
(Wikipedia) A computer is a device that can be instructed to carry out sequences of **arithmetic or logical operations** automatically via **computer programming**.

(번역) 컴퓨터는 컴퓨터 프로그래밍을 통해 자동으로 일련의 산술 또는 논리 연산을 수행하도록 지시 할 수 있는 장치입니다.

컴퓨터가 가지는 기능

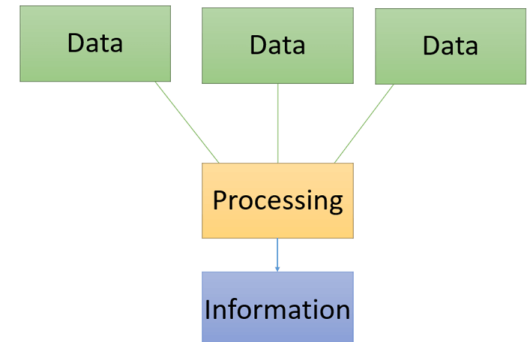
컴퓨터 - 생각하는 기계?

중국어에서는 전뇌 (电脑)라고 부른다고 하던데...



무엇을 입력 받고 처리하고 저장하고 출력하나요?

→ 정보 (information)



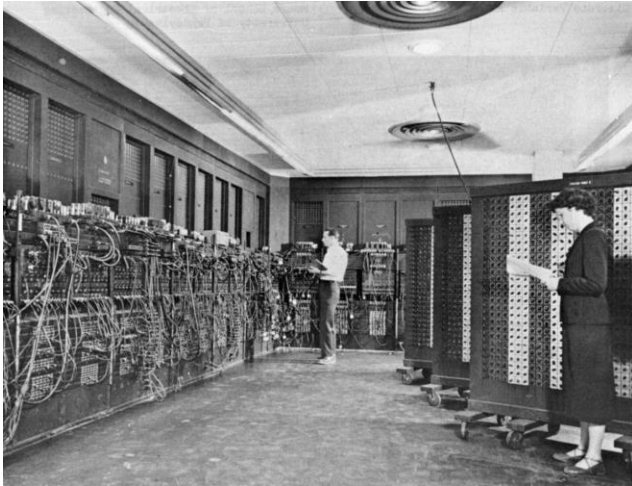
컴퓨터가 가지는 기능

- 어느 기능을 수행하는 장치냐?에 따라서 컴퓨터 하드웨어의 분류가 가능



- 이 기능들의 조합으로 바둑을 두어 사람을 이길 수 있다구요?
 - 실제로, 더 많은 의미 있는 일들을 하도록 시킬 수 있음

컴퓨터의 과거와 현재



- 모습은 많이 달라져 왔지만, 구성 요소는 여전히 (아직은) 앞에서의 분류와 비슷
 - 각 그림에서 중앙처리장치/기억장치/입출력장치는 서로 어떤 모습을 띠고 있나요?

다양한 규모의 컴퓨터



BUT, 그래도 여전히
구성 요소들의 분류는
크게 다르지 않음

컴퓨터 시스템의 설계

어울린다고 여겨지는 대로 선을 이어 보세요!

수조원을 들여서 만든
수십억원짜리 시스템

수억원을 들여서 만든
수십만원짜리 시스템

수백만원을 들여서 만든
수백만원짜리 시스템

노래방 카운터를 위한
계수/집계 시스템

일기예보용 시뮬레이션을 위한
수퍼컴퓨터

차량용 ABS 제어를 위한
내장형 시스템

컴퓨터 시스템의 설계

다음 각 시스템 응용에 따른 (1) 설계 비용과 (2) 생산원가 (판매가) 의 이해득실을 생각해 보세요!



정보의 표현

digit: 수 (數)

아날로그 (analog)

vs

디지털 (digital)



디지털 정보의 표현



켜진 상태

참 (True)

1



꺼진 상태

거짓 (False)

0

컴퓨터 정보의 표현 단위

비트 (bit; binary digit)

컴퓨터에서 디지털회로의 조합으로 정보를 표현할 때 이용되는 가장 작은 단위

컴퓨터에서 표현하는 모든 정보는 비트를 모아서 이루어짐
(즉, 이진수로 표현됨 e.g., 01010011)

바이트 (byte)

8 개의 비트를 모아서 만든 단위 - 왜 하필 “여덟” 개를?

