

컴퓨터학개론

2주차

최종석(jschoi@ssu.ac.kr)



목차

1. 수의 체계와 변환
2. 데이터의 표현
3. 논리회로

01

수의 체계와 변환

01. 수의 체계와 변환

I. 진법의 개념

- **진법** : 사용할 수 있는 숫자 개수와 각 숫자의 위치 값을 정의한 수 체계.
- 사람은 주로 10진법을, 컴퓨터는 2진법을 사용함.

하나 더 알기

진법을 알아야 하는 이유

- 컴퓨터는 0과 1, 디지털 형식으로 정보를 표현하는데, 이를 실생활에 적용하려면 아날로그 데이터를 디지털 데이터로 변환해야 함.
- 아날로그 데이터는 연속적 데이터, 디지털 데이터는 비연속적인 데이터를 의미.



그림 2-1 아날로그 시계와 디지털 시계

01. 수의 체계와 변환

II. 10진법과 2진법

- **10진법** : 0부터 9까지 10개의 숫자를 한 묶음으로 하여 10이 될 때마다 1자리씩 자리올림을 하는 방법.
- [그림 2-2]에서 자릿수의 의미
 - 4는 432를 100으로 나누었을 때의 '몫'
 - 32는 앞서 100으로 나누었을 때의 '나머지'
 - 2는 10으로 나누었을 때의 '나머지'

$$\begin{aligned} 432_{10} &= 4 \times 10^2 + 3 \times 10^1 + 2 \times 10^0 \\ &= 4 \times 100 + 3 \times 10 + 2 \times 1 \\ &= 432 \end{aligned}$$

그림 2-2 10진수 표현 방식

01. 수의 체계와 변환

II. 10진법과 2진법

- 컴퓨터는 0과 1, 두 가지 숫자로만 수를 표현하는 2진법을 사용.
- **컴퓨터가 2진법을 사용하는 이유** : 최초의 컴퓨터가 진공관을 사용했기 때문인데, 진공관은 켜고 끄는(on/off) 기능만 있었기 때문에 진공관이 꺼지면 0, 진공관이 켜지면 1로 인식.



그림 2-3 2진법 개념

01. 수의 체계와 변환

II. 10진법과 2진법

하나 더 알기

컴퓨터 용량 단위

■ 비트, 바이트, 워드

- **비트(Bit)** : 2진수를 표현하는 가장 기본 단위로, 데이터를 표현하는 최소 단위. 1비트는 0과 1, 즉 2가지 상태만 표현할 수 있음.
- **바이트(Byte)** : 문자를 표현하는 단위로, 1바이트는 8비트를 하나로 모은 것. 1바이트로는 $2^8(=256)$ 가지 상태를 표현할 수 있음.
- **워드(Word)** : 컴퓨터가 한번에 처리할 수 있는 데이터의 단위로, 1워드는 4바이트.



[비트, 디지털 세상을 열다\(05:25\)](#)

01. 수의 체계와 변환

II. 10진법과 2진법

하나 더 알기 컴퓨터 용량 단위

- 큰 용량을 표현하는 단위

표 2-1 큰 용량을 표현하는 단위 구분

용량 단위	약어	2진법 크기	바이트 대비 크기
바이트(Byte)	B	2^0	1B
킬로바이트(Kilo Byte)	KB	2^{10}	1,024B
메가바이트(Mega Byte)	MB	2^{20}	1,048,576B
기가바이트(Giga Byte)	GB	2^{30}	1,073,741,824B
테라바이트(Tera Byte)	TB	2^{40}	1,099,511,627,776B
페타바이트(Peta Byte)	PB	2^{50}	1,125,899,906,842,624B

01. 수의 체계와 변환

III. 진법 변환

▪ 10진수 → 2진수

- 정수 계산 방식 : 10진수를 계속 2로 나누면서 몫은 아래에, 나머지는 오른쪽에 기입하면 됨. 몫이 더 이상 2로 나누어지지 않을 때 아래에서부터 순서대로 나머지를 나열하면 2진수가 됨.

2)	41	(나머지)
2)	20	... 1
2)	10	... 0
2)	5	... 0
2)	2	... 1
	1	... 0

결과: 101001₂

그림 2-4 10진수 → 2진수 변환: 정수 부분

01. 수의 체계와 변환

III. 진법 변환

■ 10진수 → 2진수

- **소수 계산 방식** : 정수 부분은 앞서 말한 방식대로 변환. 소수점 아래의 소수 부분은 2를 계속 곱하면서 정수로 자리올림이 발생하는지 기록하고 이를 2진수 변환하면 됨.

$$\begin{array}{r} 0.6875 \\ \times 2 \\ \hline 1.3750 \text{ (} \rightarrow 0.1 \text{)} \\ 0.3750 \\ \times 2 \\ \hline 0.7500 \text{ (} \rightarrow 0.10 \text{)} \\ 0.7500 \\ \times 2 \\ \hline 1.5000 \text{ (} \rightarrow 0.101 \text{)} \\ 0.5000 \\ \times 2 \\ \hline 1.0000 \text{ (} \rightarrow 0.1011 \text{)} \end{array}$$

