



컴퓨터구조

Computer System Architecture

02

컴퓨터 정보의 표현과 저장

학습 내용

- 01 정보의 표현과 저장
- 02 진법 변환
- 03 보수의 개념
- 04 데이터의 2진수 표현
- 05 문자 데이터의 표현

학습 목표

- 컴퓨터의 정보 표현과 저장 방법에 대해 설명할 수 있다.
- 컴퓨터의 정보를 표현하는 진법 변환에 대해 설명할 수 있다.
- 2진수의 음수 및 데이터의 표현 방법을 설명할 수 있다.
- 문자를 2진수로 표현하는 방법을 설명할 수 있다.

지난시간 돌아보기

지/난/시/간/의/ 학/습/내/용

컴퓨터의 구성 장치와 기본구조

컴퓨터 구조의 발전 과정

컴퓨터의 분류

지난시간 돌아보기

컴퓨터의 구성요소

✓ 하드웨어

- 중앙처리장치 :
자료를 처리하는 일련의 과정을 수행
- 중앙처리장치의 중요 구성 요소 :
산술 논리 연산 장치, 제어장치, 레지스터

✓ 소프트웨어

- 정보의 이동 방향과 정보처리의 종류를 지정하고
그러한 동작들이 일어나는 시간을 지정하는
명령들의 집합
- 구분 : 시스템 소프트웨어, 응용 소프트웨어

지난시간 돌아보기

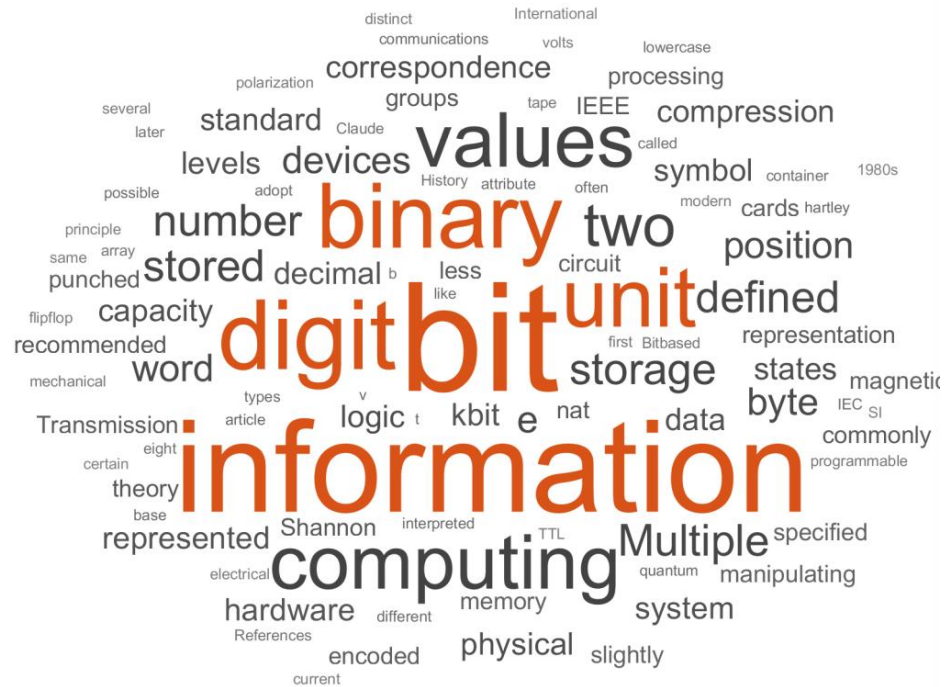
컴퓨터의 세대별 발전 과정

	1세대 컴퓨터	2세대 컴퓨터	3세대 컴퓨터
하드웨어 특징	진공관	트랜지스터	집적회로
소프트웨어 특징	일괄처리	다중프로그래밍 온라인실시간처리	시분할 처리
	4세대 컴퓨터	5세대 컴퓨터	
하드웨어 특징	LSI	VLSI	
소프트웨어 특징	인공지능 전문가 시스템	인공지능 병렬처리 자연언어처리	



생각
해보기

컴퓨터 나라에서 정보는 어떻게 표현할까요?





01

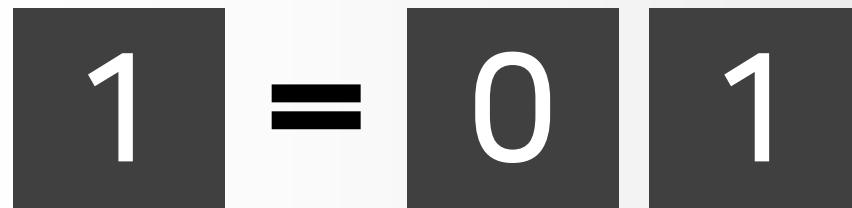
정보의 표현과 저장

- 1) 컴퓨터 정보
- 2) 프로그램 코드
- 3) 프로그램 언어 번역 소프트웨어

1) 컴퓨터 정보

- 컴퓨터 정보

▶ 2진수 비트들로 표현된 프로그램 코드와 데이터



bit

두 개의 숫자 표시

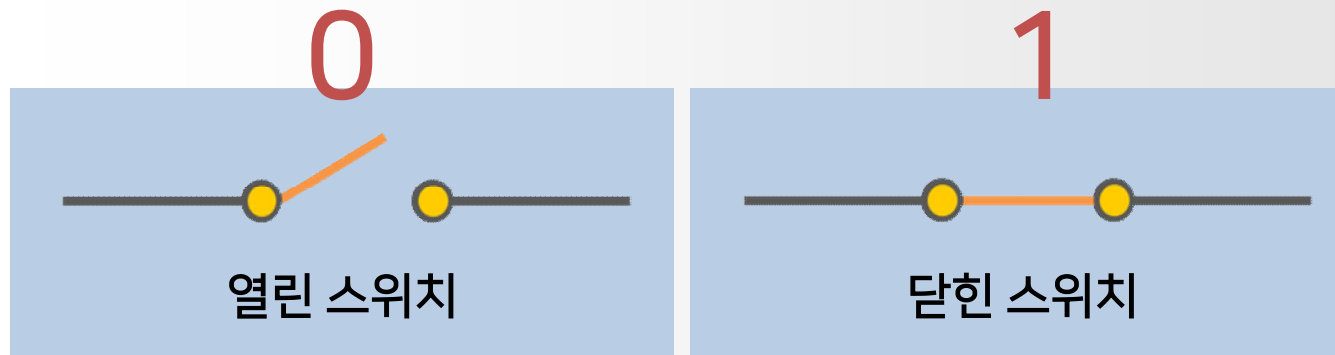
$$\text{사용 가능한 2진수 조합} = 2^n = \text{비트의 수}$$

1) 컴퓨터 정보

■ 컴퓨터 정보

▶ 비트 수가 증가하면, **정보의 양과 저장해야 할 데이터의 양 증가**

▶ 0은 열린 스위치로, 1은 닫힌 스위치로 표현 가능



2) 프로그램 코드

▪ 기계어, 어셈블리 언어, 고급 언어

기계어 (machine language)

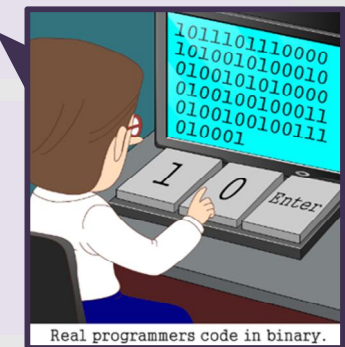
- 기계 코드 (Machine code)
- 컴퓨터 하드웨어 부품들이 이해할 수 있는 언어
- 2진 비트로 구성

어셈블리 언어 (Assembly language)

