

00

CHAPTER

진법

1. 자료 구성 단위

- **컴퓨터 정보** : 2진수 비트들로 표현된 프로그램 코드와 데이터

비트(Bit, Binary Digit)	<ul style="list-style-type: none">• 자료(정보) 표현의 최소 단위• 2가지 상태(0과 1)를 표시하는 2진수 1자리
니블(Nibble)	<ul style="list-style-type: none">• 4개의 비트(Bit)가 모여 1개의 Nibble을 구성함• 4비트로 구성되며 16진수 1자리를 표현하기에 적합함
바이트(Byte)	<ul style="list-style-type: none">• 문자를 표현하는 최소 단위로, 8개의 비트(Bit)가 모여 1Byte를 구성함• 1Byte는 $256(2^8)$가지의 정보를 표현할 수 있음• 주소 지정의 단위로 사용됨
워드(Word)	<ul style="list-style-type: none">• CPU가 한 번에 처리할 수 있는 명령 단위• 반워드(Half Word) : 2Byte• 풀워드(Full Word) : 4Byte• 더블워드(Double Word) : 8Byte

필드(Field) - 레코드(Record) - 블록(Blocks) - 파일 - DB

1 Bit

☐

1 Byte

☐☐☐☐☐☐☐☐

bit 수	1	2	3	4
가능한 2진수 숫자 패턴	0	00	000	0000 0001
	1	01	001	0010 0011
		10	010	0100 0101
		11	011	0110 0111
			100	1000 1001
			101	1010 1011
			110	1100 1101
			111	1110 1111
표현 가능한 숫자 개수	2개	4개	8개	16개
	2^1 개	2^2 개	2^3 개	2^4 개

숫자표현 단위

단위	Byte	KB	MB	GB	TB	PB	EB
의미	byte	Kilo byte	Mega byte	Giga byte	Tera byte	Peta byte	Exa byte
용량	8 bit	1024 byte	1024 Kbyte	1024 Mbyte	1024 Gbyte	1024 Tbyte	1024 Pbyte
	2진수	2^{10}	2^{20}	2^{30}	2^{40}	2^{50}	2^{60}
	10진수	10^3	10^6	10^9	10^{12}	10^{15}	10^{18}

2. 진법변환

- 컴퓨터는 2진 시스템
- 10진수의 변환
 - 정수부분 : 해당 진수로 나누어서 나머지를 역순표현

〈정수 부분〉

2진수
$\begin{array}{r} 2 \overline{) 47} \\ 2 \overline{) 23} \dots 1 \\ 2 \overline{) 11} \dots 1 \\ 2 \overline{) 5} \dots 1 \\ 2 \overline{) 2} \dots 1 \\ 1 \dots 0 \end{array}$
$(47)_{10} = (101111)_2$

- 소수부분 : 해당 진수를 곱해서
결과의 정수부분을 차례대로 표
현(0 또는 반복되는 수가 나올
때까지 진행)

〈소수 부분〉

2진수		
$\begin{array}{r} 0.625 \\ \times 2 \\ \hline 1.250 \end{array}$	$\begin{array}{r} 0.250 \\ \times 2 \\ \hline 0.5 \end{array}$	$\begin{array}{r} 0.5 \\ \times 2 \\ \hline 1.0 \end{array}$
$(0.625)_{10} = (0.101)_2$		

10진수 2진수
변환

$$(47.625)_{10} \rightarrow (101111.101)_2$$

- 2^K 값
 - 2의 배수 값을 이용
 - 1, 2, 4, 8, 16, 32, 64....
- 컴퓨터의 수는 2진수 – 사람이 다루기 어려움
 - 복잡한 2진수 표현을 간단하게 8진수, 16진수로 표현하여 사용
- 8진수($2^3 = 8$) : 2진수 세 자리가 8진수 한 자리에 해당
- 16진수($2^4=16$) : 2진수 네 자리가 16진수 한 자리에 해당
 - 16진수 기호 : 0~9, A(10), B(11), C(12), D(13), E(14), F(15)

01

CHAPTER

자료의 표현

1. 정수의 표현

- 부호 없는 수(unsigned)
 - 0을 포함하여 양의정수만 이진수 변환 그대로 표현
- 부호 있는 수(signed)
 - 처음 1비트를 부호비트로 사용(0 양수, 1 음수)
 - 부호화 크기(signed magnitude) = 부호화 절대치(절대값)
 - 부호화 1의 보수(1's complement)
 - 부호화 2의 보수(2's complement)

