

02

강

컴퓨터과학 개론

컴퓨터와 데이터 (2)

컴퓨터과학과 이관용 교수



KOREA NATIONAL OPEN UNIVERSITY



학습목차

1 데이터와정보

2 진법

3 정수 표현

4 실수 표현

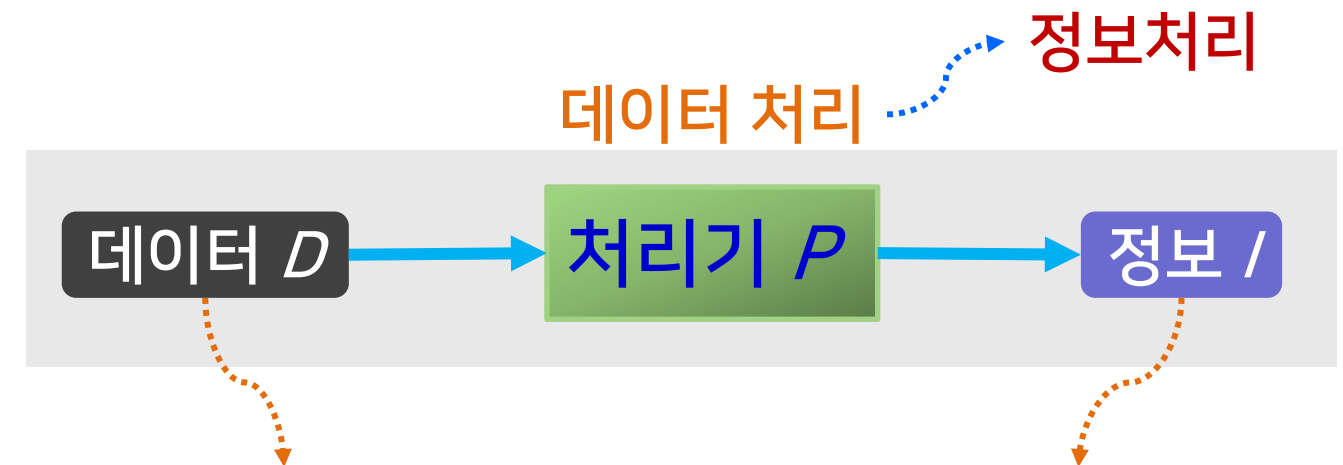
5 문자 표현

01

데이터와 정보

데이터와 정보의 관계

■ $I = P(D)$



현실 세계에서부터
관찰이나 측정을 통해
단순히 얻어지는 값/사실

어떤 상황에 대해
적절한 의사결정을
수행할 수 있게 하는 지식

데이터의 표현 형태

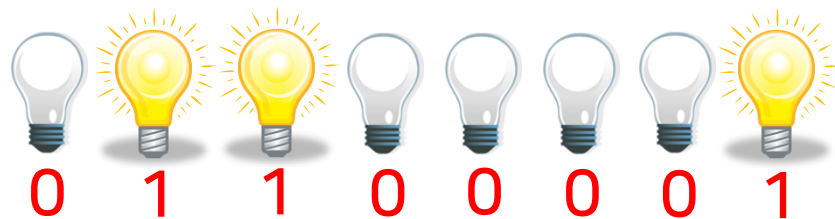
- 데이터의 유형과는 무관하게 일관된 표현 방식 사용
 - 문자, 정수, 실수, 이미지, 비디오, 오디오 등 → “비트 패턴”
 - 메모리에 저장된 데이터 유형에 맞는 해석과 처리가 필요
 - 입출력장치나 프로그램의 책임/역할

데이터의 표현 단위

■ 비트 binary digit



■ 바이트 byte



■ KB ($2^{10} \approx 10^3$), MB ($2^{20} \approx 10^6$), GB ($2^{30} \approx 10^9$), TB ($2^{40} \approx 10^{12}$), PB ($2^{50} \approx 10^{15}$), EB ($2^{60} \approx 10^{18}$), ZB ($2^{70} \approx 10^{21}$), YB ($2^{80} \approx 10^{24}$)

■ 워드 word

- 컴퓨터 연산의 기본 단위가 되는 정보의 양 → 보통 32비트, 64비트

02

진법

진법

■ 수를 세는 방법 또는 단위

- r 진법 $\rightarrow 0, 1, \dots, (r-1)$ 까지의 숫자만을 사용하는 진법 $\rightarrow r$ 진수

2진법	0, 1	1010 ₂ 1001 _b
8진법	0, 1, ..., 7	720 ₈ 257 _o
10진법	0, 1, 2, ..., 9	99 ₁₀ 123 _d
16진법	0, 1, ..., 9, A, B, C, D, E, F	2CF ₁₆ FF30 _h

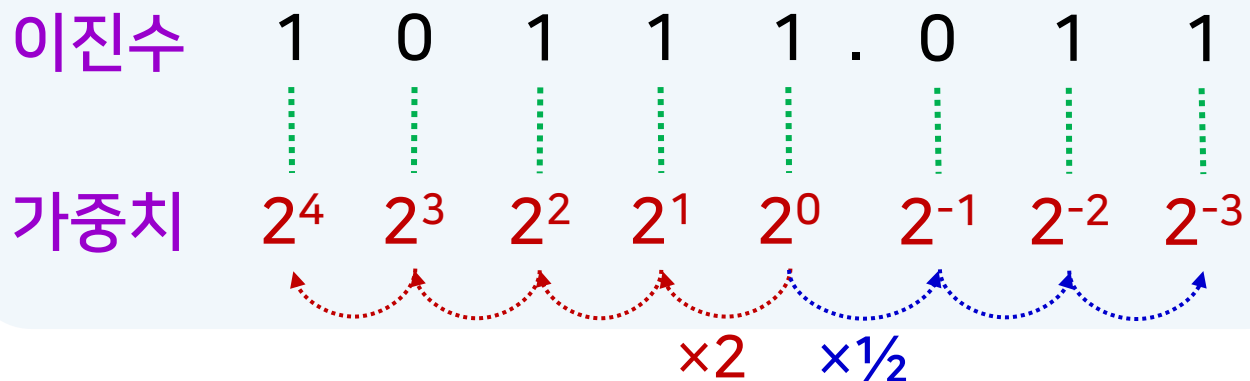
진법

진법의 각 숫자는 위치에 따라 서로 다른 가중치(자릿값)를 가짐

- r 진법의 자릿값 $\rightarrow r^x$ (x 는 숫자의 위치를 나타내는 정수)

