

# 자료의 표현과 연산

자료의 표현과 연산은 컴퓨터 과학의 핵심 개념으로, 컴퓨터가 데이터를 어떻게 저장하고 처리하는지를 이해하는 데 필수적이며, 자료의 표현, 진법 및 보수, 논리 회로와 논리 연산, 그리고 데이터 저장 방식에 대해서 알아보자.

## 자료의 표현

### 비트와 바이트

- 비트(Bit): 데이터의 최소 단위로, 0 또는 1의 값을 가집니다. 비트는 컴퓨터 내부에서 모든 정보를 표현하는 기본 요소입니다.
- 바이트(Byte): 8비트로 구성되며, 하나의 문자나 숫자를 표현하는 데 사용됩니다. 예를 들어, ASCII 코드에서 한 문자는 1바이트로 표현됩니다.

### 진법

컴퓨터는 데이터를 이진법(2진수)으로 표현합니다. 이진법은 0과 1의 두 가지 상태만을 사용하여 모든 데이터를 처리합니다. 전자 회로에서 전압의 ON/OFF 상태를 표현하기에 적합합니다. 진법의 주요 특징은 다음과 같습니다:

- 2진법(Binary): 0과 1로 구성되며, 컴퓨터 내부에서 모든 데이터는 이진수로 처리됩니다.  
예: 10진수 5는 2진수로 101101101.
- 10진법(Decimal): 인간이 일상적으로 사용하는 숫자 체계로, 0부터 9까지의 숫자를 사용합니다.
- 16진법(Hexadecimal): 0~9와 A~F(10~15)를 사용하며, 이진수를 간결하게 표현하기 위해 사용됩니다.  
예: 2진수 111111111111은 16진수로 FFF.

10진수	2진수	8진수	16진수
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

## 부동소수점 표현

실수를 표현하기 위해 부동소수점방식을 사용합니다. IEEE 754 표준에 따라 부호, 지수, 가수의 세 부분으로 나뉘어 표현됩니다. 예를 들어, 단정도(32비트) 부동소수점은 1비트의 부호 비트, 8비트의 지수부, 23비트의 가수부로 구성됩니다.

## 보수

보수는 음수를 표현하거나 뺄셈을 수행할 때 사용되는 개념입니다. 컴퓨터는 덧셈만을 수행할 수 있기 때문에, 뺄셈을 덧셈으로 변환하기 위해 보수를 사용합니다.

### 1의 보수와 2의 보수

- 1의 보수: 이진수의 모든 비트를 반전시켜 음수를 표현합니다. 예를 들어, 5의 1의 보수는 10진수로 -5를 표현하기 위해 5의 이진수(0101)를 반전하여 1010으로 나타냅니다.
- 2의 보수: 1의 보수에 1을 더하여 컴퓨터에서 음수를 표현하는 가장 일반적인 방법으로 음수를 표현합니다. 5의 2의 보수는 0101의 비트를 반전하여 1010을 만들고, 여기에 1을 더하여 1011로 표현합니다. 2의 보수는 +0과 -0의 중복 문제를 해결하고, 연산이 간단해지는 장점이 있습니다.
- 2의 보수 표현법

#### 정수 표현 범위

2의 보수를 사용하면  $n$  비트로 표현할 수 있는 정수의 범위는 다음과 같습니다:

최소값:  $-2^{(n-1)} - 2^{(n-1)}$

최대값:  $2^{(n-1)} - 1$

여기서  $n$ 은 비트 수를 의미합니다. 예를 들어, 8비트 시스템에서는 다음과 같은 범위를 가집니다:

최소값:  $-2^{(8-1)} = -128$

최대값:  $2^{(8-1)} - 1 = 127$

따라서 8비트의 2의 보수 표현법으로는 -128부터 127까지의 정수를 표현할 수 있습니다.