컴퓨터 구조가 영어권에서 발전

영대소문자, 숫자, 기타 기호 등이 8비트로 표현 충분

이진수의 표현

0



1



2



3



한 바이트를 이용해서
표현할 수 있는
가장 큰 수는?

이진수의 표현

십진수 (decimal)

3 6 5

100 ($=10^2$)의 자리 10 ($=10^1$)의 자리 1 ($=10^0$)의 자리

이진수 (binary)

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0



MSB

(most significant bit)

LSB

(least significant bit)

이진 정수의 표현 범위



한 바이트 (8 bits) 를 이용하면 0 ~ 255 ($2^8 - 1$) 의 수를 표현할 수 있음

32 개의 비트로 이루어진 워드 (word) 를 이용하면?

→ $[0, 2^{32} - 1]$ 범위의 십진 정수를 표현할 수 있음

이진법에서 단위의 이용

$1K = 2^{10} = 1,024$ 1,000 (일천)

$1M = 2^{20} = 1,048,576$ 1,000,000 (백만)

$1G = 2^{30}$ 1,000,000,000 (십억)

$1T = 2^{40}$ 1,000,000,000,000 (일조)

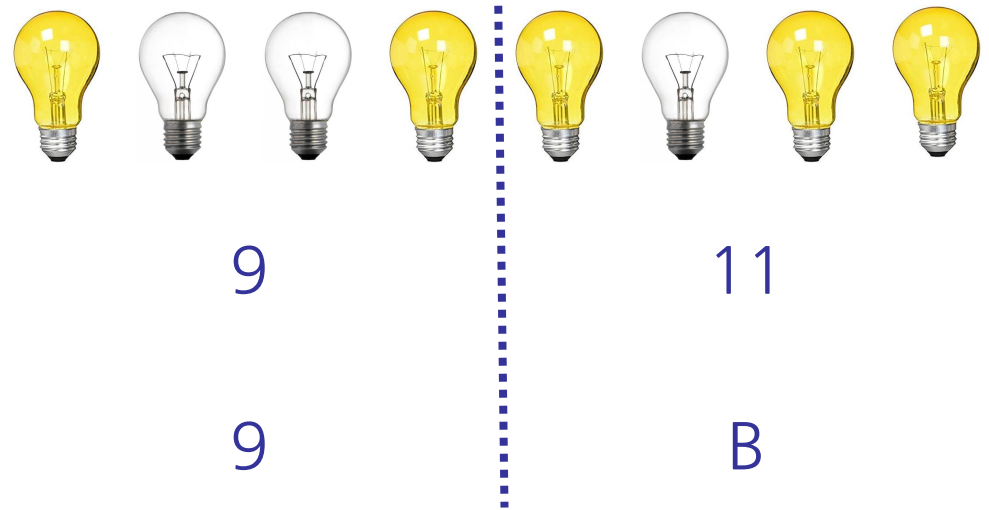
(참고) 32 비트를 이용하여 지정 가능한 서로 다른 주소의 개수

→ 2^{32} 가지 = $2^2 \times 1G$ 가지 = 4G 가지

16 진수 (Hexadecimal)

네 개의 비트 (nibble; [0, 15]) 를 모아 한 자리로 표현

Dec	Hex	Dec	Hex
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F



→ 0X9B = 0x9b = 9bh

진법 변환 연습

10진수 195 를 2진수와 16진수로 표현해 보세요!

(흠, 16진수로 먼저 변환하는 것이 덜 귀찮군?)

$$195 / 16 = 12 \cdots 3$$



C



3

h

→ 0XC3

C							3		
1	1	0	0			0	0	1	1

→ 11000011b

진법 변환 연습

아래 10진 정수들을 2진수와 16진수로 변환해 보세요.

(명시적으로 10진수임을 나타내기 위하여 “d” 를 덧붙이기도 합니다.)