- 고급 언어와 기계어 사이의 중간 언어
- 어셈블러 (Assembler)로 번역시, 기계어와 일대일 대응

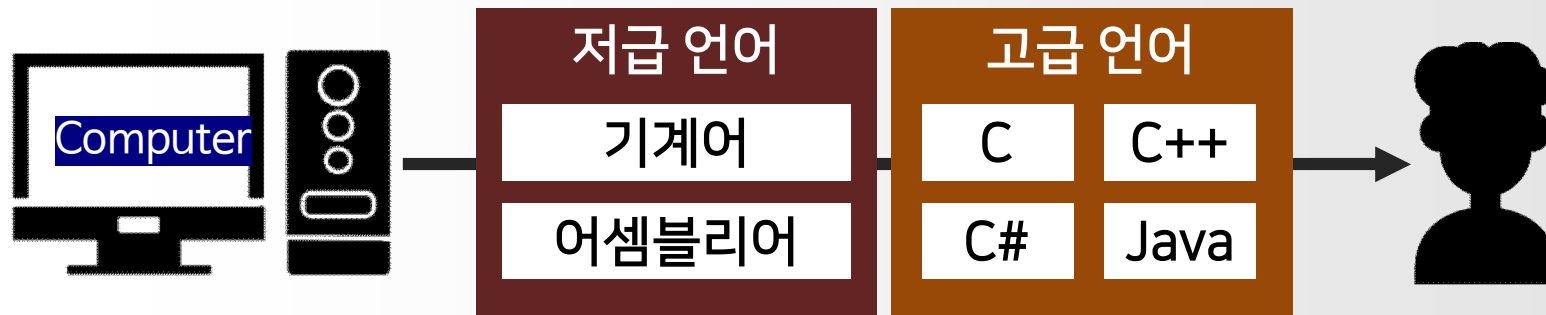
고급 언어 (High-level language)

- 영문자, 숫자로 구성되어 사람이 이해하기 쉬운 언어
- C, C++, PASCAL, FORTRAN, COBOL 등
- 컴파일러 (compiler)를 이용하여 기계어로 번역