- 2의 보수 계산 방법

2의 보수를 구하는 방법은 다음과 같습니다:

1. 주어진 이진수의 모든 비트를 반전시킵니다 (0을 1로, 1을 0으로).
2. 반전된 값에 1을 더합니다.

예를 들어, 8비트로 표현된 숫자 5의 2의 보수를 구해보겠습니다.

5의 이진수 표현: 5는 이진수로 0000 1010입니다.

비트 반전: 1111 1010

1 더하기:

1111 1010

+ 1

---

결과 값 : 1111 1011

따라서 5의 2의 보수는 1111 1011이며, 이는 -5를 나타냅니다.

다시 하나의 예를 더 들어보자.

-7을 2의 보수로 표현해 보겠습니다.

1. 7의 이진수 표현: 7은 0000 0111입니다.

2. 비트 반전: 1111 1000

3. 1 더하기:

4. 1111 1000

+            1

-----

1111 1001

따라서 -7은 2의 보수로 1111 1001로 표현됩니다. 이와 같이 2의 보수 표현법은 음수를 간단하게 표현할 수 있는 방법을 제공합니다.

## 결론 :

2의 보수 표현법은 컴퓨터에서 음수를 표현하는 데 있어 매우 효율적이며, 정수의 범위를 넓히고 계산을 간소화하는데 기여합니다. 이 방법을 통해 컴퓨터는 덧셈과 뺄셈을 동일한 방식으로 처리할 수 있다.

# 논리 회로와 논리 연산

## 논리 회로

논리 회로는 논리 게이트를 사용하여 데이터를 처리하는 전자 회로입니다. 주요 논리 게이트는 다음과 같습니다:

- AND 게이트: 모든 입력이 1일 때만 출력이 1.
- OR 게이트: 입력 중 하나라도 1이면 출력이 1.
- NOT 게이트: 입력의 반대값을 출력.
- XOR 게이트: 입력이 서로 다를 때만 출력이 1.

이러한 논리 게이트는 조합 회로와 순차 회로를 구성하여 복잡한 연산을 수행합니다. 조합 회로는 입력에 따라 즉각적으로 출력이 결정되며, 순차 회로는 이전 상태를 기억하여 출력이 결정됩니다.

## 논리 연산

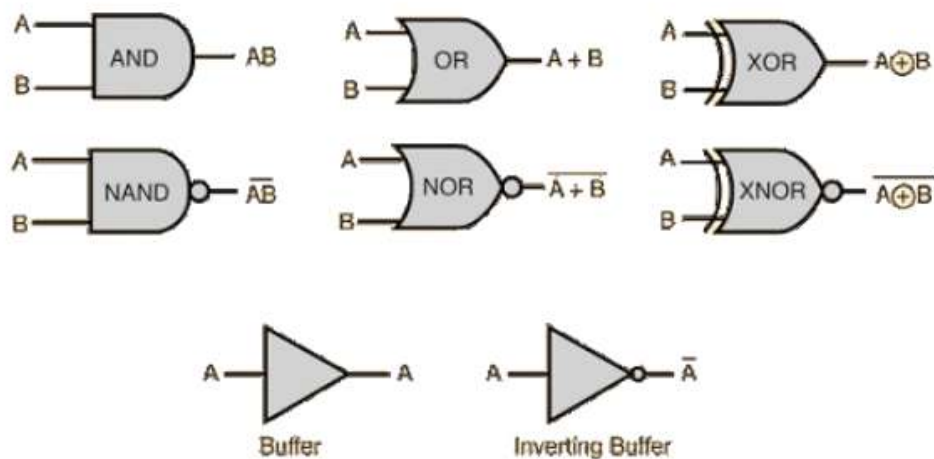
논리 연산은 불 대수에 기반하여 수행됩니다. 기본적인 논리 연산은 다음과 같습니다:

- AND 연산:  $A \cdot B$  (논리곱)
- OR 연산:  $A + B$  (논리합)
- NOT 연산:  $\overline{A}$  (부정)

이러한 연산은 컴퓨터의 산술 논리 장치(ALU)에서 수행되며, 데이터 처리의 기본적인 기능을 담당합니다.