2. 실수의 표현

- **부동소수점 표현(floating-point)**

- 소수점의 위치를 변경하여 지수를 조정하는 표현법
- 제한된 비트로 큰 숫자 또는 정밀한 숫자를 표현하기 위해 사용
(즉, 유효한 숫자의 범위를 늘리기 위해)
- 연산절차가 복잡하고 많은 시간이 걸리며 하드웨어적으로 복잡(컴퓨터에 부담이 되는 표현법임)

> 10진 부동소수점 수

$$976,000,000,000 = 9.76 \times 10^{11}$$

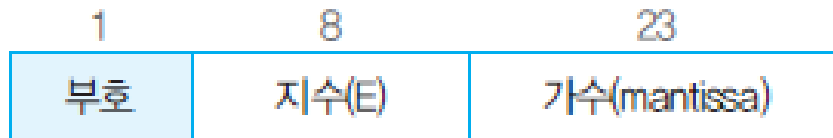
부호: __, 가수: __ 기수: __ 지수: __

$$-0.000000000000976 = -9.76 \times 10^{-12}$$

부호: __, 가수: __ 기수: __ 지수: __

- 2진 부동소수점수(IEEE 754 형식)

- 기수는 2 (표현 생략)
- 단정도 형식(32bits)
- 배정도 형식(64bits)



(a) 단정도 형식



(b) 배정도 형식

- 정규화

- 같은 수에 대한 부동소수점 표현이 여러가지 존재
- 여러 가지 표현법 중 표준 필요
- 정규화 과정 진행(가수와 지수부에 모두 정규화 과정 필요)

- 가수의 정규화

- 가수를 $1.bbb...b$ 형식으로 표현할 수 있도록 지수를 조정
- 가수를 표현할 때 소수의 $bbb...b$ 만 표현(1생략 : 표현범위 넓어짐)
- 숫자 0의 경우 모두 0

- 지수의 정규화

- 실수의 연산시 지수에 음수가 있는 경우 표현 및 연산 복잡
- 지수를 부호없는 수로 표현
- 바이어스 지수 (biased exponent)
- 바이어스수를 실제 지수에 더하여 계산

3. 기타자료의표현

❖ 문자표현 코드

- 산술연산에서 사용되는 수를 제외하고 컴퓨터에서 문자, 숫자, 특수문자들을 표현하기 위한 약속 코드들

• BCD(Binary Coded Decimal) 코드

- 2진화 10진코드
- 초창기 컴퓨터를 위해 IBM 개발
- 영어대문자 26개와 숫자 10개(0~9), 기타 특수문자 표현
 - 영어소문자 표현 못함
- 6Bit 사용(총 64개까지 구분가능)
- 2비트의 ZONE과 4비트의 Digit 비트로 구성
 - Zone은 영문인지 숫자인지 등 구분.
 - Digit는 몇 번째인지 구분.

- BCD의 표현(표준안이전. 약간씩 다름)

- 영문자 A : zone 11 digit 0001

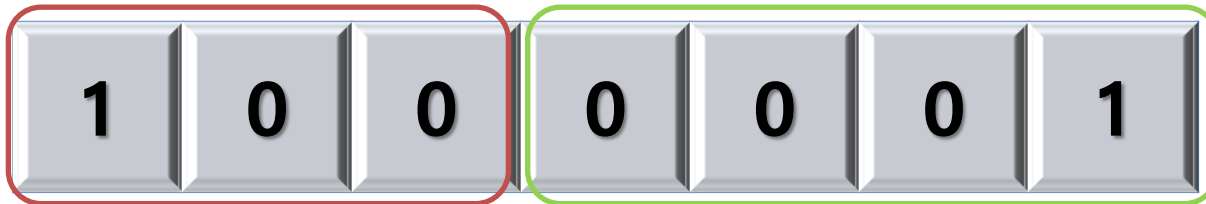


- 숫자의 zone은 00, 이를 생략하고 4bit로 2진 표현
- 10진수의 간편한 입출력에 사용
- 8421코드라 함(8421 - 2진 자리수값)
- 각 자리가 일정한 값을 표현. 가중치코드(weighted code)
- 십진수 9의 BCD코드