결과: 0.1011₂

그림 2-5 10진수 → 2진수 변환: 소수 부분

01. 수의 체계와 변환

III. 진법 변환

▪ 2진수 → 10진수

- 정수 계산 방식 : 2진수의 0과 1을 각 자릿수만큼의 2의 지수 승으로 곱한 후 모두 더하면 됨.

$$\begin{aligned} 101001_2 &= 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^0 \\ &= 1 \times 32 + 1 \times 8 + 1 \times 1 \\ &= 41_{10} \end{aligned}$$

그림 2-6 2진수 → 10진수 변환 : 정수 부분

01. 수의 체계와 변환

III. 진법 변환

▪ 2진수 → 10진수

- 소수 계산 방식 : 정수의 변환 방식과 동일하게 각 자릿수를 고려해 계산하면 됨. 단, 정수와는 반대로 소수점 아래로 내려갈수록 자릿수가 커지고 마이너스를 붙여 계산해야 함.

$$\begin{aligned} 0.1011_2 &= 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &= 1 \times 0.5 + 1 \times 0.125 + 1 \times 0.0625 \\ &= 0.6875_{10} \end{aligned}$$

그림 2-7 2진수 → 10진수 변환 : 소수 부분

01. 수의 체계와 변환

III. 진법 변환

하나 더 알기

16진법

- 사람 입장에서는 2진법이 정확히 얼마인지 단번에 파악하기가 어렵고 2진법을 표기하면 자릿수가 길어지기 때문에 편의상 16진법을 사용하기도 함.
- 16진법으로 표현하려면 16가지 기호가 필요함. A는 10진수의 10을 의미하고, F는 10진수의 15를 의미함.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

- 2진법 → 16진법 변환 : 2진수의 오른쪽에서부터 4비트씩 모아 계산하면 됨.
✓ **TIP.** 2진법 → 8진법 변환 : 2진수의 오른쪽에서부터 3비트씩 모아 계산.

02

데이터의 표현

02. 데이터의 표현

I. 정수의 표현

- 정수(Integer) : 셀 수 있는 수를 의미.
- 정수의 구성 : 음의 정수, 0, 양의 정수

■ 부호 없는 정수

- 부호 없는 정수(Undsigned Integer) : 부호를 생략한다는 의미로, 모든 숫자는 0 또는 양의 정수.
- 8비트로 부호 없는 정수를 표현하면 $0_{10} \sim 255_{10}$ 까지 나타낼 수 있음.
- N비트를 이용한 부호 없는 정수 표현은 0부터 $(2^n - 1)$ 까지 가능.

$0000\ 0000_2 \leftrightarrow 0_{10}$
 $1111\ 1111_2 \leftrightarrow 255_{10}$

그림 2-8 8비트의 2진수 값을 10진수로 변환

02. 데이터의 표현

I. 정수의 표현

▪ 부호 없는 정수

- 부호 없는 정수에서 2진법을 이용한 사칙연산 방법은 기본적으로 10진법에서의 연산 방법과 동일.
- **덧셈 연산** : 2진법의 가장 오른쪽 비트인 최소유효비트(LSB)부터 시작해 각 비트의 수를 더하고, 1+1로 자리올림이 발생하면 상위 자리로 1을 올리면 됨.

$$\begin{array}{r} 0000\ 1010_2 = 10_{10} \\ + 1000\ 1010_2 = 138_{10} \\ \hline 1001\ 0100_2 = 148_{10} \end{array}$$

(a) $00001010_2 + 10001010_2$

$$\begin{array}{r} 0110\ 1010_2 = 106_{10} \\ + 1011\ 0011_2 = 179_{10} \\ \hline 1\ 0001\ 1101_2 = 285_{10} \end{array}$$

(b) $01101010_2 + 10110011_2$

그림 2-9 2진수의 덧셈 연산

02. 데이터의 표현

I. 정수의 표현

하나 더 알기

오버플로우

- **오버플로우(overflow)** : 덧셈의 결과가 8비트로 표현할 수 있는 범위를 넘어선 상황을 의미함.
- 컴퓨터 내부에서는 논리적으로 정확한 결과를 냈다 하더라도 표현 가능한 범위를 벗어났기 때문에 실제 수행 결과는 옳은 값이 아님.
- 즉, 계산 결과를 정확히 표현할 수 없음.
- 실제로 컴퓨터 프로그래밍 시 아주 큰 수끼리의 연산에서 오버플로우가 발생함.
- 문제는 연산 결과에 오류가 나더라도 프로그램은 계속 실행된다는 점.
- 틀린 값을 정확히 체크하지 않으면 최종 결과에 영향이 미침.