123 (일이삼)

→ 일백이십삼 ($= 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$)



2진수를 10진수로 변환

■ 10진수 = \sum (각 비트값 × 해당 비트 위치의 가중치)

... 2^4 2^3 2^2 2^1 2^0 . 2^{-1} 2^{-2} 2^{-3} 2^{-4} ...
 ... 16 8 4 2 1 . $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$ $\frac{1}{16}$...

1 0 1 . 1 0 0 1

$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$

5.5625

8/16진수를 10진수로 변환

■ 10진수 = \sum (각 숫자값 \times 해당 위치의 가중치)

$\dots 8^4 \quad 8^3 \quad 8^2 \quad 8^1 \quad 8^0 \cdot 8^{-1} \quad 8^{-2} \quad 8^{-3} \quad 8^{-4} \dots$
 $\dots 16^4 \quad 16^3 \quad 16^2 \quad 16^1 \quad 16^0 \cdot 16^{-1} \quad 16^{-2} \quad 16^{-3} \quad 16^{-4} \dots$

$$3456_8 = 3 \times 8^3 + 4 \times 8^2 + 5 \times 8^1 + 6 \times 8^0$$

$$= 1838$$

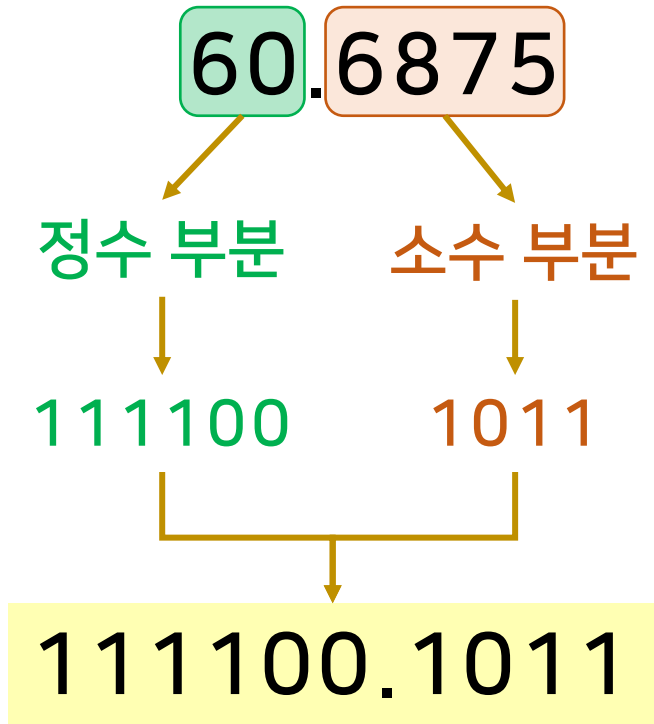
$$AE7_{16} = A \times 16^2 + E \times 16^1 + 7 \times 16^0$$

$$= 10 \times 16^2 + 14 \times 16^1 + 7 \times 16^0$$

$$= 2791$$

10진수를 r 진수로 변환 ($r = 2, 8, 16$)

- 정수 부분과 소수 부분을 구분하여 각각의 방법으로 처리한 후, 각 결과를 단순히 연결해서 나열



10진수_정수 부분 \rightarrow r 진수

```
입력값 = 10진수(정수 부분); i = 0;  
몫 = 입력값 / r; 나머지 = 입력값 mod r;  
결과(i) = 나머지;  
while (몫  $\neq$  0)  
    입력값 = 몫; i = i+1;  
    몫 = 입력값 / r;  
    나머지 = 입력값 mod r;  
    결과(i) = 나머지  
end  
출력[결과(i), 결과(i-1), ..., 결과(0)];
```

10진수_정수 부분 → r진수

2진수

$$\begin{array}{r}
 2 \overline{) 60} \\
 2 \overline{) 30} \cdots 0 \\
 2 \overline{) 15} \cdots 0 \\
 2 \overline{) 7} \cdots 1 \\
 2 \overline{) 3} \cdots 1 \\
 2 \overline{) 1} \cdots 1 \\
 0 \cdots 1
 \end{array}$$

$$60_{10} = 111100_2$$

8진수

$$\begin{array}{r}
 8 \overline{) 60} \\
 8 \overline{) 7} \cdots 4 \\
 0 \cdots 7
 \end{array}$$

$$60_{10} = 74_8$$

16진수

$$\begin{array}{r}
 16 \overline{) 60} \\
 16 \overline{) 3} \cdots C \\
 0 \cdots 3
 \end{array}$$

$$60_{10} = 3C_{16}$$

10진수_소수 부분 \rightarrow r진수

입력값 = 10진수(소수 부분);

i = 0;

while (입력값 \neq 0)

 임시변수 = 입력값 \times r;

 결과(i) = 임시변수의 정수 부분;

 i = i+1;