2) 프로그램 코드

- 기계어, 어셈블리 언어, 고급 언어

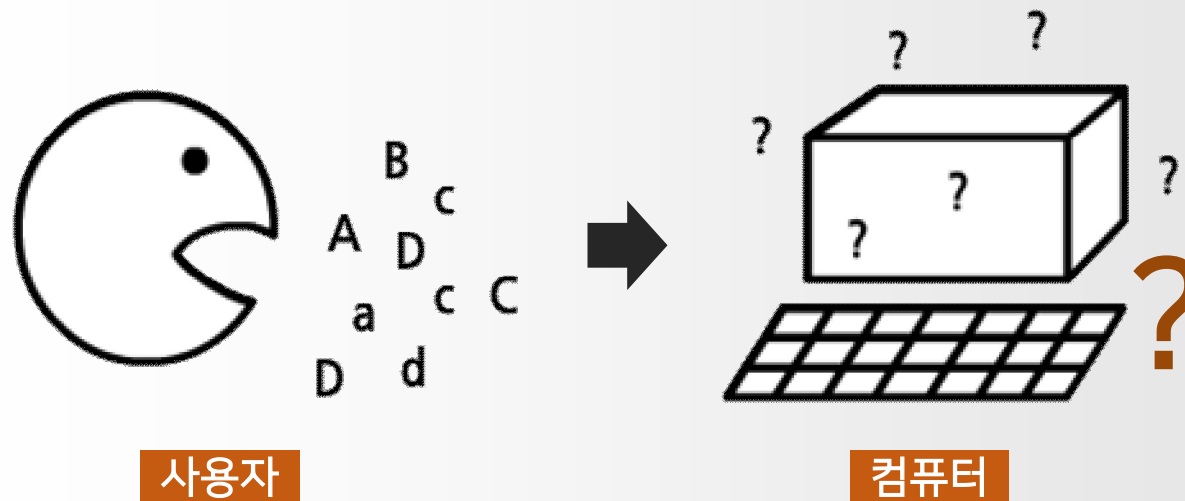


2) 프로그램 코드

- 컴파일러의 역할

컴파일러가 없을 때

컴파일러가 있을 때

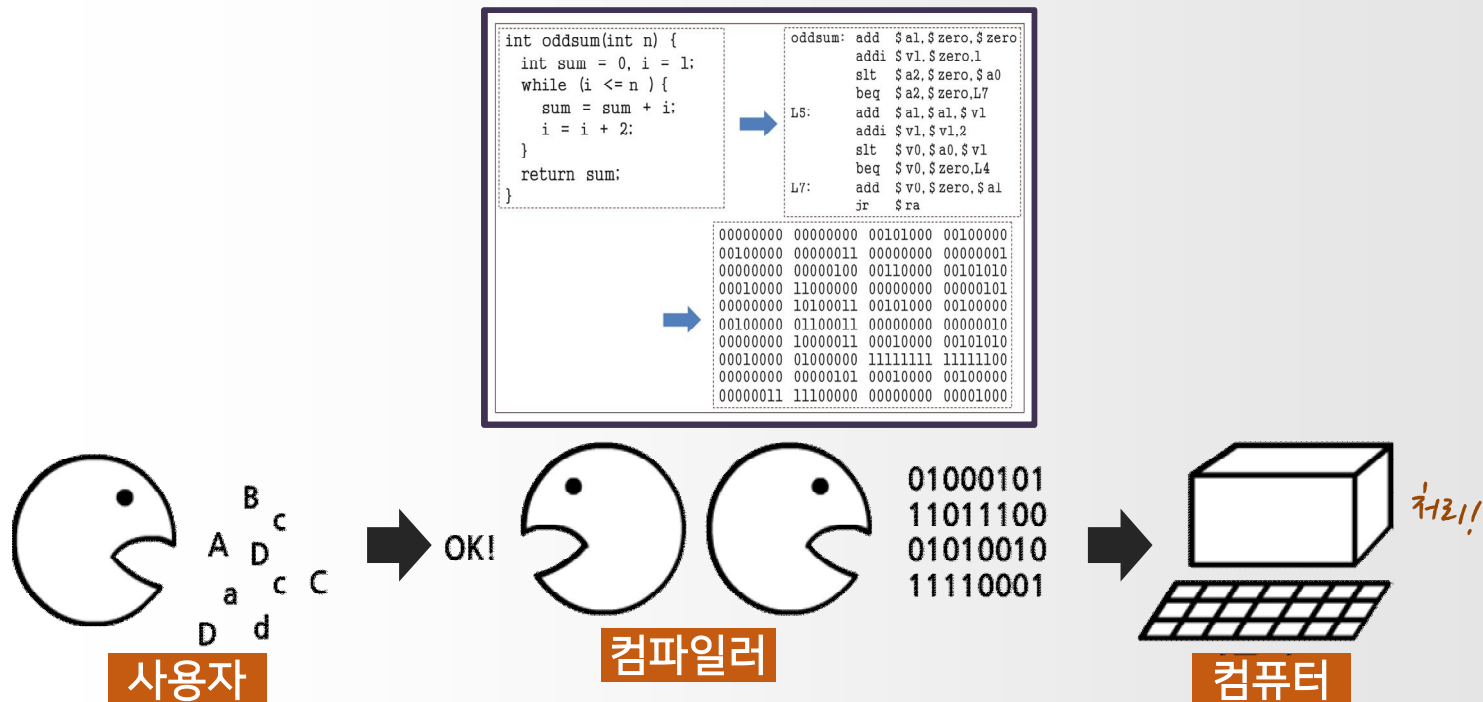


2) 프로그램 코드

컴파일러의 역할

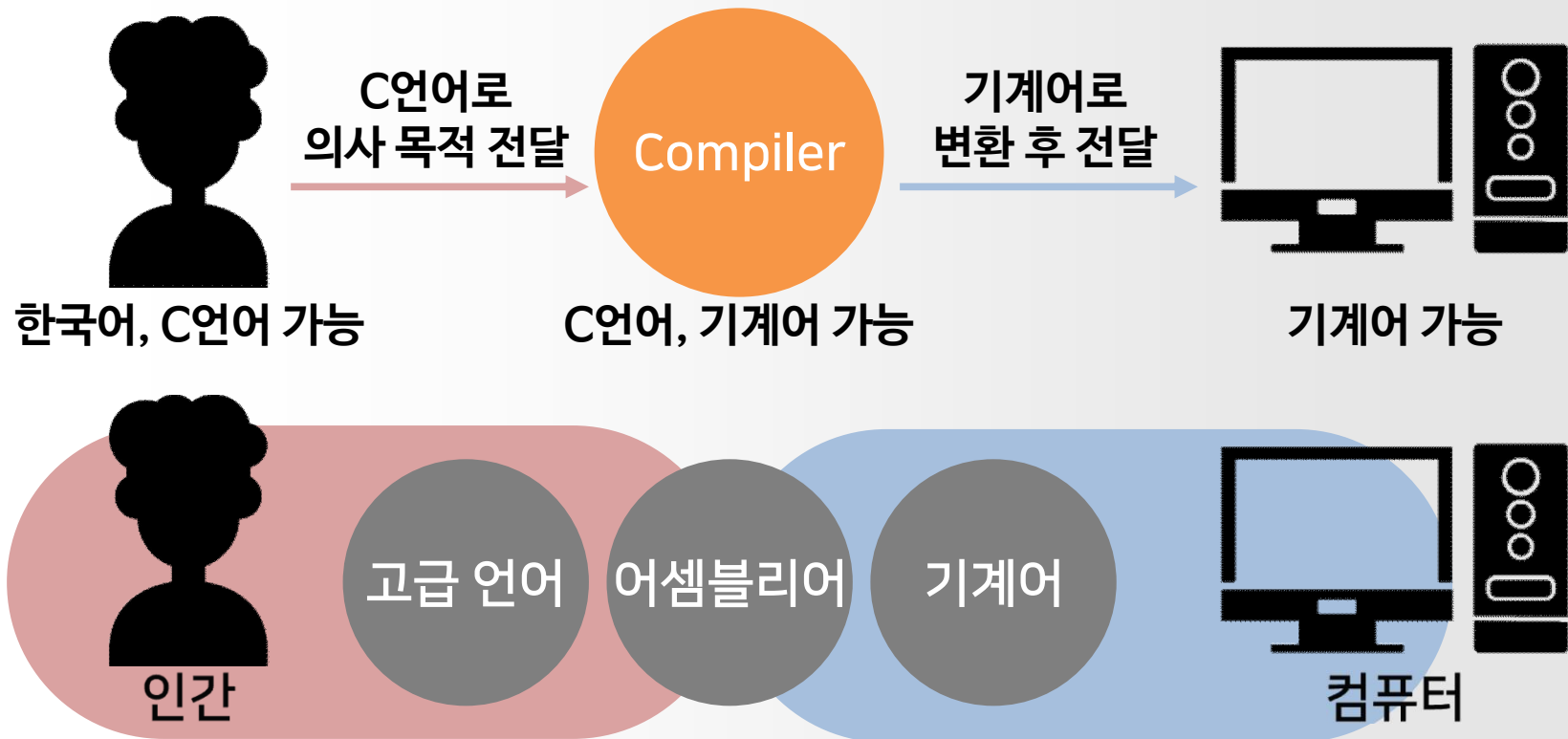
컴파일러가 없을 때

컴파일러가 있을 때



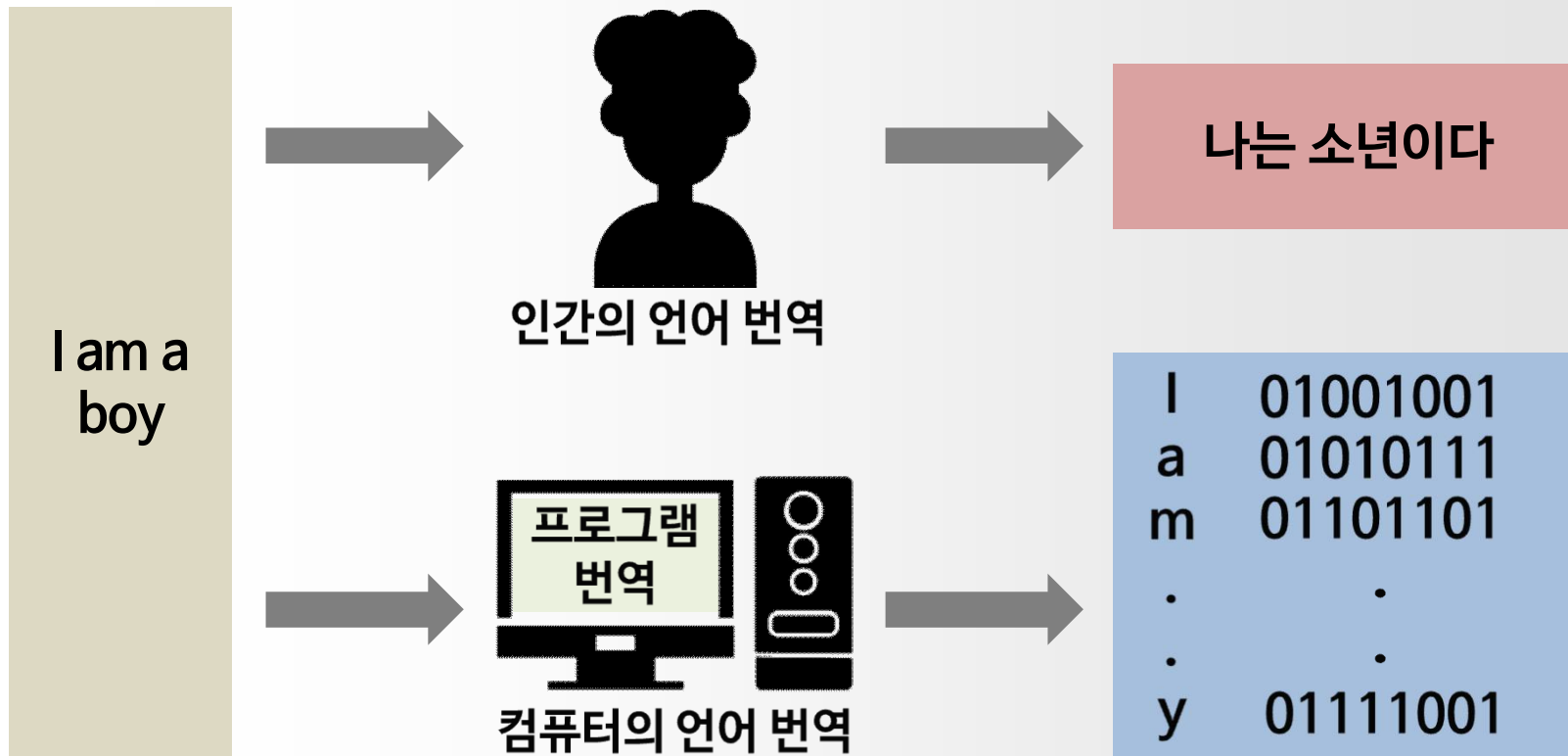
2) 프로그램 코드

■ 컴파일러의 역할



2) 프로그램 코드

- 인간과 컴퓨터의 언어 번역



(예) 고급 언어로 작성된 $Z = X + Y$ 의 컴퓨터 인식

$Z = X + Y$

LOAD A, X

기억장치 X번지의 내용을 읽어서,
레지스터 A에 적재 (Load)

ADD A, Y

기억장치 Y번지 내용을 읽어서, 레지스터 A에
적재된 값과 더하고, 결과를 다시 A에 적재

STOR Z, A

그 값을 기억장치 Z 번지에 저장 (Store)

고급 언어
프로그램

• $Z = X + Y$

어셈블리
프로그램

• LOAD A, X
• ADD A, Y
• STOR Z, A



기계어 프로그램

• 00100101
• 10000110
• 01000111



3) 프로그램 언어 번역 소프트웨어

- 컴파일러 (Compiler)와 어셈블러 (Assembler)

컴파일러 (Compiler)	어셈블러 (Assembler)
<ul style="list-style-type: none">고급 언어 프로그램을 기계어 프로그램으로 번역	<ul style="list-style-type: none">어셈블리 프로그램을 기계어 프로그램으로 번역니모닉스 (Mnemonics)<ul style="list-style-type: none">어셈블리 명령어가 지정하는 연산을 가리키는 알파벳 기호'LOAD', 'ADD', 'STOR' 등
 <pre>graph LR; A[C언어] --> B[컴파일러]; B --> C[기계어];</pre>	 <pre>graph LR; A[어셈블리어 프로그램] --> B[어셈블러]; B --> C[기계어, 로더 위한 정보들]; D[데이터베이스] --- B;</pre>

3) 프로그램 언어 번역 소프트웨어

- 컴파일러, 인터프리터, 어셈블러

컴파일러	인터프리터	어셈블러
<ul style="list-style-type: none">고급 언어로 작성한 원시프로그램을 기계어로 번역하여 목적프로그램으로 생성하는 프로그램각종 고급 언어마다 고유의 컴파일러 존재	<ul style="list-style-type: none">원시 프로그램을 구성하는 각각의 명령문을 한 줄씩 변환시키면서 즉시 시행하는 PGM BASIC, LISP, APL 등의 언어	<ul style="list-style-type: none">어셈블리어로 작성한 프로그램을 기계어로 번역