128d

65d

237d

322d

문자의 표현 - 코드 (code)

문자를 표현하기 위해서는,

각 글자 (character) 에 대하여 대응하는 수를 연결하는 표를 이용

ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1111111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(88	58	1011000	130	X					
41	29	101001	51)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					

7 비트의 이진 정수를
영문 대소문자, 숫자, 특수기호 등에
할당한 표준 코드

한글 표현은 어떻게 하나요?
→ (연구과제) Unicode 에 대하여 조사

이진수 연산 연습

[illegible]

2진수 음수의 표현 (참고)

○ 부호 크기 체계 (signed magnitude, sign and magnitude system)

- 첫번째 비트 (부호 비트, sign bit) : 부호 비트 (보통 가장 왼쪽 첫번째 비트, MSB)
 - . '1' -> 음수, '0' -> 양수
- 나머지 비트들 : 절대값 크기 (magnitude)
 - . 표시범위 : $-2^{n-1} \sim 2^{n-1}$

○ 보수 체계 (complement number system, two's complement)

- 2의 보수 (Two's Complement)
 - . 표시범위 : $-2^{n-1} \sim 2^{n-1}-1$
 - .. 컴퓨터 (디지털시스템) 에서 보다 많이 사용
- 1의 보수 (One's Complement)
 - . 표시범위 : $-2^{n-1} \sim 2^{n-1}$

○ 2진 보수 (2의 보수, 1의 보수) 장점

- '음수 표현' 및 '보다 쉽게 연산을 하기' 위함
 - . 컴퓨터는 뺄셈 회로가 없으며, 보수를 취하는 회로 만 있음.
 - .. 보수를 취하고 더하면 뺄셈이 되므로, 굳이 뺄셈 회로를 따로 둘 필요 없음

○ 부호크기체계 및 1의 보수 단점

- 영 (0) 이 '양의 영 (+0)', '음의 영 (-0)' 2개를 동시에 갖으므로 부주의한 프로그래밍 발생 가능성 있음

○ 비교표 (부호크기체계, 2의보수, 1의보수)

10진	부호화 크기체계	2의 보수	1의 보수
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	1000	-	1111
-1	1001	1111	1110
-2	1010	1110	1101
-3	1011	1101	1100
-4	1100	1100	1011
-5	1101	1011	1010
-6	1110	1010	1001
-7	1111	1001	1000
-8	-	1000	-

• 양수는 모두 같음
• MSB(최상위비트)가 '0' -> 양수

• 음수는 달라짐
• MSB(최상위비트)가 '1' -> 음수

(부호)+(크기) 1의 보수에 1을 더함 양수를 반전시킴

음수의 표현 - 2의 보수 (2's complement)

주어진 2진수에 대한 2의 보수

→ 모든 비트를 0 은 1 로, 1 은 0 으로 바꾼 뒤 (1의 보수)

여기에 1 을 더한다.

62	0	0	1	1	1	1	1	0
----	---	---	---	---	---	---	---	---

62의 1의 보수	1	1	0	0	0	0	0	1
-----------	---	---	---	---	---	---	---	---

62의 2의 보수	1	1	0	0	0	0	1	0
-----------	---	---	---	---	---	---	---	---

최상위 비트 (MSB) 는 부호를 나타내고 (0: 양수, 1: 음수)

나머지 비트들을 이용하여 절대값을 나타낸다 (음수의 경우 직접적이지 않음).

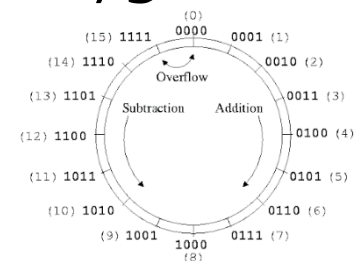
2의 보수를 이용한 뺄셈

$$\begin{array}{r} 135 \\ - 62 \\ \hline \end{array} \quad \begin{array}{r} \overset{1}{} \\ + \\ \hline 0100 \mid 1001 \end{array}$$

$$= 49h = 4 * 16 + 9 = 73$$

(연구과제) 도대체 왜 이게 성립할까?