게이트	기호	의미	진리표	논리식															
AND		입력신호가 모두 1일 때 1출력	<table><tr><th>A</th><th>B</th><th>Y</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	Y	0	0	0	0	1	0	1	0	0	1	1	1	$Y = A \cdot B$ $Y = AB$
A	B	Y																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OR		입력신호 중 1개만 1이어도 1출력	<table><tr><th>A</th><th>B</th><th>Y</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	Y	0	0	0	0	1	1	1	0	1	1	1	1	$Y = A + B$
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NOT		입력된 정보를 반대로 변환하여 출력	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0	$Y = A'$ $Y = \overline{A}$									
A	Y																		
0	1																		
1	0																		
BUFFER		입력된 정보를 그대로 출력	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	Y	0	0	1	1	$Y = A$									
A	Y																		
0	0																		
1	1																		
NAND		NOT + AND, 즉 AND의 부정	<table><tr><th>A</th><th>B</th><th>Y</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	Y	0	0	1	0	1	1	1	0	1	1	1	0	$Y = \overline{A \cdot B}$ $Y = \overline{AB}$
A	B	Y																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
NOR		NOT + OR, 즉 OR의 부정	<table><tr><th>A</th><th>B</th><th>Y</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	Y	0	0	1	0	1	0	1	0	0	1	1	0	$Y = \overline{A + B}$
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
XOR		입력신호가 모두 같으면 0, 한 개라도 틀리면 1출력	<table><tr><th>A</th><th>B</th><th>Y</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	Y	0	0	0	0	1	1	1	0	1	1	1	0	$Y = \underline{A} \oplus \underline{B}$ $Y = AB + \overline{A}\overline{B}$
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
XNOR		NOT + XOR, 즉 XOR의 부정	<table><tr><th>A</th><th>B</th><th>Y</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	Y	0	0	1	0	1	0	1	0	0	1	1	1	$Y = A \odot B$ $Y = \overline{A \oplus B}$ $Y = AB + \overline{A}\overline{B}$
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

## 논리회로



# 데이터 저장 방식

## 메모리와 저장 장치

컴퓨터에서 데이터 저장은 크게 두 가지로 나눌 수 있습니다: 휘발성 저장(메모리)과 비휘발성 저장(저장 장치)

### 휘발성 저장:

- RAM(Random Access Memory)이 대표적
- 전원이 꺼지면 데이터가 사라짐
- 빠른 읽기/쓰기 속도

### 비휘발성 저장:

- HDD(Hard Disk Drive), SSD(Solid State Drive) 등
- 전원이 꺼져도 데이터 유지
- 상대적으로 느린 속도지만 대용량 저장 가능

**메모리:** RAM(임의 접근 메모리)은 데이터를 임시로 저장하며, 전원이 꺼지면 데이터가 사라집니다. SRAM(정적 랜덤 액세스 메모리/Static Random Access Memory)과 DRAM(동적 랜덤 액세스 메모리/Dynamic Random Access Memory)이 있으며, SRAM은 빠르지만 비쌉니다. DRAM은 느리지만 용량이 큼니다. 다시 말해서 SDRAM은 주로 캐시 메모리와 같은 고속 데이터 접근이 필요한 곳에서 사용되고, DRAM은 일반적으로 컴퓨터의 주 메모리로 사용됩니다.

**저장장치:** 하드드라이브(HDD)나 SSD와 같은 영구 저장장치에 데이터를 저장합니다. 이들은 전원이 꺼져도 데이터를 유지합니다.

## 데이터 구조

데이터는 다양한 형태로 저장될 수 있습니다:

- 수치 데이터: 정수, 실수 등.
- 문자 데이터: ASCII, 유니코드 등으로 표현됩니다.
- 논리 데이터: 참과 거짓을 표현하는 데이터.
- 문자열 데이터: 여러 문자를 연속적으로 저장하는 형식입니다.