02

진법 변환

- 1) 컴퓨터에서 정보 표현
- 2) 수의 진법

1) 컴퓨터에서 정보 표현

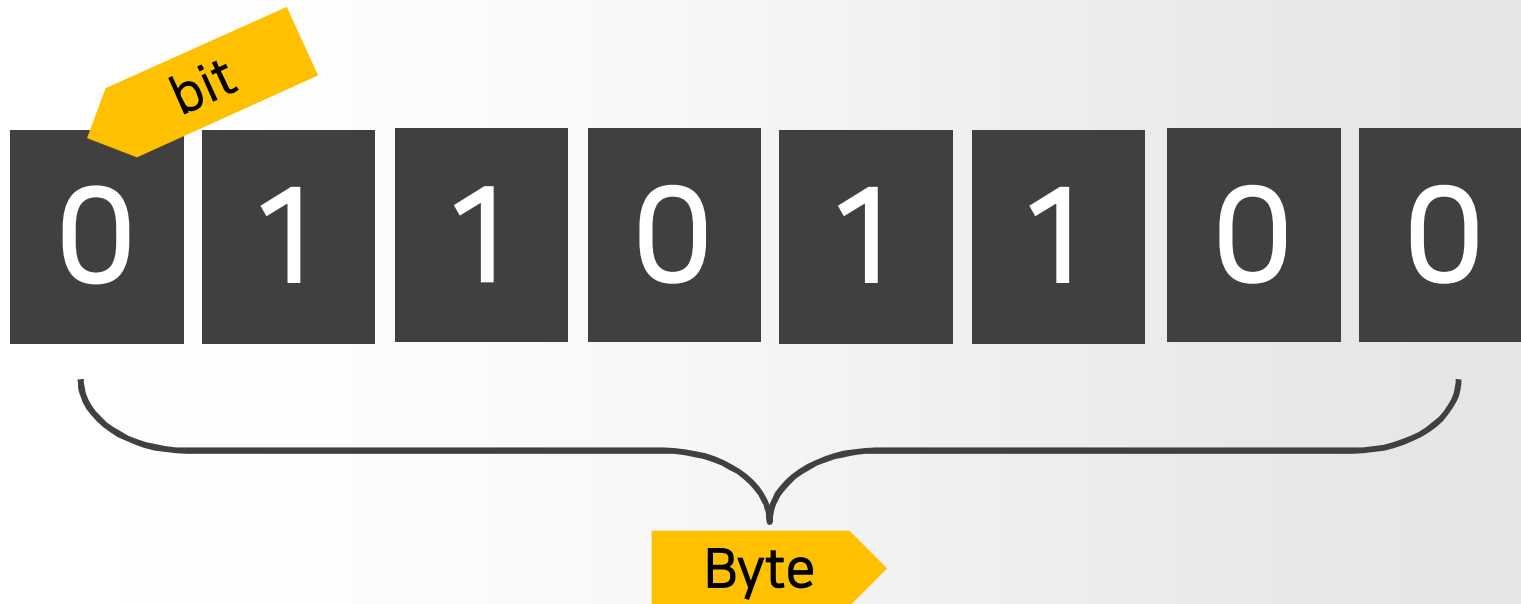
▪ 비트, 바이트, 워드

컴퓨터에서는 데이터 **1비트를 기본으로**
0, 1 두 개의 숫자를 표시하는 2진법을 사용

비트(Bit)	<ul style="list-style-type: none">• 2진수에서 데이터를 표현하는 단위• 2진수의 조합은 2^n만큼의 조합을 가질 수 있음 (n : 비트의 수)
바이트(Byte)	<ul style="list-style-type: none">• 정보처리를 위해 사용되는 비트의 집합• 8Bit를 1Byte로 규정함
워드(Word)	<ul style="list-style-type: none">• 컴퓨터가 한 번에 처리할 수 있는 데이터의 양• 컴퓨터 종류에 따라 구성됨(2바이트, 4바이트 등)• 2비트(4바이트)가 가장 많이 쓰임

1) 컴퓨터에서 정보 표현

- 비트, 바이트, 워드



2진수의 하나의 자리 수, 비트가 8개 모인 것

1) 컴퓨터에서 정보 표현

- 비트당 사용 가능한 2진수의 조합 = 2^n (n : 비트의 수)

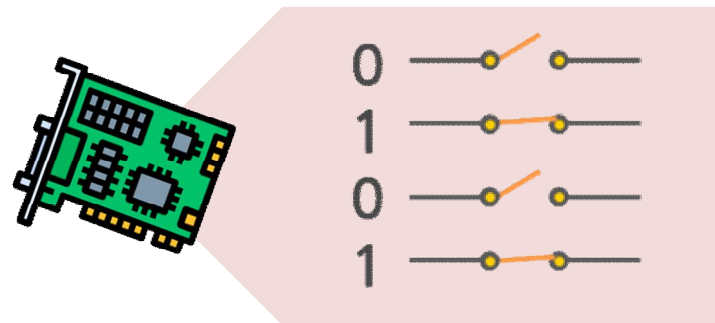
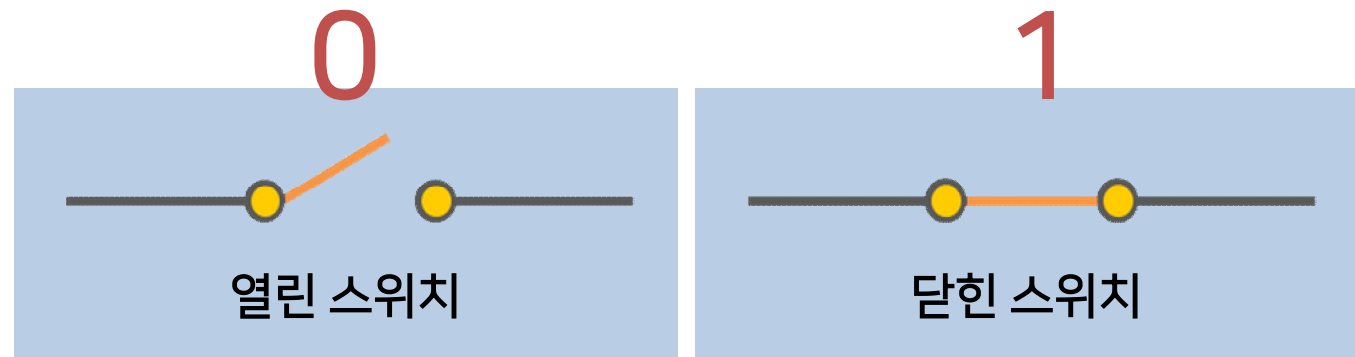
비트의 수	사용 가능한 2진수 조합
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256

1) 컴퓨터에서 정보 표현

- 디지털 정보의 표현 단위(비트)

이름	약어	크기
Kilo	K	$2^{10} = 1,024$
Mega	M	$2^{20} = 1,048,576$
Giga	G	$2^{30} = 1,073,741,824$
Tera	T	$2^{40} = 1,099,511,627,776$

비트는 컴퓨터 내부에서 구체적으로 어떻게 표현되는가?



2) 수의 진법

■ 진법에 따른 수의 변환

10진법	2진법	8진법	16진법
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A

10진법	2진법	8진법	16진법
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2) 수의 진법

- 10진법 (Decimal notation)과 2진법 (Binary notation)

10진법 (Decimal notation)

- 0~9사이의 10개의 숫자를 사용하여 정보를 표현

10진수 $(431)_{10}$ 의 표현

$$(431)_{10} = 4 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$$

10^5	10^4	10^3	10^2	10^1	10^0
0	0	0	4	3	1

2진법 (Binary notation)

