

2장. 수체계, 연산 및 코드



- 10진수, 2진수 (1, 2)
- 10진수-2진수 변환(3)
- 2진 산술연산(4)
- 2진수의 1의 보수, 2의 보수(5)
- 부호 수 및 산술연산(6, 7)
- 16진수, 8진수(8, 9)
- BCD, 디지털 코드 및 패리티(10, 11)

10진수(Decimal Numbers)



◎ 10진수

- 0, 1, ~, 9의 10개의 숫자 사용
- 숫자의 위치에 따라 가중치(weight)가 달라짐

.... 10^5 10^4 10^3 10^2 10^1 10^0 . 10^{-1} 10^{-2} 10^{-3} 10^{-4} 10^{-5} ...

예제 2-1) 47

예제 2-2) 568.23

2진수(Binary Numbers)

2진 계수

• 0000 → 0001 → 0010 → 0011 → 0100

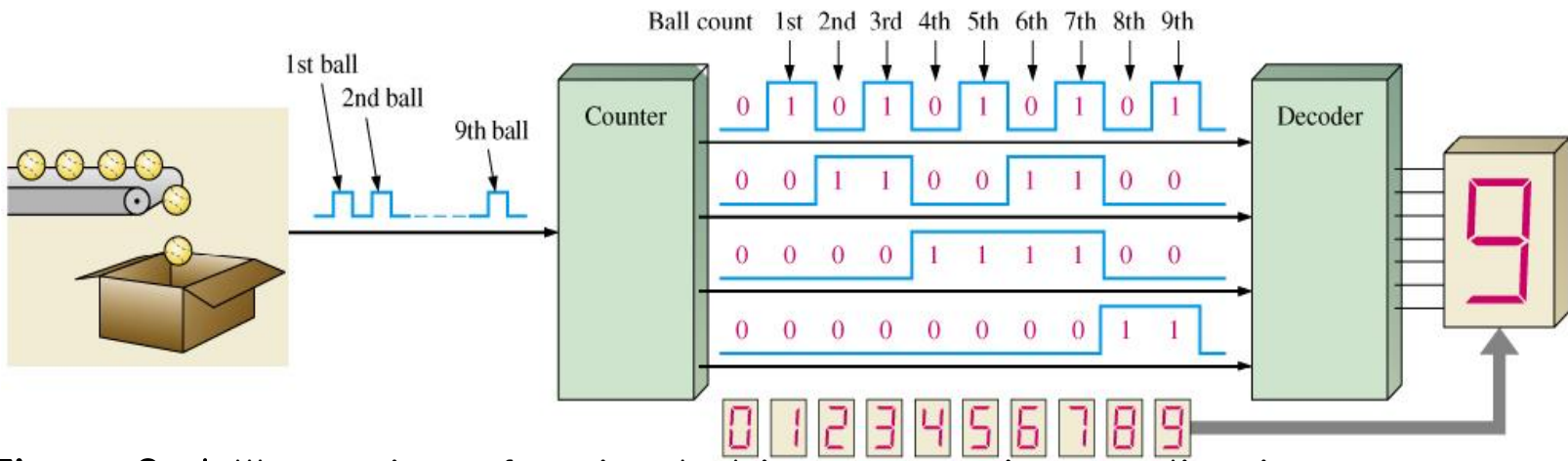


Figure 2-1 Illustration of a simple binary counting application.

2진수(Binary Numbers)

2진수의 가중치 구조(Weighting-structure)

$$2^{n-1} \dots 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 . \ 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \ 2^{-5} \dots 2^{-n}$$

TABLE 2-2 Binary weights.

Positive Powers of Two (Whole Numbers)									Negative Powers of Two (Fractional Number)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}

2진수-10진수 변환



- 예제 2-3 ; 2진 정수 1101101을 10진수로 변환하라

가중치 : $\dots 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 . 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \dots$

- 예제 2-4 ; 소수 2진수 0.1011을 10진수로 변환하라

10진수-2진수 변환



가중치의 합 방법(Sum-of-Weights Method)

예제 2-5 :