[힌트] n-bit 를 이용한 “X 의 2의 보수” = $2^n - X$



음수의 표현 - 2의 보수 (2's complement)

주어진 2진수 (75d)에 대한 2의 보수 예

01001011 의 모든 자리의 수를 반전시킨다.

10110100 (이것을 1의 보수라고 부른다)

여기에 1을 더한다.

10110100 --> 180

+) 00000001 --> 1

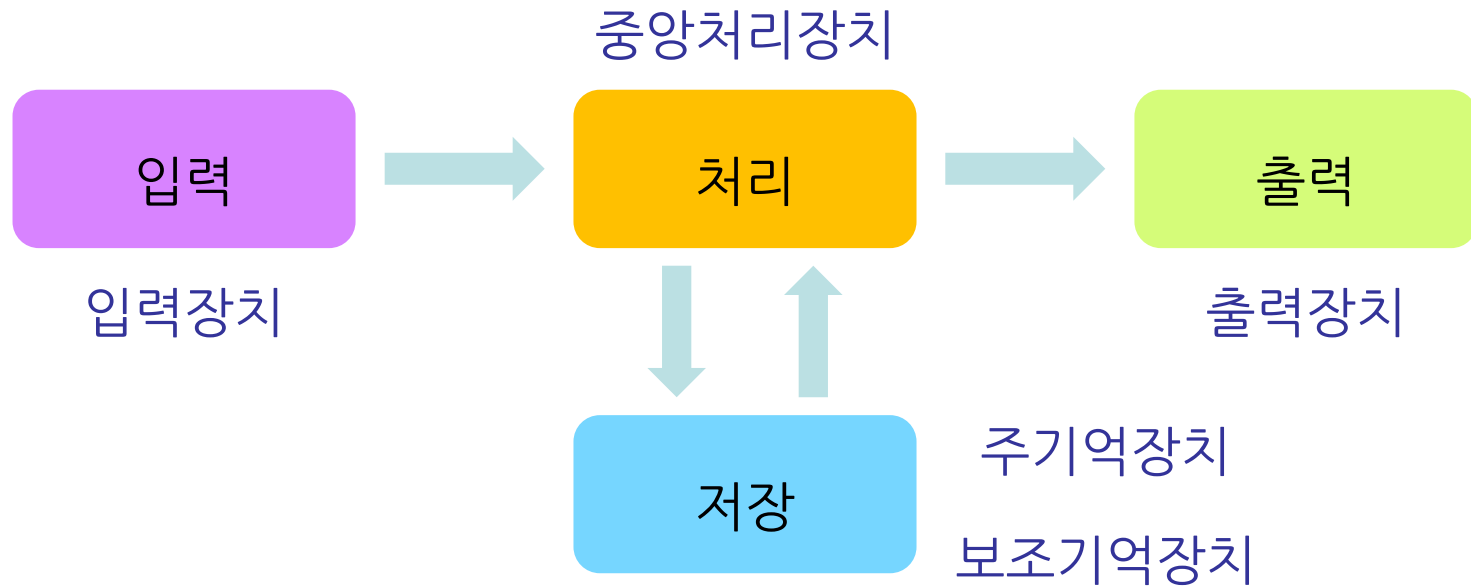
10110101 --> 181

100000000 (8+1=9자리) --> 256

-) 01001011 (8자리) ---> 75

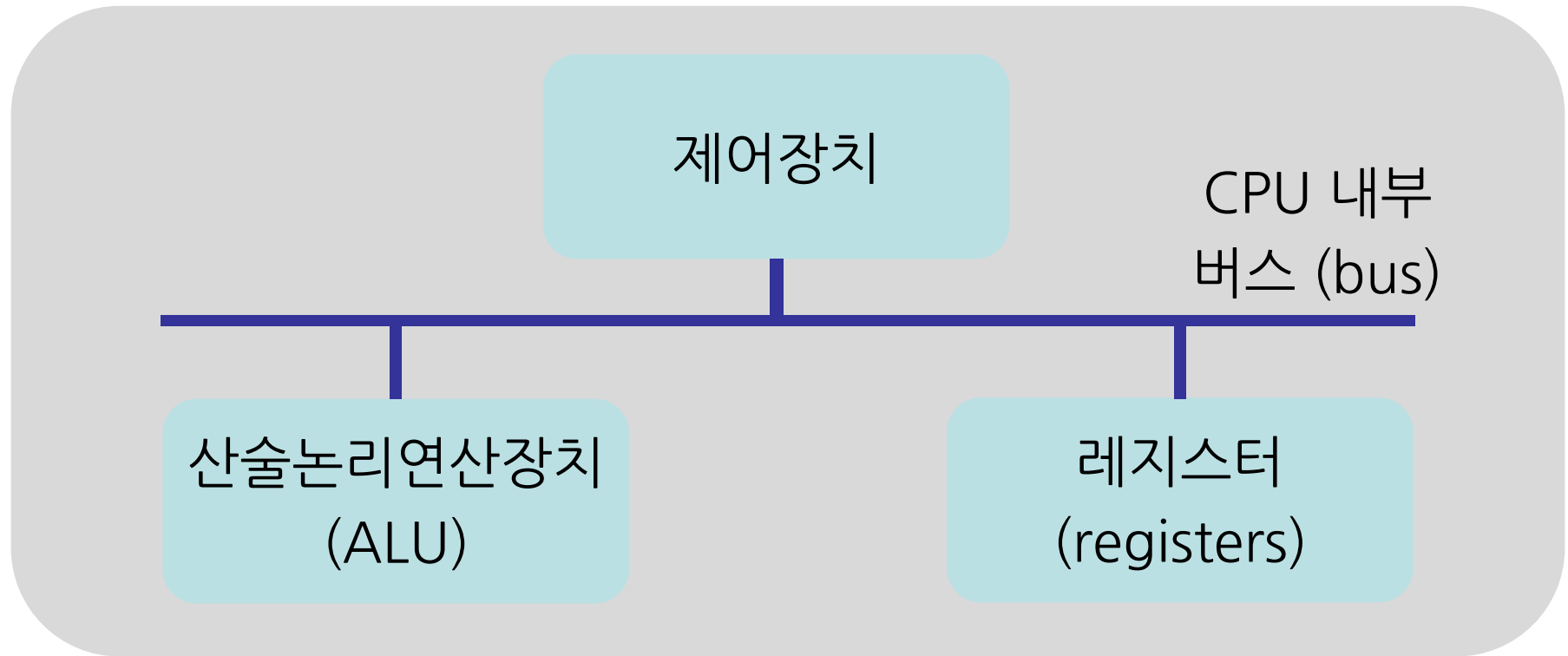