- 0과 1을 사용하여 정보를 표현

2진수 $(0111)_2$ 의 표현

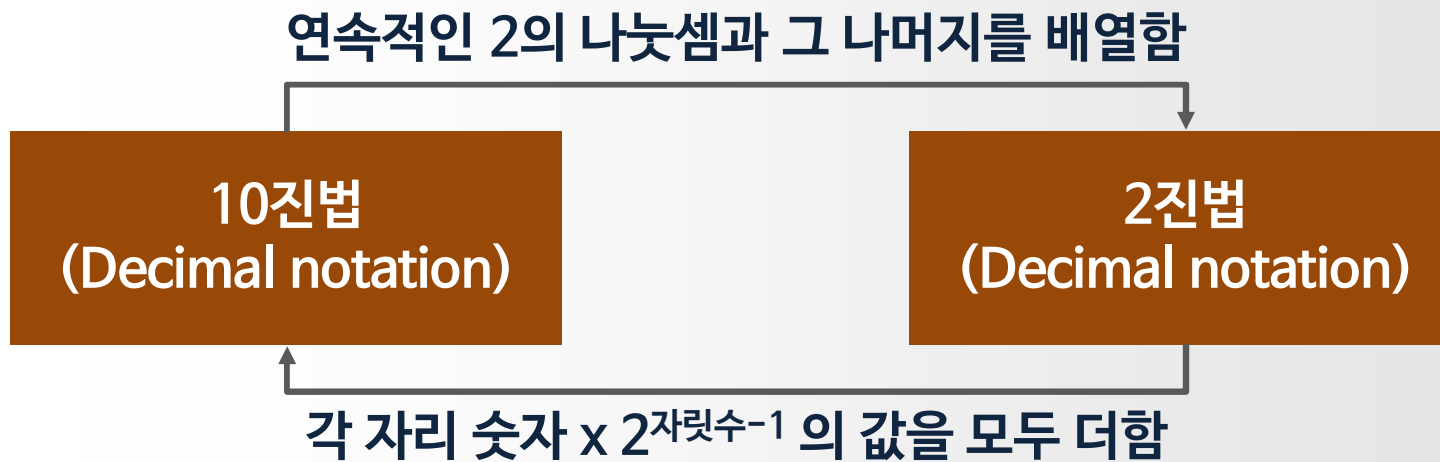
$$(0111)_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	1	1

2) 수의 진법

- 10진법 (Decimal notation)과 2진법 (Binary notation)

▶ 각 진법간의 변환



예제

$(431)_{10}$ 을 2진법의 수로 변환해 보세요.

2	431	
2	251	...1
2	107	...1
2	53	...1
2	26	...1
2	13	...0
2	6	...1
2	3	...0
	1	...1

2진법의 수로 변환 ► $(110101111)_2$



예제

$(11001011001)_2$ 을 10진법의 수로 변환해 보세요.

$$\begin{aligned}(11001011001)_2 &= 1 \times 2^{10} + 1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 \\ &\quad + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1024 + 512 + 64 + 16 + 8 + 1 \\ &= 1625\end{aligned}$$

10진법의 수로 변환 ► $(1625)_{10}$



예제

$(2613)_8$ 을 10진법의 수로 변환해 보세요.

$$\begin{aligned}(2613)_8 &= 2 \times 8^3 + 6 \times 8^2 + 1 \times 8^1 + 3 \times 8^0 \\ &= 1024 + 394 + 8 + 3 \\ &= 1419\end{aligned}$$

10진법의 수로 변환 ► $(1419)_{10}$



예제

$(314)_{10}$ 을 8진법의 수로 변환해 보세요.

$$\begin{array}{r|l} 8 & 314 \\ \hline 8 & 39 \cdots 2 \\ & 4 \cdots 7 \end{array}$$

8진법의 수로 변환 ► $(472)_8$





03

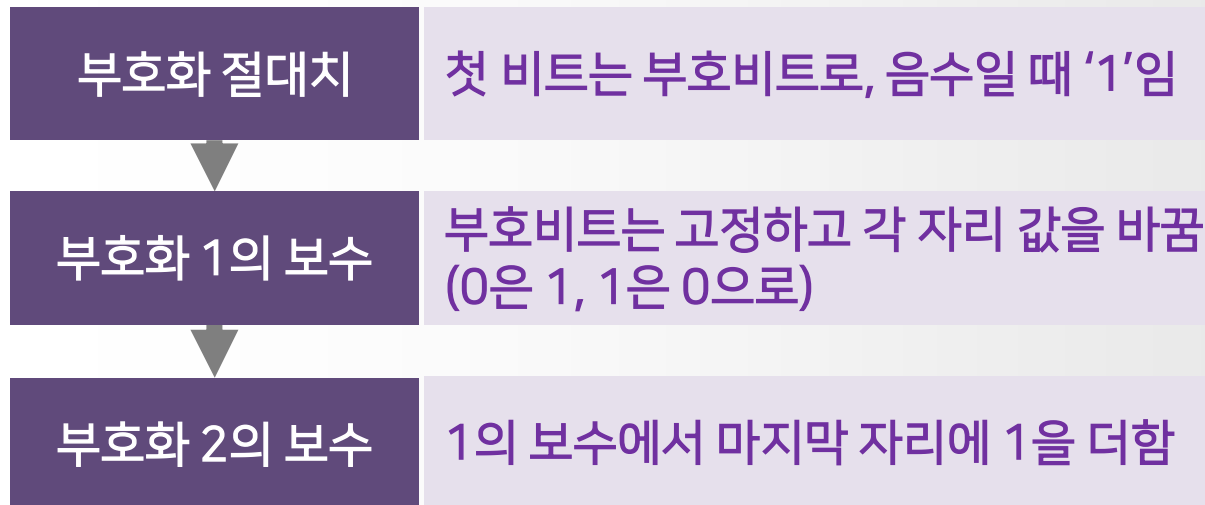
보수의 개념

1) 보수의 개념

1) 보수의 개념

- 보수란?

▶ 컴퓨터에서 음수를 표현할 때 사용



예제

$(-14)_{10}$ 를 8Bit로 표현해 보세요.

2	14	
2	7	...0
2	3	...1
	1	...1

① 2진수로 변환 ▶ $(1110)_2$

② 8비트로 표현

0 0 0 0 1 1 1 0

▶ 부호비트

부호화 절대치

1 0 0 0 1 1 1 0

첫 비트는 부호비트로,
음수일 때 '1'임

부호화 1의 보수

1 1 1 1 0 0 0 1

부호비트는 고정하고,
각 자리 값을 바꿈

부호화 2의 보수

1 1 1 1 0 0 1 0

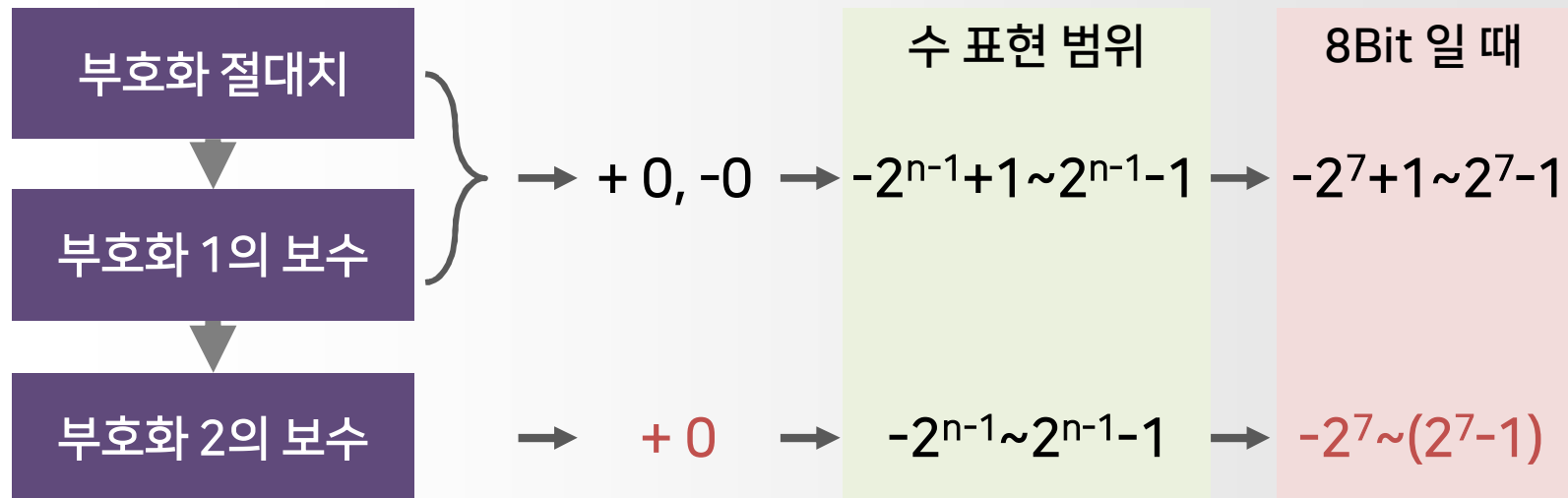
마지막 자리에 1을 더함



1) 보수의 개념

■ 보수란?