02. 데이터의 표현

II. 실수의 표현

- 실수(Real Number) : 유리수와 무리수를 총칭하여 확장한 수로, 수직선 위에 나타낼 수 있는 모든 수를 의미함.
- 컴퓨터 내부에서 실수를 표현하는 방법 : 고정소수점 표현, 부동소수점 표현

■ 고정소수점 표현

- [그림 2-14]와 같이 16비트를 사용하는 경우, 앞의 8비트는 정수 부분을 표현하고 나머지 8비트는 소수 부분으로 할당됨.



$$5.34_{10} = 101.01010111_2$$

그림 2-14 고정소수점 표현법

02. 데이터의 표현

II. 실수의 표현

▪ 고정소수점 표현

- 고정소수점 표현(Fixed-point Representation) : 소수점의 위치를 고정시켜 표현한다는 의미.
- 고정소수점 표현은 숫자 표현이 간단하기 때문에 연산 속도가 빠르다는 장점이 있지만, 일반적으로 컴퓨터에서는 이런 방식을 사용하지 않음.

02. 데이터의 표현

II. 실수의 표현

▪ 부동소수점 표현

- **부동소수점 표현(Floating-point Representation)** : 소수점의 위치를 고정시키지 않고 가수와 지수를 사용해 실수를 표현한다는 의미.
- 부동소수점으로 표현할 때는 정수 부분에 '0 아닌 수를 하나'만 남기는 정규형으로 먼저 바꿔야 함.
- [그림 2-15]를 보면 2진수 비트 열에서 지수는 가수 앞에 위치하며, 부호는 가수의 부호를 나타냄.

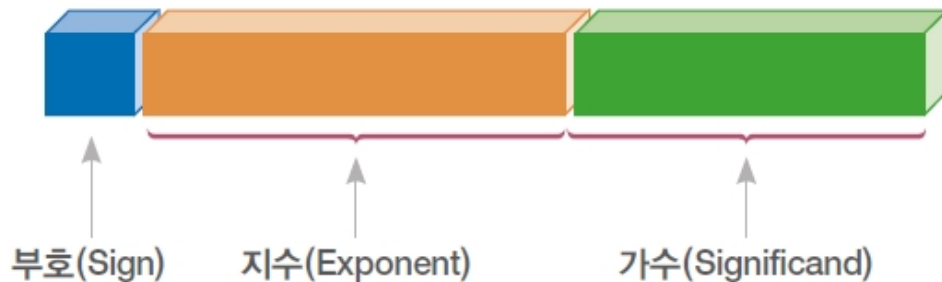


그림 2-15 부동소수점 표현 구조

02. 데이터의 표현

III. 문자의 표현

■ 아스키코드

- **아스키코드(ASCII Code)** : 초창기 컴퓨터로 문자를 표현하는 과정에서 여러 문제가 발생해 문자를 표현할 약속 체계를 만들었는데, 그중 가장 많이 사용하는 코드 체계가 미국표준협회에서 만든 아스키코드임.
- 아스키코드는 7비트로 구성되어 있으며, 표현할 수 있는 문자 개수는 $128(2^7)$ 임.

02. 데이터의 표현

III. 문자의 표현

■ 아스키코드

표 2-2 아스키코드

10 진수	16 진수	부호	10 진수	16 진수	부호	10 진수	16 진수	부호	10 진수	16 진수	부호
0	0	NUL	32	20	SP	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(72	48	H	104	68	h
9	9	HT	41	29)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p

02. 데이터의 표현

III. 문자의 표현

■ 아스키코드

10 진수	16 진수	부호	10 진수	16 진수	부호	10 진수	16 진수	부호	10 진수	16 진수	부호
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

02. 데이터의 표현

III. 문자의 표현

▪ 유니코드

- **유니코드(Unicode)** : 아스키코드는 영어 문화권에만 사용 가능해, 다양한 나라의 언어를 표현하고자 만든 코드 체계임.

	AC0	AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8	AC9	ACA	ACB	ACC	ACD	ACE	ACF
0	가 AC00	감 AC10	갸 AC20	갓 AC30	갈 AC40	각 AC50	갬 AC60	거 AC70	검 AC80	겐 AC90	갸 ACA0	결 ACB0	격 ACC0	겜 ACD0	고 ACE0	곰 ACF0
1	각 AC01	갑 AC11	갸 AC21	갹 AC31	갈 AC41	갹 AC51	겜 AC61	걱 AC71	겁 AC81	겜 AC91	겹 ACA1	결 ACB1	겹 ACC1	겜 ACD1	곡 ACE1	굽 ACF1
2	갹 AC02	갹 AC12	겹 AC22	갹 AC32	갹 AC42	겹 AC52	겜 AC62	긔 AC72	긔 AC82	겹 AC92	겹 ACA2	겹 ACB2	겹 ACC2	겹 ACD2	긔 ACE2	긔 ACF2

그림 2-17 한글 유니코드 일부