= 10110101 (8자리) ---> 181

컴퓨터 하드웨어의 구성



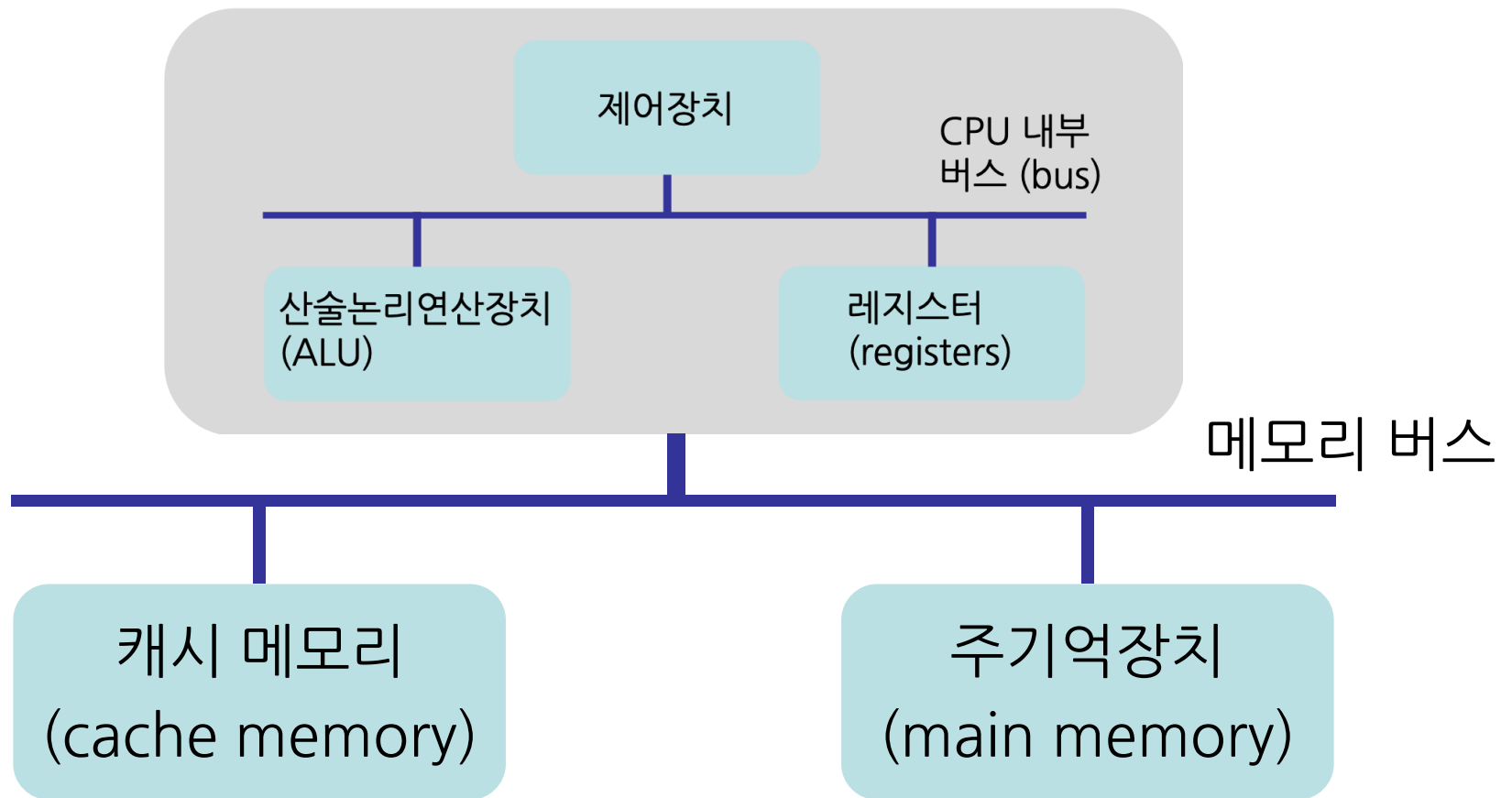
- 중앙처리장치 - CPU (central processing unit)
- 주기억장치 - memory (RAM / ROM)
- 보조기억장치 - storage (HDD, SSD, optical discs, magnetic tapes, ...)
- 입출력장치 - I/O devices (keyboard, mouse, display, printer, network, ...)

중앙처리장치 (CPU)