▶ 컴퓨터에서 음수를 표현하는 방법



- 서로 상반되는 수(보수)를 이용하면 가산기(덧셈)을 이용한 뺄셈 가능

1) 보수의 개념

- r 진법에서 $(r-1)$ 의 보수

▶ A라는 수에 B라는 수를 더한 결과값의 각 자리가 $(r-1)$ 이 될 때, B를 A에 대한 $(r-1)$ 의 보수라고 함

예 $(237)_{10}$ 에 대한 9의 보수를 B라고 가정할 때, B의 값은 얼마인가?

$$237 + B = 999$$

$$\Leftrightarrow 237 + B = (1000 - 1)$$

$$\Leftrightarrow B = (1000 - 1) - 237 = 762$$

1) 보수의 개념

- r진법에서 r의 보수

▶ A라는 수에 B라는 수를 더해서 각 자리마다 자리올림이 발생하고 해당 자리는 0이 될 때, B를 A에 대한 r의 보수라고 함

예 (237)₁₀에 대한 10의 보수를 B라고 가정할 때, B의 값은 얼마인가?

$$237 + B = 1000$$

$$\Leftrightarrow B = 1000 - 237 = 763$$

1) 보수의 개념

- 부호가 없는 2진수의 뺄셈 연산에서 보수의 활용

▶ 컴퓨터에서 뺄셈 연산은 보수를 이용하는 것이 효율적!

예 $8 - 6 = 2$ 의 뺄셈 연산에서 보수의 활용

감수(빼는 수)
↓
 $8 - 6 = 2$
↑
피감수(빼어지는 수)

$$8 + 4 = 12$$

① 감수를 10의 보수로 표현하면

뺄셈 연산은 덧셈 연산으로 대체할 수 있음

6을 9의 보수로 표현

$$(10 - 1) - 6 = 3$$

6을 10의 보수로 표현

$$3 + 1 = 4$$

② 10의 자리는 버리고 1의 자리만 취함

예제

보수를 이용해 $(1011)_2 - (0100)_2$ 를 계산해보세요.

① 감수 0100을 2의 보수로 표현

0100을 1의 보수로 표현 $(10000 - 00001) - 0100 = 1111 - 0100 = 1011$

0100을 2의 보수로 표현 $1011 + 0001 = 1100$

② 구해진 2의 보수를 피감수와 덧셈을 수행

$$1011 + 1100 = 10111$$

★ 자리올림으로 발생한 최상위 자리의 값을 버리고 나머지 값들을 취해야 함! ★

$$10111 \blacktriangleright 0111$$





04

데이터의 2진수 표현

- 1) 데이터 표현 방법
- 2) 소수의 표현

1) 데이터 표현 방법

- 일반적인 디지털 장치에서 데이터의 표현

- ▶ 2진수로 양의 정수, 음의 정수, 소수를 표현

- ▶ 2진수는 0, 1, 부호 및 소수점의 기호를 이용하여 수를 표현

예제

다음 값을 2진수로 나타내 보세요.

$$(-137.625)_{10}$$

2	137	
2	68	...1
2	34	...0
2	17	...0
2	8	...1
2	4	...0
2	2	...0
2	1	...0
	0	...1

자리올림수

$$\begin{array}{rcl}
 0.625 \times 2 = 1.25 & \dots\dots 1 & \text{(최상위 비트)} \\
 0.25 \times 2 = 0.50 & \dots\dots 0 & \\
 0.50 \times 2 = 1.00 & \dots\dots 1 & \text{(최하위 비트)}
 \end{array}$$

최하위 비트
(LBS)

1 0 0 0 1 0 0 1
 ↑ ↑
 최상위 비트 최하위 비트
 (MBS) (LBS)

1 0 1
 ↑
 최상위 비트
 (MBS)

$$= (-10001001.101)_2$$



2) 소수의 표현

■ 고정소수점 (Fixed-point)과 부동소수점의 표현

고정소수점 (Fixed-point) 표현

- 소수가 고정된 소수점을 통해서 구분하여 표현된 방식임
- 표현 범위의 한계가 있어 아주 큰 값과 매우 작은 값을 표현하는 것이 불가능함
- 예 : $(17.60)_{10}$

부동소수점 표현

- B는 기수(Base), E는 지수(Exponent)를 나타냄
- 지수를 사용해 소수점의 위치를 이동하여 수의 표현 범위를 확대함
- 예 : $(176,000)_{10} = 1.76 \times 10^5$,
 $(0.000176)_{10} = 1.76 \times 10^{-4}$



05

문자 데이터의 표현

- 1) 문자 표현 방법
- 2) 패리티 검사 코드
- 3) 해밍 코드(Hamming code)

1) 문자 표현 방법

■ 문자의 표현 방법

영숫자 코드
(Alphanumeric
Code)

표준 BCD 코드
(Binary Coded
Decimal)

ASCII 코드

- 컴퓨터에 사용되는
영문자와 숫자, 특수문자의 데이터를
0과 1의 조합으로 구성된 코드로 표현

1) 문자 표현 방법

■ 문자의 표현 방법

영숫자 코드
(Alphanumeric
Code)

표준 BCD 코드
(Binary Coded
Decimal)

ASCII 코드

- IBM, 6bit ($2^6 = 64$ 개 자료 표현)
- 이진화 십진 코드
- 기본적으로 6비트의 길이를 갖는 코드지만
좀 더 효율적으로 사용하기 위해서
존(zone)비트와 숫자(digit)비트로 분리하고
이를 조합해서 코드 생성



1) 문자 표현 방법

■ 문자의 표현 방법

영숫자 코드
(Alphanumeric
Code)

표준 BCD 코드
(Binary Coded
Decimal)

ASCII 코드

- 미국 국립 표준협회
- 7비트와 패리티 비트가 추가된 두 종류의 8비트의 길이를 갖는 코드
- $128(2^7)$ 가지의 정보를 표현



1) 문자 표현 방법

▪ BCD 코드와 3초과 Code

BCD 코드	3초과 Code
<ul style="list-style-type: none">• 10진수 1자리를 2진수 4자리(Bit)로 표현하는 가중치 코드• 10진수 입/출력이 편함	<ul style="list-style-type: none">• 8421 코드 + (3)10• 비가중치 코드, 자기보수 코드

= 2진수 10진수, 8421 코드

비교

8421코드와 다른 점

- 0000~0010을 표현할 수 없음
- 1010~1100을 표현할 수 있음

1) 문자 표현 방법

▪ BCD 코드와 3초과 Code

10진수	8421	3초과
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

10진수	8421	3초과
10	0001 0000	0100 0011
11	0001 0001	0100 0100
12	0001 0010	0100 0101
13	0001 0011	0100 0110
23	0010 0011	0101 0110

1) 문자 표현 방법

▪ 2진수와 BCD 코드의 차이점

2진수	BCD
<ul style="list-style-type: none">• 각 자리마다 무게 값이 있음	<ul style="list-style-type: none">• 10진수 한자리를 4비트로 구성된 2진수로 표현함• 4개의 비트 사용시 0000 ~ 1111까지 16개 표현이 가능하나, 10개만 사용하고 1010 ~ 1111까지는 사용하지 않음

기초예제

다음 값을 2진수와 BCD 코드로 나타내 보세요.

$(256)_{10}$

2	256	
2	128	...0
2	64	...0
2	32	...0
2	16	...0
2	8	...0
2	4	...0
2	2	...0
	1	...0

2진수 ▶ $(100000000)_2$

BCD 코드 ▶ 0010/0101/0110



예제

36 + 42를 BCD로 바꿔서 연산해 보세요.

10진 연산 ►

$$\begin{array}{r} 36 \\ + 42 \\ \hline 78 \end{array}$$

BCD 연산 ►

$$\begin{array}{r} 0011 \quad 0110 \\ + 0100 \quad 0010 \\ \hline 0111 \quad 1000 \end{array}$$



기초예제

다음 값을 BCD 코드로 나타내 보세요.

$(52.76)_{10}$

BCD 코드 ▶ 0101/0010/.0111/0110

5 2 . 7 6



예제

7 + 5를 BCD로 바꿔서 연산해 보세요.