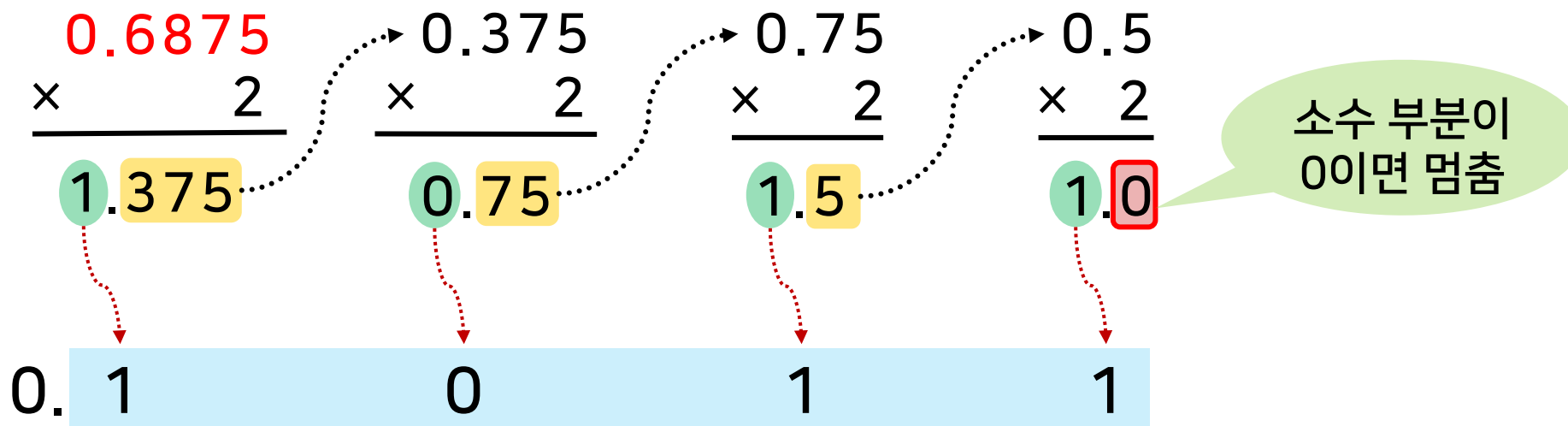
 입력값 = 임시변수의 소수 부분;

end

출력[0.결과(0), 결과(1), ..., 결과(i)];

10진수_소수 부분 → r진수

2진수



10진수_소수 부분 \rightarrow r진수

8진수

$$\begin{array}{r}
 0.6875 \\
 \times 8 \\
 \hline
 5.5
 \end{array}
 \quad
 \begin{array}{r}
 0.5 \\
 \times 8 \\
 \hline
 4.0
 \end{array}$$