- 제어장치 - 명령어를 해석하여 그 실행에 필요한 제어 신호를 발생시킴
- ALU - 산술 연산 및 논리 연산 등을 수행
- 레지스터 - 연산의 대상 (operand) 및 연산 결과 등을 저장

주기억장치



메모리의 분류

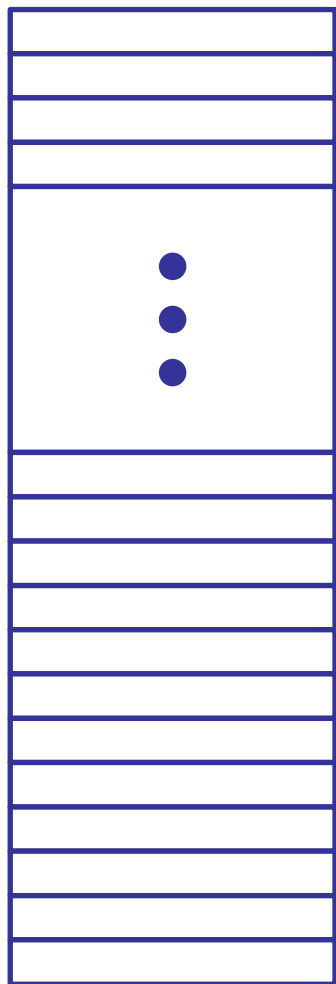
- RAM (random access memory)
 - 읽기/쓰기가 가능하고 임의 접근이 가능하지만
 - 전원 공급이 중단되면 기억된 내용이 사라짐
- ROM (read-only memory)
 - 전원이 중단되어도 내용이 유지되지만
 - 읽기만 가능하고 데이터를 갱신할 수 없음
- Flash memory
- NV(no-volatile)-RAM