10진 연산 ▶

$$\begin{array}{r} 7 \\ + 5 \\ \hline 12 \end{array}$$

BCD 연산 ▶

$$\begin{array}{r} 0111 \\ + 0101 \\ \hline 1100 \quad (\text{BCD 수가 아님}) \\ + 0110 \quad (\text{6을 더함}) \\ \hline 0001\ 0010 \quad (12_{10} \text{의 BCD 수}) \end{array}$$

↑
자리올림 발생

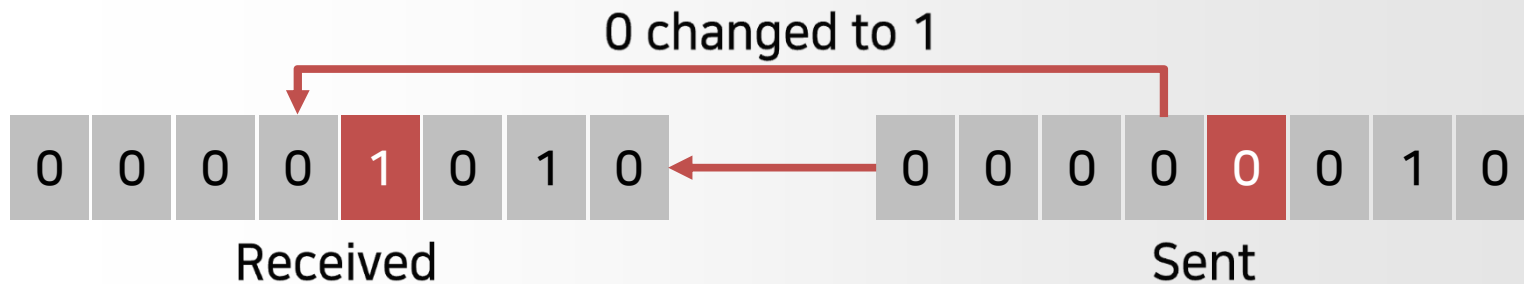


2) 패리티 검사 코드

■ 패리티 검사 코드

▶ 데이터는 전송 중에 변경될 수 있어 신뢰성 있는 통신을 위해 오류들은 검출·정정되어야 하는데, 오류 검출에서 가장 널리 사용하는 코드

▶ 단일 - 비트 오류



2) 패리티 검사 코드

■ 패리티 비트

- ▶ 중복 비트로 데이터 단위에 덧붙임
- ▶ 패리티 비트를 포함한 데이터 단위 내의 1의 전체 개수가 짝수(또는 홀수)가 되도록 함

짝수 패리티 (Even parity)

- 1의 전체 개수가 **짝수**가 되도록 함
- '0' 추가

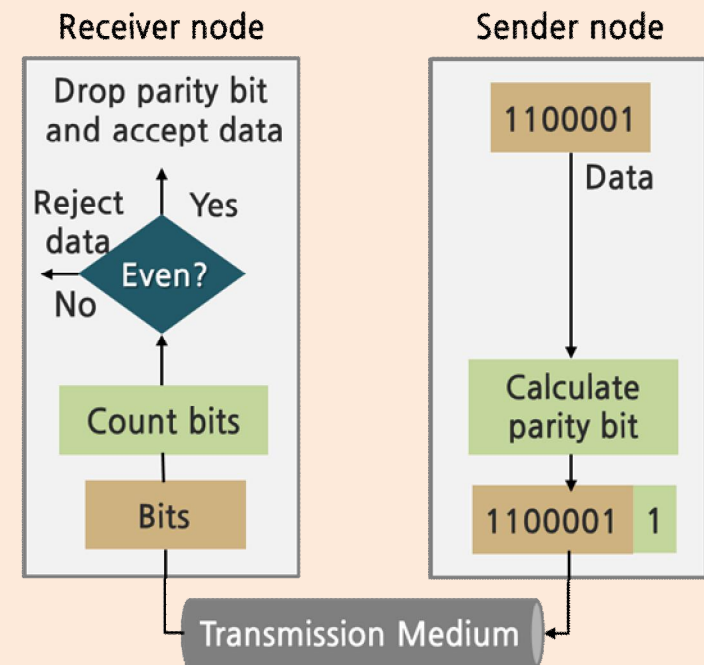
홀수 패리티 (Odd parity)

- 1의 전체 개수가 **홀수**가 되도록 함
- '1' 추가

예제

송신자가 'world'라는 단어를 보내고자 한다.
ASCII를 사용하면 다섯 글자는 다음처럼 코드화 된다.

	실제 전송되는 비트
1110111	11101110
1101111	11011110
1110010	11100100
1101100	11011000
1100100	11001001



3) 해밍 코드(Hamming code)

■ 해밍 코드

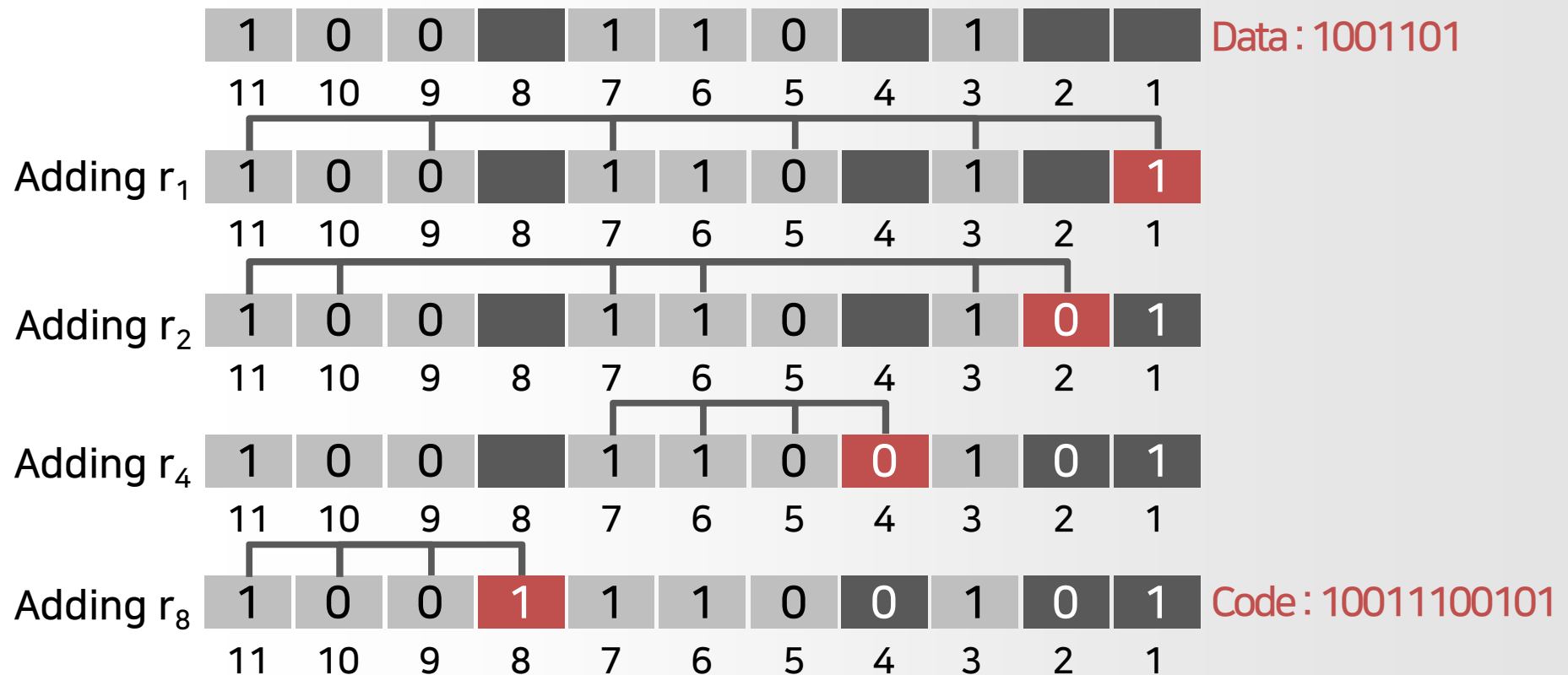
- ▶ 오류를 스스로 검출하여 교정이 가능한 코드
- ▶ 1Bit의 오류만 교정이 가능
- ▶ 데이터 비트 외에 여러 개의 교정을 위한 잉여 비트가 요구
- ▶ 1, 2, 4, 8, 16, ... , 2^n 번째 비트는 오류 검출을 위한 패리티 비트



- r_1 = bits 1, 3, 5, 7, 9, 11
- r_2 = bits 2, 3, 6, 7, 10, 11
- r_4 = bits 4, 5, 6, 7
- r_8 = bits 8, 9, 10, 11