10진수 $\rightarrow$ 2진수

$$12 = 8 + 4 = 2^3 + 2^2 \Rightarrow 1100$$

$$25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0 \Rightarrow 11001$$

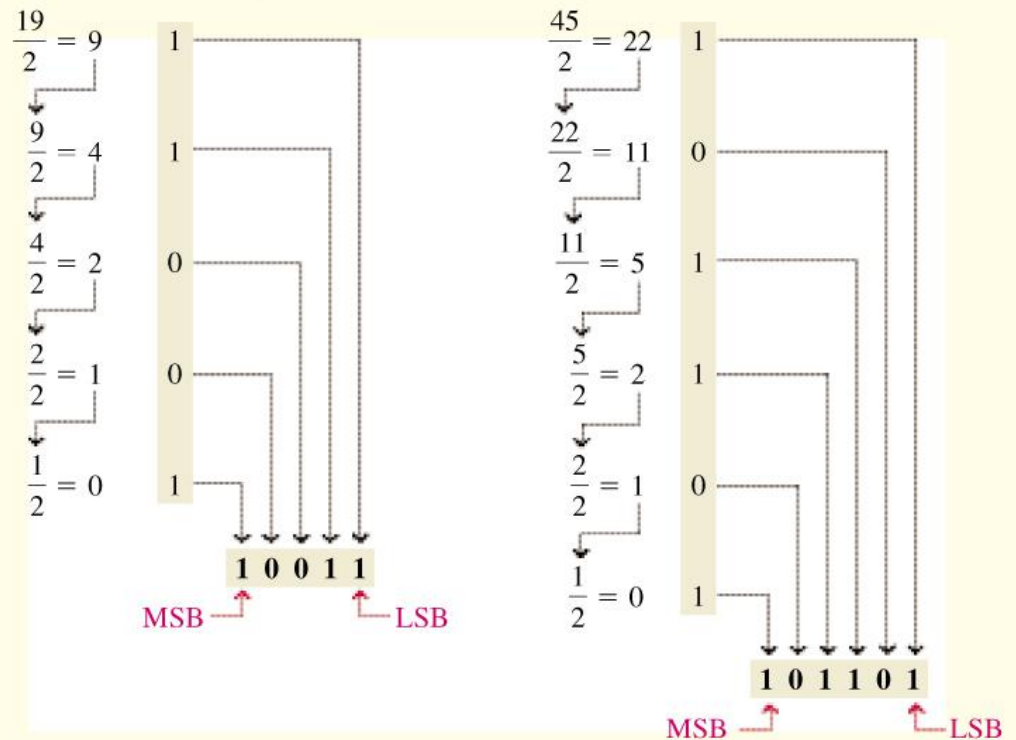
58

82

가중치 : $\dots 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \cdot \ 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \dots$

10진수-2진수 변환

- 반복적으로 2로 나누는 방법(Repeated Division-by-2 Method)



Related Problem Convert decimal number 39 to binary.

10진 소수의 2진수 변환



- 가중치의 합 방법(Sum-of-Weights Method)
- 반복적으로 2로 나누는 방법(Repeated Division-by-2 Method)

2진 산술(Binary Arithmetic)



- 2진 덧셈

2진 뱀셈

○ 2진 뱀셈



2진 곱셈



2진 나눗셈



EXAMPLE 2-11

Perform the following binary divisions:

(a) $110 \div 11$ (b) $110 \div 10$

Solution

	$\begin{array}{r} 10 \\ 11 \overline{)110} \\ \underline{11} \\ 000 \end{array}$	$\begin{array}{r} 2 \\ 3 \overline{)6} \\ \underline{6} \\ 0 \end{array}$		$\begin{array}{r} 11 \\ 10 \overline{)110} \\ \underline{10} \\ 10 \\ \underline{10} \\ 00 \end{array}$	$\begin{array}{r} 3 \\ 2 \overline{)6} \\ \underline{6} \\ 0 \end{array}$
--	--	---	--	---	---

Related Problem Divide 1100 by 100.

2진수의 1의 보수

2진수의 1의 보수 구하는 법

Binary number 0001100100101

응용 예

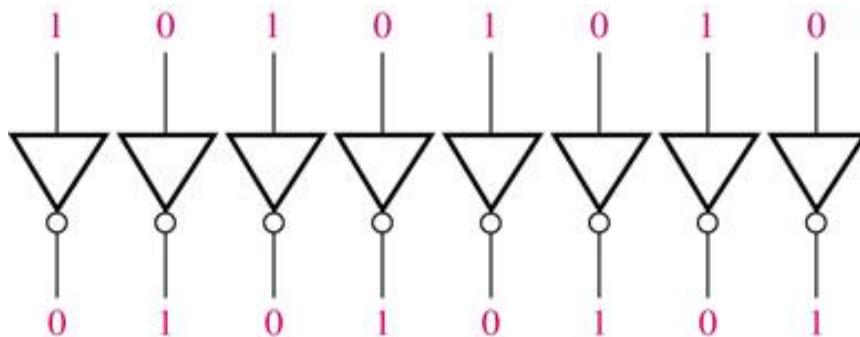


Figure 2-2 Example of inverters used to obtain the 1's complement of a binary number.