메인 메모리로
이용

코드 메모리 등에
이용

메모리 주소 공간

0x FFFF FFFF



0x 0000 0003

0x 0000 0002

0x 0000 0001

0x 0000 0000



메모리 내의 매 바이트 (byte) 마다
독립된 (그리고 연속된) 주소를 부여하는 것이
일반적인 컴퓨터 메모리 조직

(예) 32-bit 주소를 이용한다면
얼만큼의 메모리에 주소를 지정할 수 있나요?

보조기억장치



HDD (hard disk drive)

자기 디스크를 이용한 데이터 저장 장치

회전하는 디스크 위에 헤더를 움직여 데이터 읽고 쓰기



SSD (solid state drive)

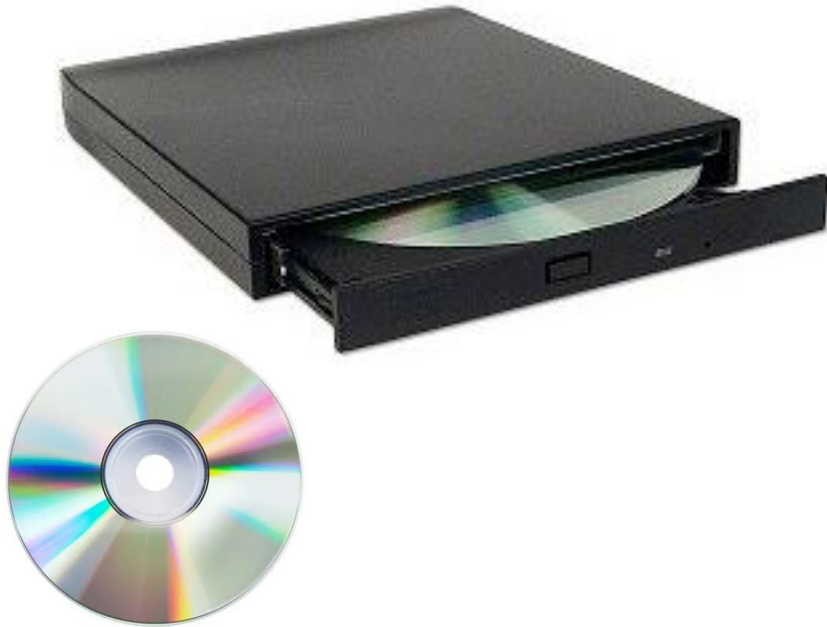
반도체 (플래시 메모리) 를 이용한 데이터 저장 장치

기구 부품이 없다는 점에서 HDD 보다 좋은 특성

비교

- 데이터 입출력 속도
(처리율, 응답시간)
- 비트당 가격
- 안정성/내구성

보조기억장치



ODD (optical disk drive)

회전하는 광 디스크에 저장된 데이터를

렌즈가 부착된 헤드를 이용하여 읽음 (또는 씀)