- ASCII(American Standard Code for Information Interchange) 코드

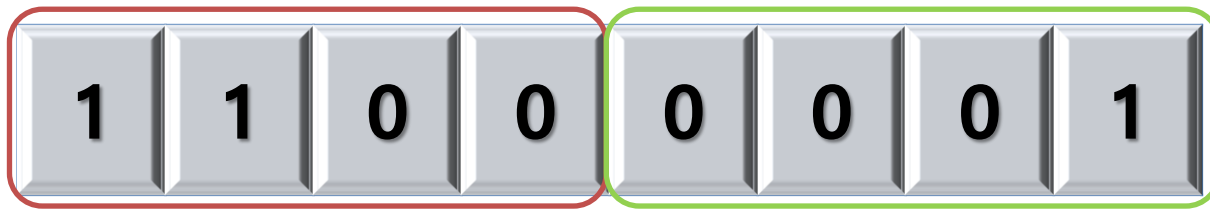
- 미국표준협회에서 개발한 7bit 코드
- 데이터통신 및 마이크로 컴퓨터용
- 영어대소문자, 숫자, 특수문자, 제어문자 등을 구분
- 3비트의 ZONE과 4비트의 Digit 비트로 구성
- 영문자 A : zone 100 digit 0001



- 실제 사용할 때는 패리티비트를 추가하여 8비트로 사용

- EBCDIC(Extended BCD Interchange Code) 코드

- 확장 2진화 10진 코드
- 대형컴퓨터 시스템을 위해 8Bit로 IBM에서 개발
- 4비트의 ZONE과 4비트의 Digit 비트로 구성
- 영문자 A : zone 1100 digit 0001



- 패리티비트를 추가하여 9비트로 사용

- UniCode 유니코드

- 전세계의 문자를 표현하기 위한 통합코드
- 2바이트 , 65536개의 문자 표현(한글 11172자)

❖ 특수용도 코드

• 3초과 코드(Excess-3 Code, xs-3 Code)

- BCD 코드에 3을 추가하여 나타냄
- 10진숫자의 연산을 수행하기 위해 사용
- 6의 3초과코드 = 6의 BCD코드 0110 + 0011(3) = 1001



- 비가중치 코드(non-weighted code) : 자리값 의미 없음
- 자보수코드(self-complement code) : 그 안에 보수값을 가지고 있음

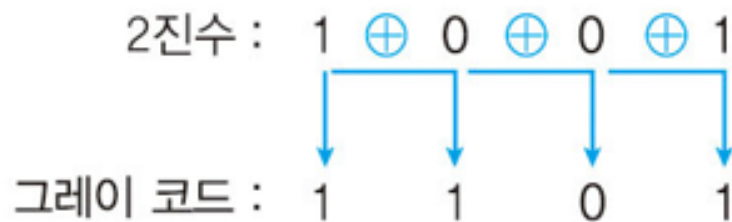


- 그레이 코드(Gray Code)

- 이진수의 인접하는 비트를 $\text{XOR} \oplus$ 해서 만든 코드
 - XOR : 두 비트를 비교해서 서로 다르면 1, 같으면 0
- A/D 변환기, 입출력장치 등에서 숫자표현에 사용
- 1Bit만으로 수치증가가 되기때문에 하드웨어적인 오류율이 적음
 - 0001(이진수1) \rightarrow 0010(이진수2) 두비트 변화
 - 0001(그레이코드1) \rightarrow 0011(그레이코드2) 한비트 변화
 - 모든 수치가 1비트씩만 변환됨. 연속데이터표현에 좋음
- 비가중치코드(non-weighted code)

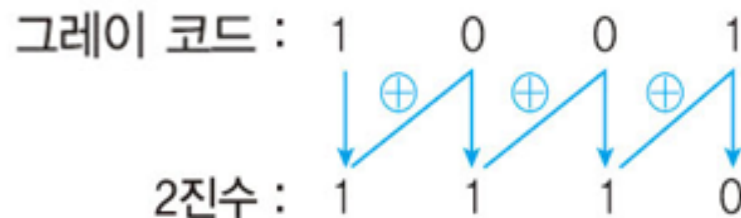
- 이진수를 **그레이 코드로 변환**하는 방법

- 1) 그레이 코드 첫번째는 이진수 첫번째 비트를 그대로 내려쓴다
- 2) 두번째부터는 이진수에서 왼쪽비트와 하나씩 XOR
- 3) 마찬가지로 방법으로 계속 이진수를 XOR



- 그레이 코드를 **이진수로 변환**하는 방법

- 1) 이진수 첫번째는 그레이 코드 첫번째 비트를 그대로 내려쓴다
- 2) 내려쓴 첫번째 이진수와 두번째 그레이 코드 비트를 XOR
- 3) 마찬가지로 방법으로 계속 그레이 코드와 이진수를 XOR



❖ 에러검출코드

• 패리티 검사 코드(Parity Bit)

- 코드의 오류를 검사하기 위해서 데이터 비트 외에 **1비트의 패리티 체크 비트를 추가**하여 전송
- **오류검출코드(1비트의 오류만 검출?)** / 오류정정은 하지 못함
- 종류
- Odd(기수) Parity(홀수패리티)
: 코드에서 1인 비트의 개수가 홀수가 되도록 0이나 1을 추가
- Even(우수) Parity(짝수패리티)
: 코드에서 1인 비트의 개수가 짝수가 되도록 0이나 1을 추가