0.5 (from 5.5) \rightarrow 0.5
 0.5 (from 4.0) \rightarrow 4

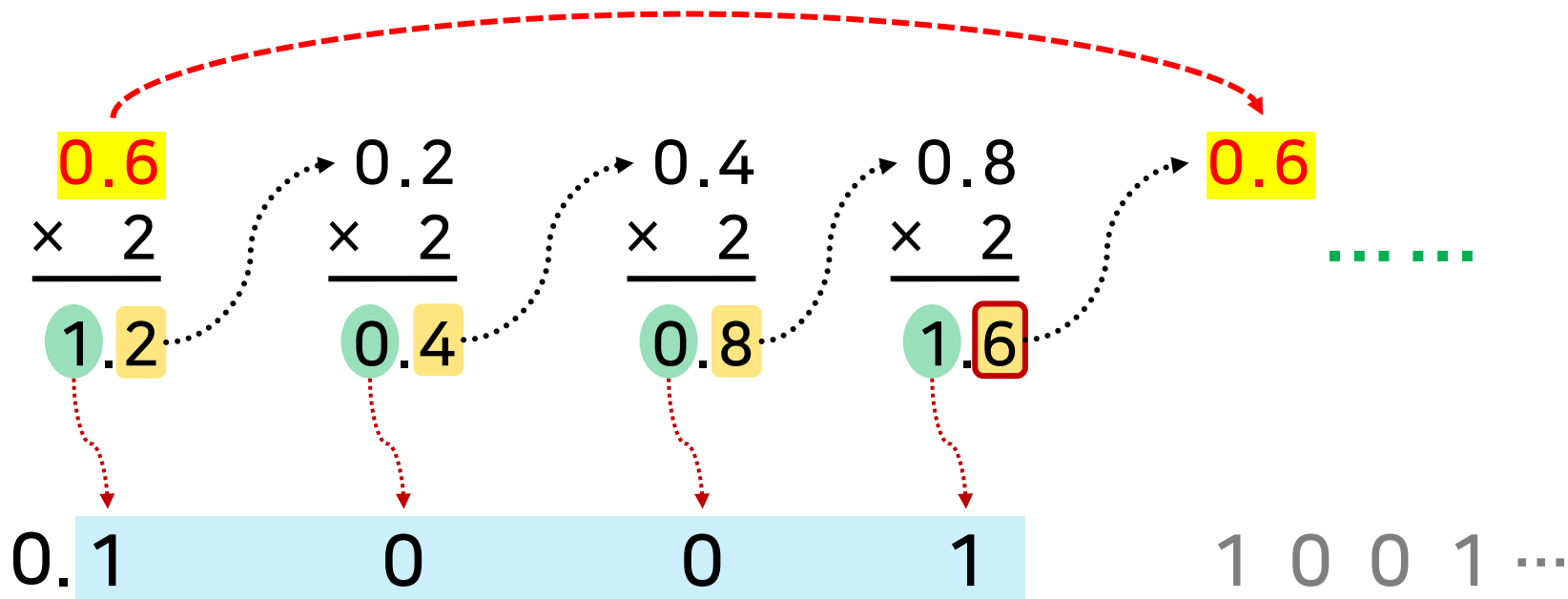
16진수

$$\begin{array}{r}
 0.6875 \\
 \times 16 \\
 \hline
 11.0
 \end{array}$$

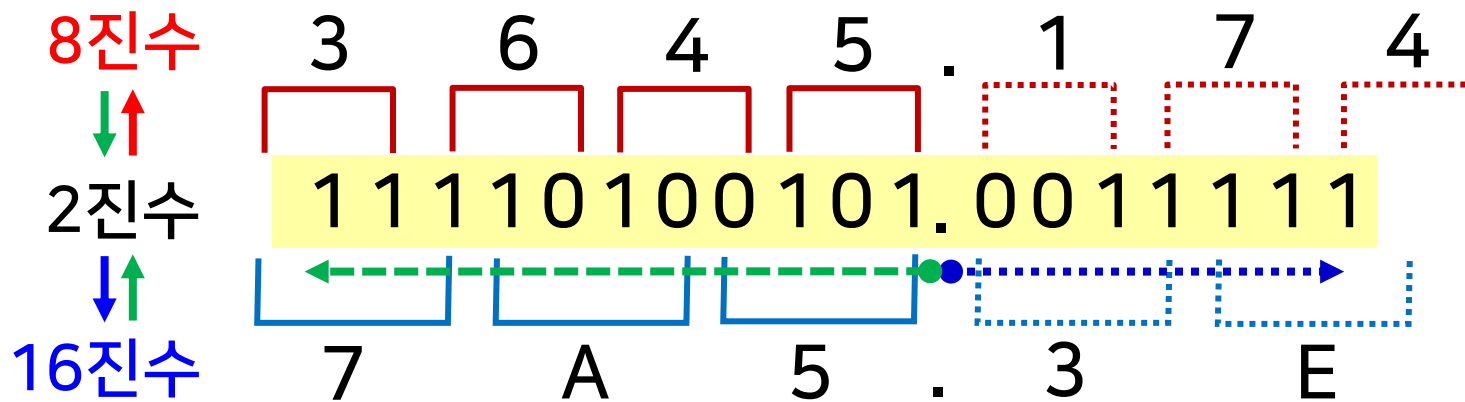