2진수의 2의 보수



2진수의 2의 보수 구하는 법

방법 1 :

(예 2-12) 10110010의 2의 보수

방법 2 :

(예 2-13) 10111000의 2의 보수

2진수의 2의 보수



응용 예

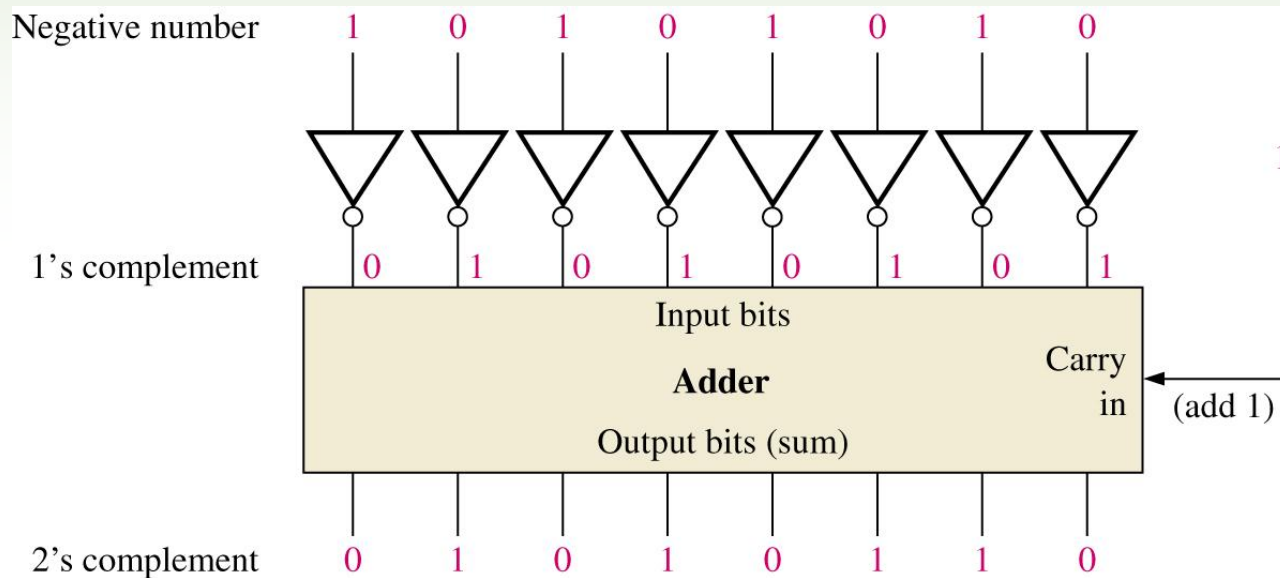


Figure 2-3 Example of obtaining the 2's complement of a negative binary number.

부호 수(Signed Number)



- 2진 부호 수 ; 2진수에서 부호있는 수를 표현.
=> 부호-크기, 1의 보수, 2의 보수
- 부호 비트(sign bit) ;
- 부호 크기 형태(Sign-Magnitude Form) ;

부호 수(Signed Number)



◎ 1의 보수 형태(1's Complement Form);

- 음수는 상응하는 양수의 1의 보수로 표현 ;
(예) -25 ; $+ 25$ 의 1의 보수로 표현

◎ 2의 보수 형태(2's Complement Form);

부호 수 예



예 2-14) 10진수 -39를 8비트 수로 표현하라.

(1) +39 00100111

(2) 부호-크기 :+39 00100111
 -39

(3) 1의 보수 :+39 00100111
 -39

(4) 2의 보수 :+39 00100111
 -39

부호 수의 10진 값



◎ 부호-크기

- ◎ 예제2-15 : 부호-크기로 표현된 2진수 10010101의 십진값은?