---

## QUIZE

문제 1: 컴퓨터에서 사용하는 기본적인 수 체계는 무엇인가요?

- ㄱ) 10진법
- ㄴ) 8진법
- ㄷ) 2진법
- ㄹ) 16진법

정답:ㄷ) 2진법

정답 이유:컴퓨터는 전자 회로를 기반으로 작동하며, 전자 회로는 두 가지 상태(켜짐과 꺼짐)를 표현할 수 있습니다. 이러한 두 가지 상태는 0과 1로 표현되며, 이를 2진법이라고 합니다. 2진법은 컴퓨터의 기본적인 데이터 표현 방식입니다.

문제 2: 다음 중 1바이트(Byte)의 비트 수는 얼마인가요?

- ㄱ) 4비트
- ㄴ) 8비트
- ㄷ) 16비트
- ㄹ) 32비트

정답:ㄴ) 8비트

정답 이유:1바이트는 8개의 비트로 구성되어 있습니다. 비트는 컴퓨터에서 정보를 표현하는 최소 단위이며, 1바이트는 256가지의 서로 다른 값을 표현할 수 있습니다. 이는 2의 8승( $2^8$ )으로 계산됩니다.

문제 3: 2진수에서 음수를 표현하기 위해 가장 일반적으로 사용되는 방법은 무엇인가요?

- ㄱ) 부호화-크기 표현
- ㄴ) 1의 보수 표현
- ㄷ) 2의 보수 표현
- ㄹ) 그레이 코드

정답:ㄷ) 2의 보수 표현

정답 이유:2의 보수 표현은 음수를 표현하는 데 가장 널리 사용되는 방법입니다. 이 방법은 양수의 이진수 표현을 반전한 후 1을 더하여 음수를 생성합니다. 이 방식은 덧셈과 뺄셈 연산을 단순화하고, 0의 유일한 표현을 보장하는 장점이 있습니다.

문제 4: 다음 중 AND 연산의 결과가 1이 되는 경우는 언제인가요?

- ㄱ) A=0, B=0
- ㄴ) A=0, B=1
- ㄷ) A=1, B=0
- ㄹ) A=1, B=1

정답:ㄹ) A=1, B=1

정답 이유:AND 연산은 두 입력이 모두 1일 때만 출력이 1이 됩니다. 따라서 A와 B가 모두 1일 때만 결과가 1이 되며, 나머지 경우는 모두 0이 됩니다.

문제 5: 다음 중 OR 연산의 진리표에서 출력이 0이 되는 경우는 무엇인가요?

ㄱ)  $A=0, B=0$

ㄴ)  $A=1, B=0$

ㄷ)  $A=0, B=1$

ㄹ)  $A=1, B=1$

정답: ㄱ)  $A=0, B=0$

정답 이유: OR 연산은 두 입력 중 하나라도 1이면 출력이 1이 됩니다. 따라서 A와 B가 모두 0일 때만 출력이 0이 되며, 나머지 경우는 모두 1이 됩니다.

문제 6: 다음 중 XOR 연산의 결과가 1이 되는 경우는 무엇인가요?

ㄱ)  $A=0, B=0$

ㄴ)  $A=0, B=1$

ㄷ)  $A=1, B=1$

ㄹ)  $A=1, B=0$

정답: ㄴ)  $A=0, B=1$  및 ㄹ)  $A=1, B=0$

정답 이유: XOR 연산은 두 입력이 서로 다를 때만 출력이 1이 됩니다. 즉, A와 B가 같을 때는 0, 다를 때는 1이 되므로,  $A=0, B=1$  또는  $A=1, B=0$ 일 때 결과가 1이 됩니다.