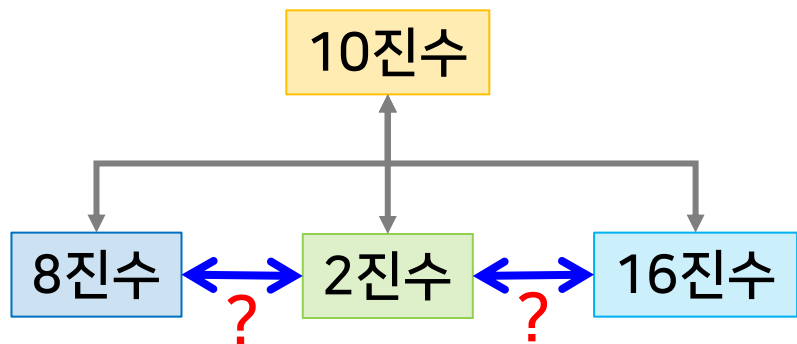
$0.B$ (from 11.0)

$$0.6875_{10} = 0.1011_2 = 0.54_8 = 0.B_{16}$$

10진수_소수 부분 \rightarrow r 진수



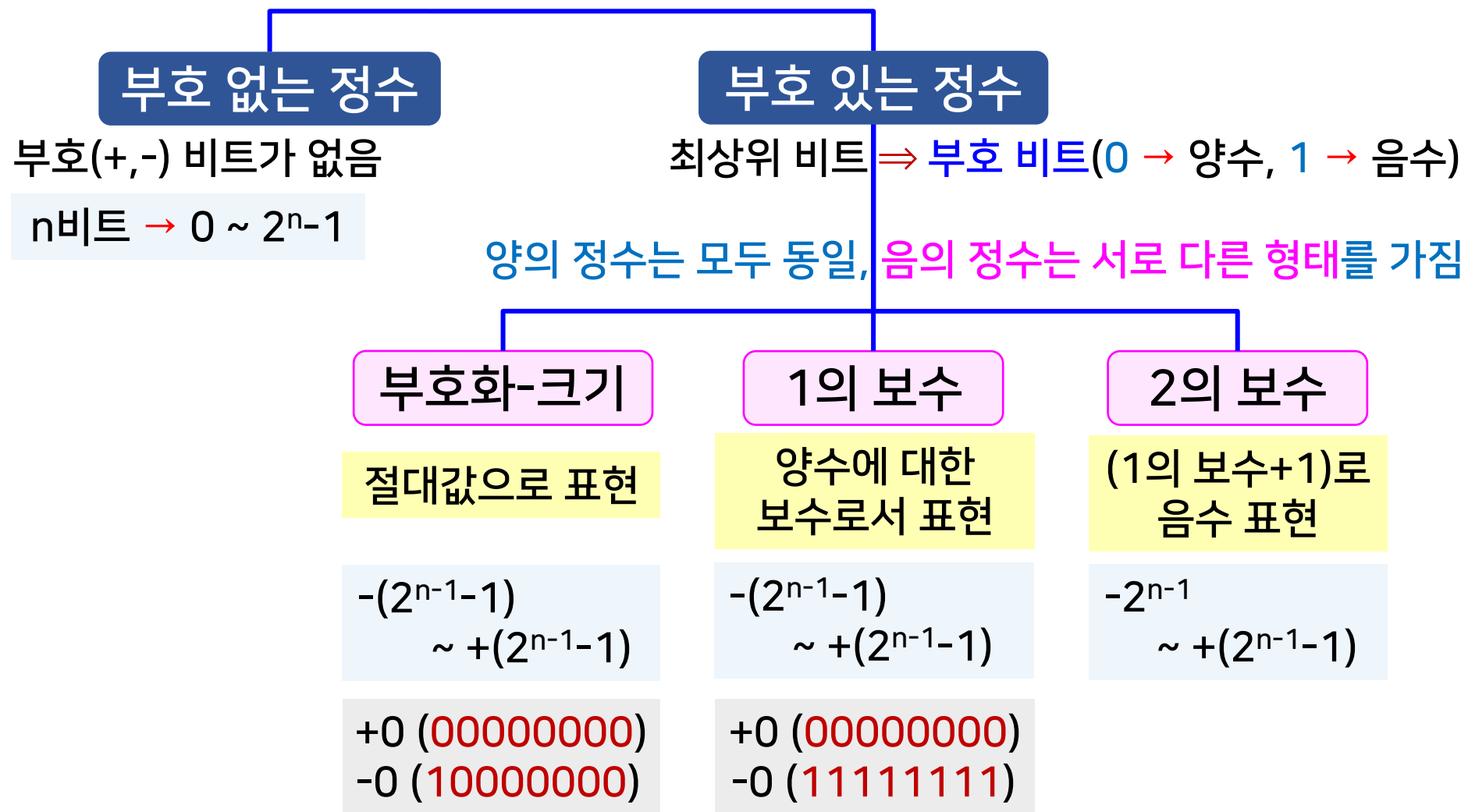
r 진수 간의 변환 ($r = 2, 8, 16$)