부호 수의 10진 값



◎ 1의 보수

- (1) 사인 비트의 자리에는 음의 가중치를 할당한다.
- (2) 양수 : 1이 있는 모든 비트들의 가중치를 합한다.
- (3) 음수 : 1이 있는 모든 비트들의 가중치를 합한 후 1을 더한다.

- ◎ 예제2-16 : 1의 보수로 표현된 2진수 00010111, 11101000의 십진값은?

부호 수의 10진 값



- 2의 보수

- 예제2-17 : 2의 보수로 표현된 2진수 01010110,
10101010의 십진값은?

가중치 -2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

정수의 표현 범위



◎ n 비트로 표현할 수 있는 수

전체 표현 가능한 수 : 2^n 개

2의 보수를 사용한 부호수에서 표현 가능한 수 :
 $-(2^{n-1}) \sim +(2^{n-1} - 1)$

부동-소수점 수 (Floating-Point Numbers)



○ 부동-소수점 수

비트 수를 증가시키지 않고, 매우 큰수와 작은수
그리고 정수와 소수부분을 갖는 수를 표현할 수 있다.

$$241,506,800 \Rightarrow \underbrace{.2415068}_{\text{가수(mantissa)}} \times 10^{\underbrace{9}_{\text{지수(exponent)}}}$$

단-정밀도(Single-Precision) : 32 비트

배-정밀도(Double-Precision) : 64 비트

확장-정밀도(Extended-Precision) : 80비트

단-정밀도 부동-소수점 2진수 (Single-Precision Floating-Point Binary Numbers)



S	Exponent(E)	Mantissa(fraction, F)
---	-------------	-----------------------

부호 ; 지수 ; 가수 ;
 1비트 8비트, 23비트
 실제 지수 + 127;
 $-126 \leq \text{실제지수} \leq 128$

- 2진수의 부동-소수점 형태 표현

$$1011010010001 = 1.011010010001 \times 2^{12} \quad (\text{양수라 가정})$$

$$\Rightarrow \begin{array}{|c|c|c|} \hline 0 & 10001011 & 011010010001000000000000 \\ \hline \end{array}$$

$$\begin{aligned} \text{양수} \quad & 12 + 127 \\ & = 139 \end{aligned}$$

단-정밀도 부동-소수점 2진수 (Single-Precision Floating-Point Binary Numbers)



- 2진수의 부동-소수점 형태 표현 $\Rightarrow (-1)^S(1+F)(2^{E-127})$

1	10010001	100011100010000000000000
---	----------	--------------------------

$\Rightarrow$ 바이어스된 지수 : $10010001 = 145$

$$\Rightarrow (-1)^1(1.10001110001)(2^{145-127})$$

$$= (-1)^1(1.10001110001)(2^{18})$$

$$= -1100011100010000000$$

- 예제 2-18) 10진수 3.248×10^4 을 단-정밀도 부동-소수점 2진수로 변환.

$$\Rightarrow 3.248 \times 10^4 = 32480 = 111111011100000_2 = 1.11111011100000 \times 2^{14}$$

$$\text{지수 : } 14 + 127 = 141 = 10001101_2$$

0	10001101	111110111000000000000000
---	----------	--------------------------

부호수의 산술연산



* 2의 보수 형태로 표현한 부호수의 가감승제

○ 덧셈

부호수의 산술연산(덧셈)



- 오버플로우 조건(Overflow condition)

부호수의 산술연산(뱌셈)



○ 뱌셈

예제 2-20)

$$\begin{array}{l} 00001000 - 00000011 \\ \Rightarrow 8 - 3 = 8 + (-3) = 5 \end{array}$$

$$\begin{array}{l} 00001100 - 11110111 \\ \Rightarrow 12 - (-9) = 12 + 9 = 21 \end{array}$$

부호수의 산술연산(곱셈)

- 곱셈 방법 1 ; 피승수를 승수만큼 더한다.

예제 2-21 : 01001101 X 00000100 (= 141X 4)

01001101	1회
+ 01001101	2회
10011010	
+ 01001101	3회
11100111	
+ 01001101	4회
100110100	곱

부호수의 산술연산(곱셈)



◎ 곱셈 방법 2 ;

1. 곱의 부호를 결정
2. 음수를 보수화 되지 않은
형태로 변경(2의 보수로 바꾼다)
3. 부분곱을 생성
4. 부분곱들의 합을 구한다.
5. 1단계에서의 부호가 음이면
결과에 2의 보수를 취하고,
양이면 그대로 둔 후,
부호 비트를 붙인다.