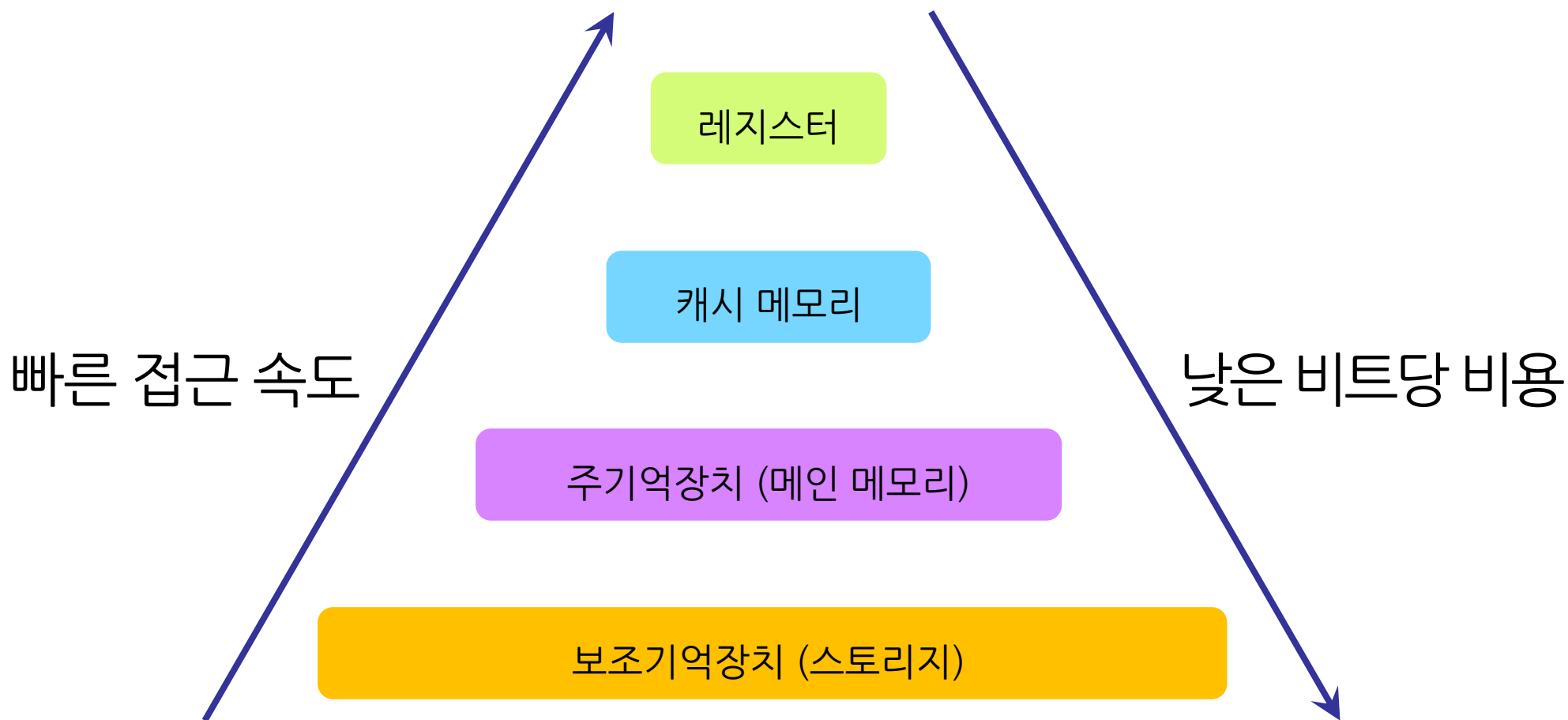
tape drive

자기 테이프에 데이터를 기록

임의 접근 (random access) 이 느려서

주로 백업 데이터 저장 용도로 사용

계층적 메모리 구조



입출력장치



생각해 볼 과제

- 컴퓨터에서는 반드시 이진법을 이용해야만 할까?
- 2의 보수 체계는 어떤 원리를 가지고 있는가?
- 정수 이외의 수를 나타내는 데에는 어떤 방법이 이용될까?
 - 부동소수점 (floating point) 표현 방식
 - 고정소수점 (fixed point) 표현 방식
- 이미지, 사운드 등은 컴퓨터에서 어떻게 표현되는 것일까?

제출할 과제

- 컴퓨터 시스템의 설계 (개발) 비용과 제조 원가
- 디지털 정보와 아날로그 정보의 비교
- 진법변환 문제 풀이
- 유니코드 (Unicode) 란 무엇인가
- 2의 보수 체계를 이용한 뺄셈의 원리를 수식으로