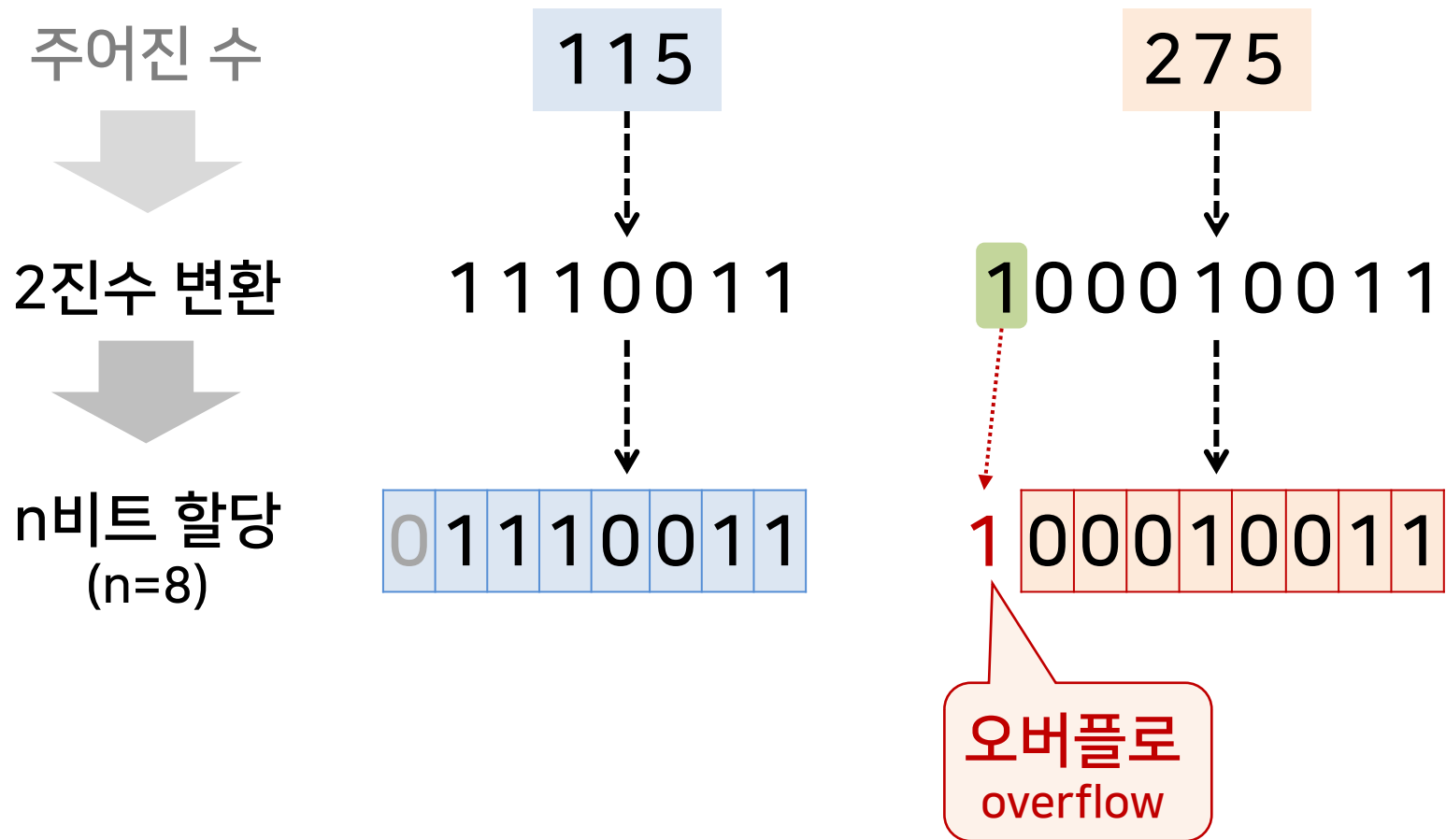
03

정수 표현

정수 표현 방법




부호 없는 정수

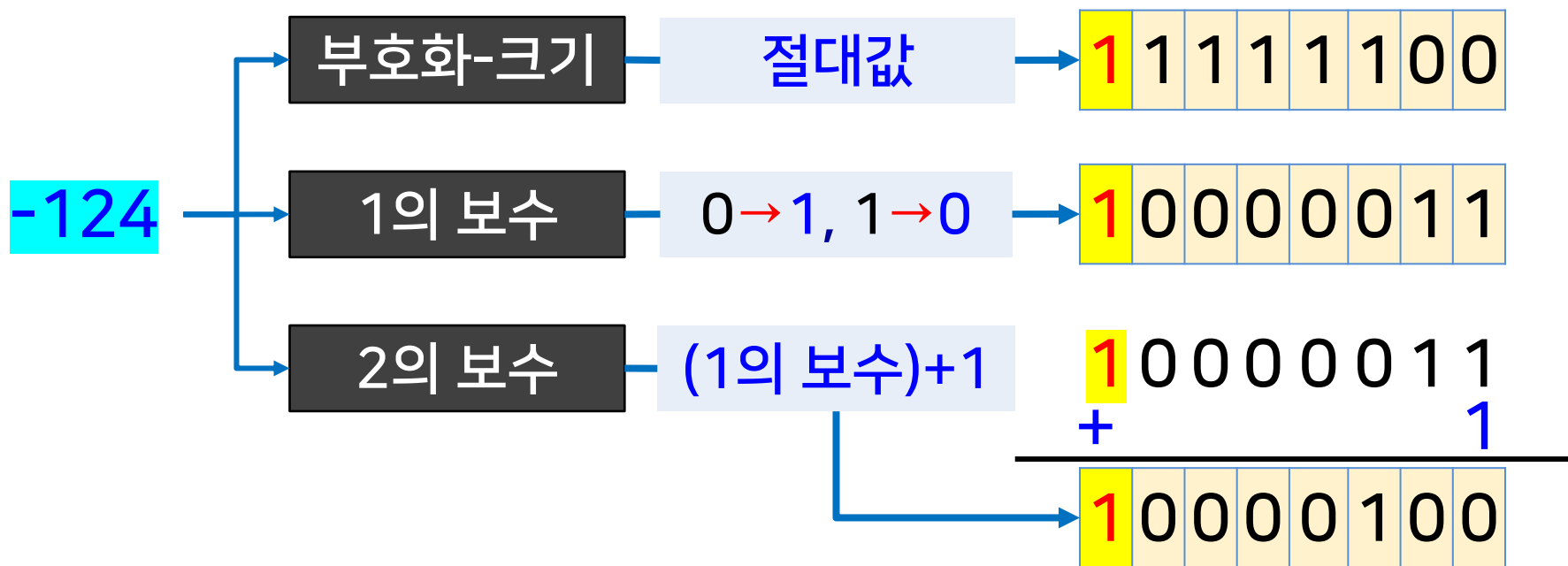


부호 있는 정수

■ $n = 8$ 비트인 경우

양수 → 0  124 

음수 → 1 



정수 표현 방법의 비교

이진표현	부호없는정수	부호있는정수		
		부호화-크기	1의 보수	2의 보수
00000000	0	+ 0	+ 0	+ 0
00000001	1	+ 1	+ 1	+ 1
00000010	2	+ 2	+ 2	+ 2
00000011	3	+ 3	+ 3	+ 3
...
01111100	124	+ 124	+ 124	+ 124
01111101	125	+ 125	+ 125	+ 125
01111110	126	+ 126	+ 126	+ 126
01111111	127	+ 127	+ 127	+ 127
10000000	128	-0	-127	-128
10000001	129	-1	-126	-127
10000010	130	-2	-125	-126
10000011	131	-3	-124	-125
...
11111100	252	-124	-3	-4
11111101	253	-125	-2	-3
11111110	254	-126	-1	-2
11111111	255	-127	-0	-1

2의 보수 방식의 응용

■ 뺄셈: 24 - 17

B에 대한 2의 보수를
취한 후 덧셈 수행

A - B

A + (-B)

00011000	+24		00011000	+24
- 00010001	+17	2의 보수	+ 11101111	-17
<hr/>			<hr/>	
			100000111	+7
			무시	

2의 보수 방식의 응용

이진수 10001101은 십진수로 얼마인가?

- 8비트, 2의 보수 방식

① 10진수를 이진수로 변환 → ② 보수 계산 → ③ 보수 결과 + 1

?