예제 2-22) 01010011 X 11000101

1. 곱의 부호 비트는 1(음)
2. 11000101 의 2의 보수 : 00111011
- 3,4. 크기비트만을 사용, 부분곱 생성

$$\begin{array}{r} 1010011 \\ \times 0111011 \\ \hline 1010011 \\ +1010011 \\ \hline 11111001 \end{array}$$

.

5. 부호비트가 1이므로 최종 곱
1001100100001에 대해 2의 보수를 취하고,
부호비트를 붙인다.
=> 1 0110011011111

부호수의 산술연산(나눗셈)



◎ 나눗셈

예제 2-23) $01100100 \div 00011001$

1. 몫의 부호를 결정, 몫을 0으로 초기화
2. 피제수에서 제수를 빼고 1을 몫에 더한다.
부분 나머지가 양이면 단계 3으로 간다.
부분 나머지가 0 이면 완결
부분 나머지가 음이면 몫에서 1을 빼고 완결
3. 부분 나머지에서 제수를 빼고 1을 몫에 더한다.
결과가 양이면 반복.
결과가 0이면 완결
결과가 음이면 몫에서 1을 빼고 완결

$$\begin{array}{r} 5. \quad 00011001 \\ + \quad 11100111 \\ \hline 00000000 \end{array}$$

몫 = 00000100

1. 몫의 부호는 양, 몫=00000000

2. 제수의 2의 보수 : 11100111

$$\begin{array}{r} 01100100 \\ + \quad 11100111 \\ \hline 01001011 \end{array}$$

몫 = 00000001

3. 01001011

$$\begin{array}{r} + \quad 11100111 \\ \hline 00110010 \end{array}$$

몫 = 00000010

4. 00110010

$$\begin{array}{r} + \quad 11100111 \\ \hline 00011001 \end{array}$$

몫 = 00000011

16진수 (Hexadecimal Numbers)



Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

16진수



- 16진 계수
- 2진수-16진수 변환
- 16진수-2진수 변환

16진수



○ 16진수-10진수 변환

* 방법 1 : 16진수 $\rightarrow$ 2진수 $\rightarrow$ 10진수

* 방법 2 : 가중치

16진수



10진수-16진수 변환

예제 2-28) 10진수 650을 16진수로 변환하라.

$$650/16 = 40.625 \rightarrow 0.625 \times 16 = 10 \Rightarrow A \quad (\text{몫 : 40, 나머지 : 10})$$

$$40/16 = 2.5 \rightarrow 0.5 \times 16 = 8 \Rightarrow 8 \quad (\text{몫 : 2, 나머지 : 8})$$

$$2/16 = 0.125 \rightarrow 0.125 \times 16 = 2 \Rightarrow 2 \quad (\text{몫 : 0, 나머지 : 2})$$

$$\Rightarrow 28A_{16}$$

16진수 덧셈



○ 덧셈 규칙

예제 2-29)

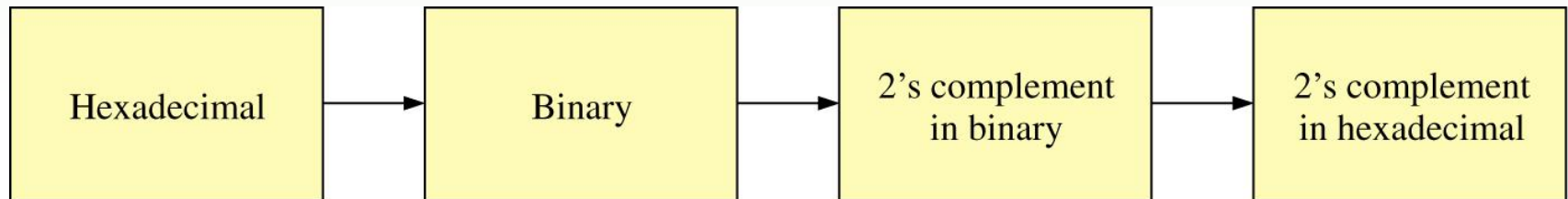
$$\begin{array}{r} 23_{16} \\ + 16_{16} \\ \hline 39_{16} \end{array}$$