- 해밍 코드(Hamming Code)

- 패리티를 활용한 오류검출 및 오류정정 코드
- 2비트 오류검출 및 1비트 오류정정
- 1, 2, 4, 8..... 2^n 비트 자리에 패리티가 추가됨
- 해밍 코드의 패리티 비트수 결정 공식(식을 만족하는 최소값)

$$n + p \leq 2^p - 1$$

n : 데이터비트수

p : 패리티비트수

– 해밍 코드에서 패리티 계산 방법

- n 번째 패리티는 $2^{(n-1)}$ 번째 위치하고, $2^{(n-1)}$ 개씩 건너뛰며, $2^{(n-1)}$ 개씩 비트를 묶어 패리티 계산

- 결국 : 1, 2, 4, 8....번째 위치에 패리티가 추가

- 1번에 위치하는 패리티는 **1번째 패리티**로, 1번부터 시작해서 1비트씩, 1개 건너뛰며 모아 패리티 계산

- 2번에 위치하는 패리티는 **2번째 패리티**로, 2번부터 시작해서 2비트씩, 2개 건너뛰며 모아 패리티 계산

- 4번에 위치하는 패리티는 **3번째 패리티**로, 4번부터 시작해서 4비트씩, 4개 건너뛰며 모아 패리티 계산

- **8번에 위치**하는 패리티는 **4번째 패리티**로, **8번부터 시작**해서 **8비트씩, 8개 건너뛰며** 모아 **패리티 계산**

:

:

값	...패리티추가...	P5(16)	P4(8)	P3(4)	P2(2)	P1(1)
1	...	0	0	0	0	1
2		0	0	0	1	0
3		0	0	0	1	1
4		0	0	1	0	0
5		0	0	1	0	1
6		0	0	1	1	0
7		0	0	1	1	1
8		0	1	0	0	0
9		0	1	0	0	1
10		0	1	0	1	0
11		0	1	0	1	1
12		0	1	1	0	0
13		0	1	1	0	1
14		0	1	1	1	0
15		0	1	1	1	1
16		1	0	0	0	0
:		1	0	0	0	1

EX) 데이터 1011을 기수패리티를 사용하는 해밍 코드로 변환하여 전송할 때 올바른 해밍 코드는?

- 1) 추가할 패리티 비트수는? 3비트(P1, P2, P3)
- 2) 패리티 위치는?



- 3) 패리티 계산은?



P1 → __P1__ 1 0 1 → 기수패리티이므로 **1 추가**



P2 → __P2__ 1 1 1 → **0 추가**



P3 → __P3__ 0 1 1 → **1 추가**



EX) 오류검출 및 정정

기수패리티를 사용하는 해밍코드 1010011의 오류를 검증하여라.



P1 → 1 1 0 1 → ok → 0



P2 → 0 1 1 1 → ok → 0



P3 → 0 0 1 1 → **ERROR** → 1

P3P2P1 순(역순)으로 구성 → **100** → 4번째 비트오류
→ 최종 정정코드



❖ 연산처리용 10진데이터표현

- BCD, EBCDIC 등은 외부적으로 10진데이터를 표현하는 방법
- 컴퓨터 내부에서 정수의 처리를 위해 사용하는 표현방법
 - 1) 2진데이터 표현(부호화 절대치, 1의보수, 2의보수)
 - 2) 10진데이터 표현(pack형, unpack형)

• Pack형(Packed Decimal Number Format)

- 10진데이터 연산처리에 사용
- 4비트로 10진수 하나를 나타냄
- 숫자를 묶어서 표현하고 **마지막에 부호(양수 1100, 음수 1101)**

• Unpack형(UnPacked, Zoned Decimal Number Format)

- 10진데이터 입출력에 사용
- 4비트 Zone(1111)과 4비트 숫자로 10진수 하나를 나타냄
- **마지막 숫자 Zone에 부호(양수 1100, 음수 1101)를 표시**

– Pack형 표현

+494의 표현

494C 로 표현(16진수)

0100 1001 0100 1100

-494의 표현

494D 로 표현(16진수)

0100 1001 0100 1101

– Unpack형(Zone형) 표현

+494의 표현

F 4 F 9 C 4 로 표현(16진수)

1111 0100 1111 1001 1100 0100

-2825의 표현

F 2 F 8 F 2 D 5 로 표현(16진수)

1111 0010 1111 1000 1111 0010 1101 0101