10001101

11110011
↓
-115

10001100
↓
11110011

10001101
- 1

10001100

04

실수 표현

실수 표현

과학적 표기법을 활용한 부동소수점 방식으로 표현



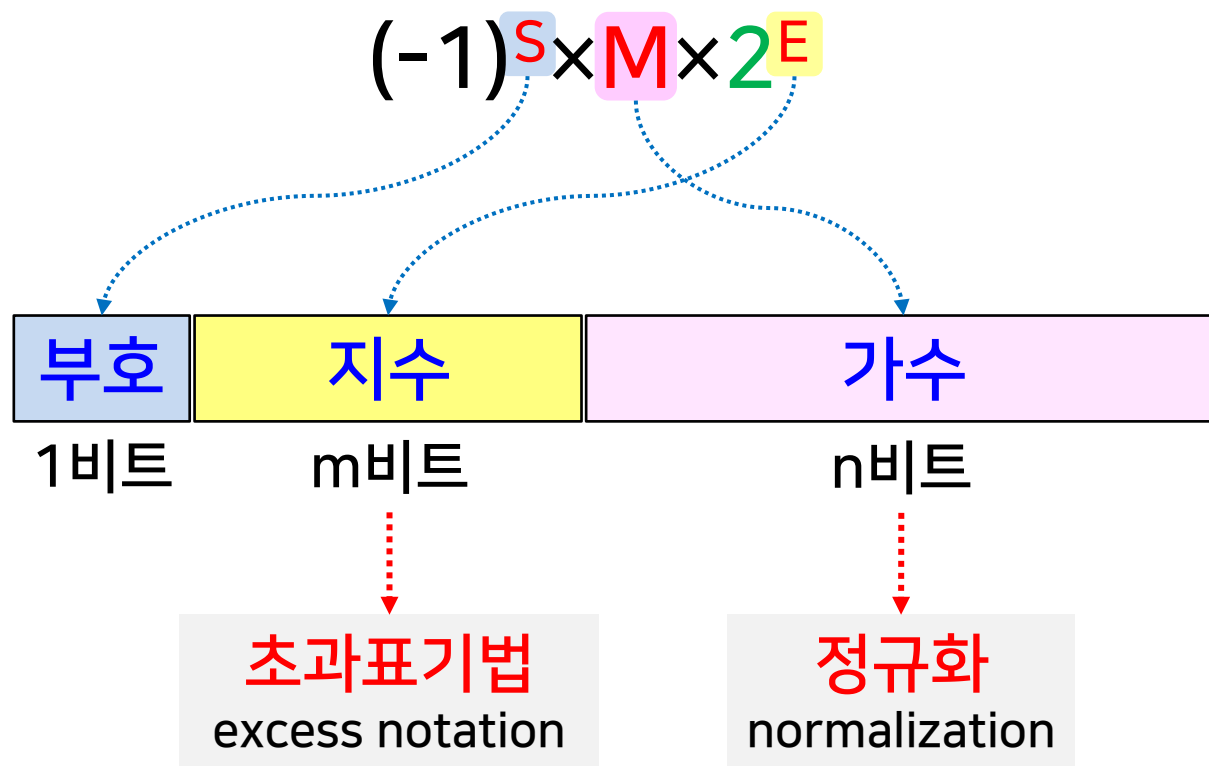
↓

$$(-1)^S \times M \times B^E$$

↓

부호 S	0	1
가수 M	1.234	5.678
기저 B	10	10
지수 E	12	-10

실수 표현

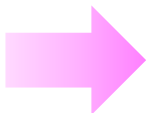


초과표기법

■ 부동소수점 방식의 **지수 부분**의 표현만을 위한 정수 표현 방법

- 매직 넘버 → 지수 부분이 m 비트로 구성 → 2^{m-1} 또는 $2^{m-1} - 1$

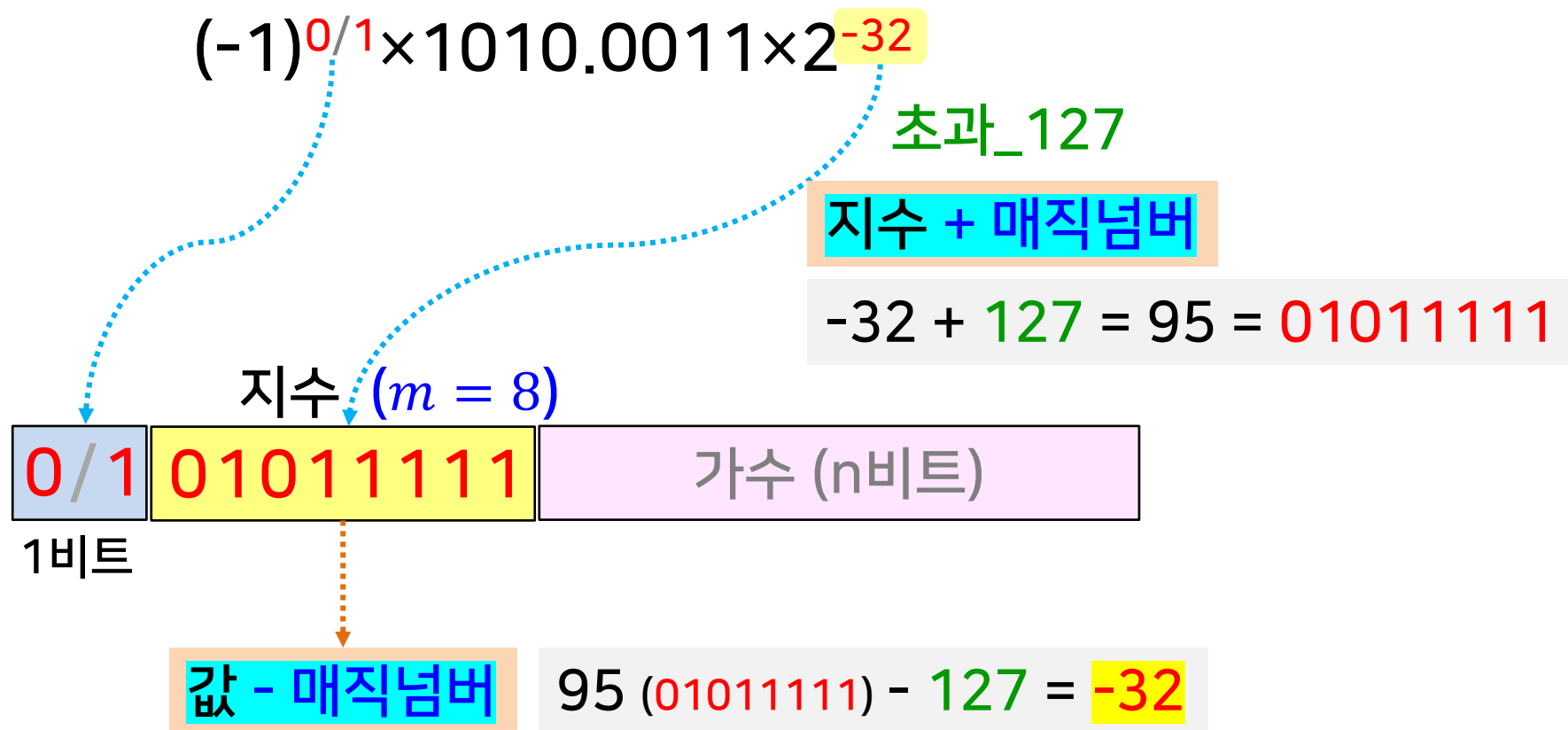
$$m = 8$$



$$2^{8-1} = 2^7 = 128 \text{ ('초과_128')}$$

$$2^{8-1}-1 = 2^7-1 = 127 \text{ ('초과_127')}$$

초과표기법



정규화

가수를 표현할 때 표준화된 형식이 필요

-1010.00110011 $\times 2^3$

-10.1000110011 $\times 2^5$

-101000110.011 $\times 2^{-2}$

-0.0101000110011 $\times 2^8$

정규화

-1.01000110011 $\times 2^6$

0.0000011011 $\times 2^0 \longrightarrow 1.1011 \times 2^{-6}$

10111010.0101 $\times 2^0 \longrightarrow 1.01110100101 \times 2^7$

실수 표현의 예

60.6875

(4바이트 $\rightarrow m = 8, n = 23$), 초과_127

111100.1011

정규화(\leftarrow , 5칸)

1.111001011 $\times 2^5$

초과표기법($5+127=132=10000100_2$)

0	10000100	111001011000000000000000
---	----------	--------------------------

8비트

23비트

IEEE 부동소수점 방식의 표준 형식

■ 단정도 single precision → 4바이트

초과_127



■ 배정도 double precision → 8바이트