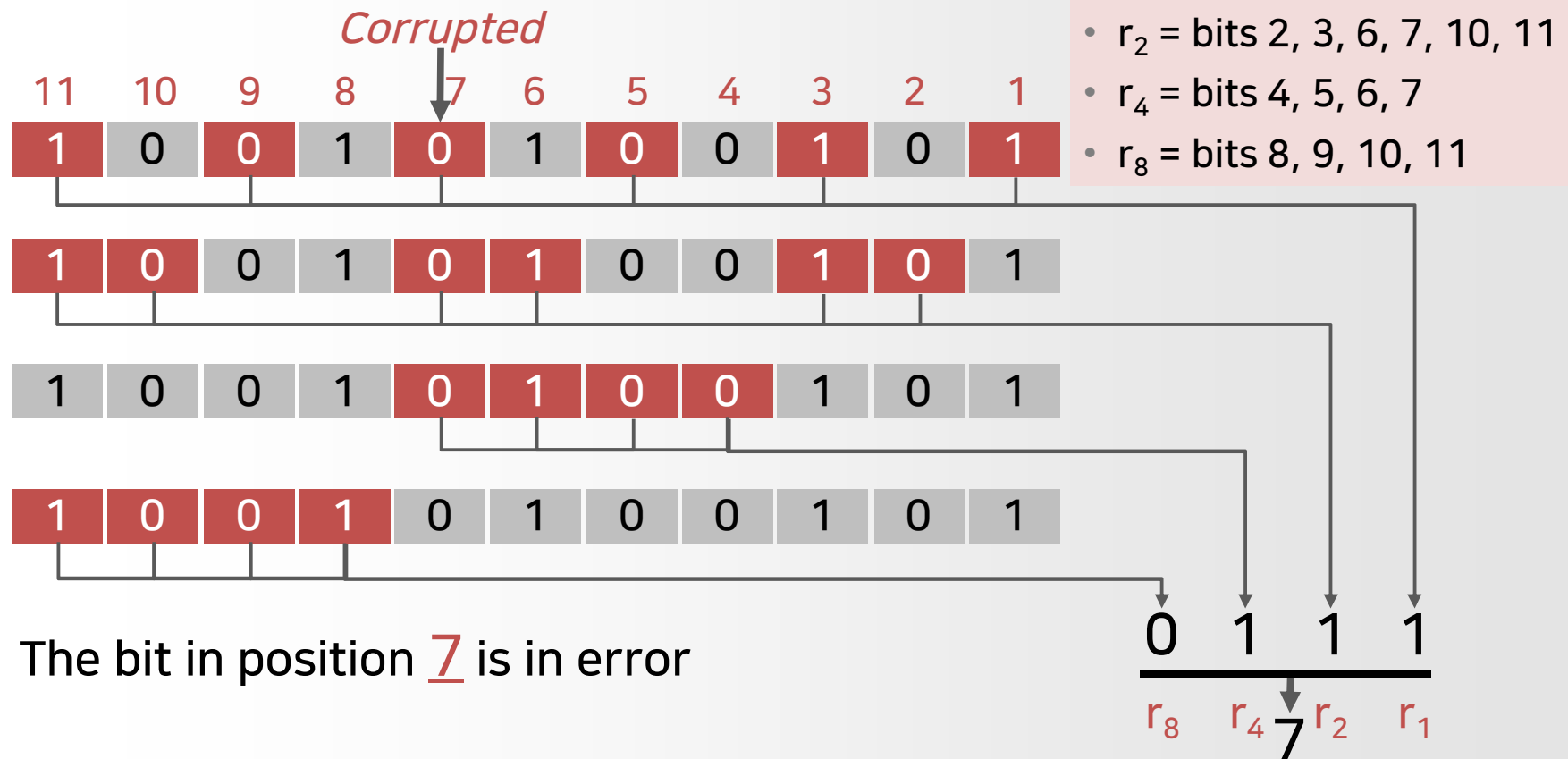
3) 해밍 코드(Hamming code)

▪ 해밍 코드



3) 해밍 코드(Hamming code)

■ 해밍 코드



예제

수신된 데이터가 짝수 패리티를 갖는 7비트 데이터 1000010인 경우 오류의 발생 위치를 찾아보세요.

- P_1 : 1, 3, 5, 7행에 대해서 짝수 패리티가 되도록 함 (첫번째 비트가 0이 되어야 짝수)
→ 1로 되어 있음 ► $C_1 = 1$
- P_2 : 2, 3, 6, 7행에 대해서 짝수 패리티가 되도록 함 (두번째 비트가 1이 되어야 짝수)
→ 0로 되어 있음 ► $C_2 = 1$
- P_3 : 4, 5, 6, 7행에 대해서 짝수 패리티가 되도록 함 (네번째 비트가 1이 되어야 짝수)
→ 0로 되어 있음 ► $C_3 = 1$

1	2	3	4	5	6	7bit
P_1	P_2	D_1	P_3	D_2	D_3	D_4
1	0	0	0	0	1	0

따라서

① $C_3C_2C_1 = 111$

② $C_3C_2C_1$ 값이 오류 발생의 위치로,

$C_3C_2C_1 = 000$ 일 때 오류가 발생하지 않은 것이므로, **오류가 발생**되었음을 알 수 있음

$C_3C_2C_1 = 111$ 이므로 7행에서 오류가 발생하였고, 실제 데이터는 1000011임



정리 하기

정보의 표현

- ✓ 컴퓨터 정보 :
2진수 비트들로 표현된 프로그램 코드와 데이터
- ✓ 프로그램 코드

	내용
기계어 (Machine language)	<ul style="list-style-type: none">• 기계 코드 (Machine code)• 컴퓨터 하드웨어 부품들이 이해할 수 있는 언어로서, 2진 비트들로 구성
어셈블리 언어 (Assembly language)	<ul style="list-style-type: none">• 고급 언어와 기계어 사이의 중간 언어• 어셈블러로 번역시, 기계어와 일대일 대응
고급 언어 (High-level language)	<ul style="list-style-type: none">• 영문자와 숫자로 구성되어 사람이 이해하기 쉬운 언어• C, C++, PASCAL, FORTRAN, COBOL 등• 컴파일러 (compiler) 이용해 기계어로 번역

정리 하기

진법의 변환

✓ 10진법 (Decimal notation)

- 0~9사이의 10개의 숫자를 사용하여 정보를 표현
- 10진수 431의 표현 :
 $431 = 4 \times 10^2 + 3 \times 10^1 + 1 \times 10^0$

✓ 2진법 (Binary notation)

- 0, 1만 사용하여 정보를 표현
- 2진수 $(0111)_2$:
 $(0111)_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

정리 하기

보수

- ✓ 컴퓨터가 기본적으로 수행하는 덧셈 회로를 이용하여 뺄셈을 수행하기 위해 사용
- ✓ r 의 보수
 - 10진법에는 10의 보수가 있고, 2진법에는 2의 보수가 있음
 - 보수를 구할 숫자의 자릿수 만큼 0을 채우고 가장 왼쪽에 1을 추가하여 기준을 만들

(예) 33의 10의 보수는?

$$33 + X = 100 \rightarrow X = 100 - 33 \rightarrow X = 67$$

(예) 10101의 2의 보수는?

$$10101 + X = 100000 \rightarrow X = 100000 - 10101 \rightarrow X = 01011$$

정리 하기

보수

✓ $r-1$ 의 보수

- 10진법에는 9의 보수가 있고, 2진법에는 1의 보수가 있음
- 10진수 N 에 대한 9의 보수는 주어진 숫자의 자릿수 만큼 9를 채워 기준을 만들

(예) 33의 9의 보수는?

$$33 + X = 99 \rightarrow X = 99 - 33 \rightarrow X = 66$$

- 2진수 N 에 대한 1의 보수는 주어진 숫자의 자릿수 만큼 1을 채워 기준을 만들

(예) 10101의 2의 보수는?

$$10101 + X = 11111 \rightarrow X = 11111 - 10101 \\ \rightarrow X = 01010$$

정리 하기

데이터의 2진수 표현

- ✓ 양의 정수, 음의 정수, 소수를 표현
- ✓ 2진수는 0, 1, 부호 및 소수점의 기호를 이용하여 수를 표현
- ✓ 부호가 존재하는 2진 정수의 표현
(1의 보수, 2의 보수)

문자 데이터의 표현

- ✓ BCD
- ✓ ASCII
- ✓ 패리티 검사
- ✓ 해밍 코드 등



- 다음 시간에 살펴 볼 내용 -

03주차
디지털 논리회로

수고하셨습니다.