초과_1023



05

문자 표현

문자 표현

- 키보드에서 입력되는 문자는 내부적으로 2진수로 표현되어 처리
- 각 문자마다 유일한 값으로써 코드를 할당할 수 있는
약속된 문자 체계가 필요
 - 종류 → ASCII, 유니코드, ...

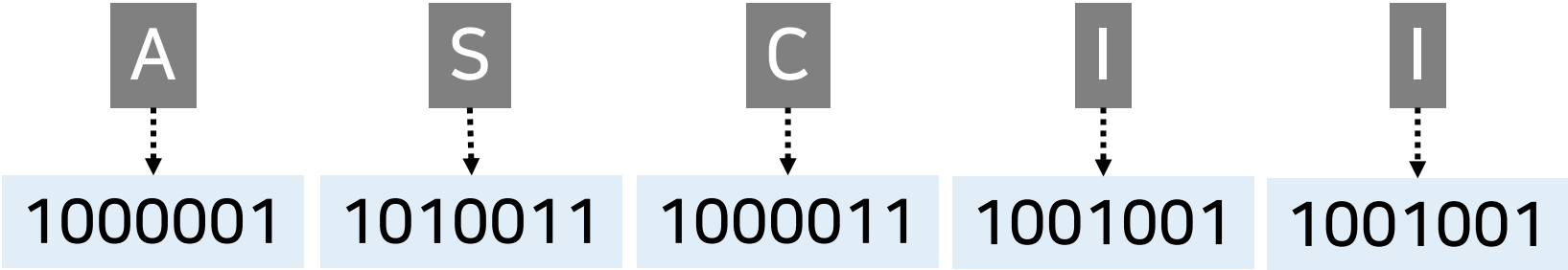
ASCII

■ American Standard Code for Information Interchange

- 미국표준협회(ANSI)
- 7비트 코드 → 128개(2^7)의 서로 다른 문자 표현
 - 확장된 아스키 Extended ASCII → 1비트 + 7비트



ASCII



48 0110000	0	49	1	57	9
65 1000001	A	66	B	90	Z
97 1100001	a	98	b	122	z

유니코드

- **세계의 모든 문자를 컴퓨터에서
일관되게 표현하고 다룰 수 있도록 설계된 표준**
 - 1990. 애플, IBM, MS 등 컨소시엄으로 설립한 유니코드가 첫 버전 발표
 - 1995년 국제 표준으로 제정 → 공식 명칭: ISO/IEC 10646
 - 사용 중인 플랫폼, 프로그램, 언어에 무관
 - 16비트 코드 체계 → 65,536개(2^{16})의 서로 다른 문자 표현

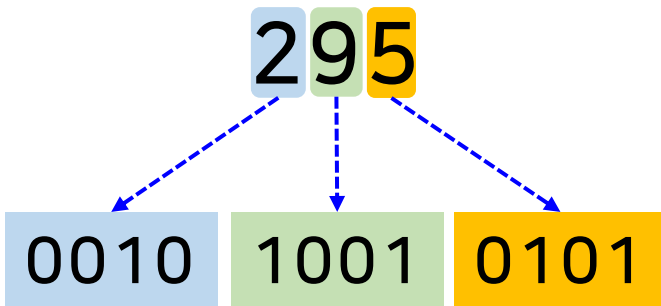
기타 코드 체계

■ EBCDIC Extended Binary Coded Decimal Interchange Code

- IBM 개발, 8비트 코드 → 실제 사용되는 문자 코드는 128개
 - IBM 메인프레임에서만 사용

■ BCD Binary Coded Decimal

- 4비트로 구성된 10개의 코드로 10진수를 표현하는 방식 → ‘8421 코드’





1. 데이터와 정보

- 정보 = P(데이터), 비트 패턴
- 데이터 표현 단위 → 비트, 바이트, KB, MB, GB, TB, PB, EB, ZB, YB, 워드

2. 진법

- 변환 → 2/8/16진수 ↔ 10진수, 8진수 ↔ 2진수 ↔ 16진수

3. 정수 표현

- 부호없는 정수
- 부호있는 정수 → 양수는 동일, 음수 표현(부호화 크기, 1의 보수, 2의 보수)

4. 실수 표현

- $(-1)^S \times M \times 2^E$, 초과표기법, 정규화

5. 문자 표현

- ASCII, 유니코드

03

강

다음시간 안내

자료구조 (1)

04강. 자료구조 (2)

05

강

알고리즘 (1)