$$\begin{array}{r} 58_{16} \\ + 22_{16} \\ \hline 7A_{16} \end{array}$$

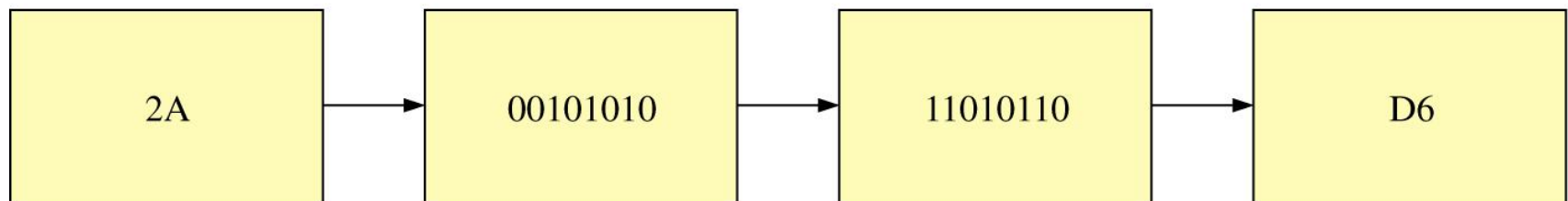
16진수 뱀셈



16진수에 대한 2의 보수 구하기 방법 1



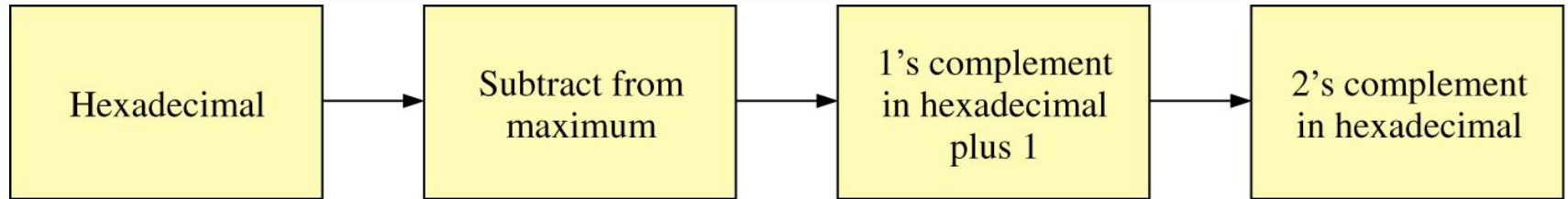
Example:



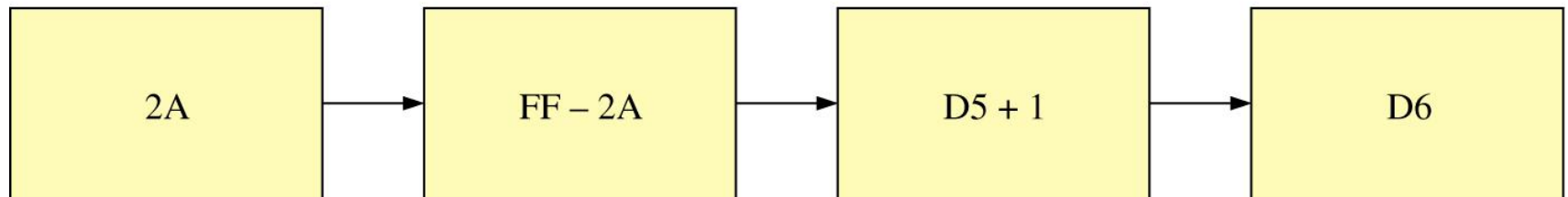
16진수 뱀셈



16진수에 대한 2의 보수 구하기 방법 2



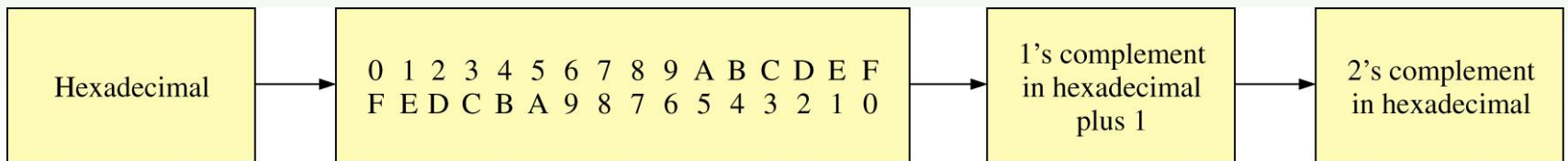
Example:



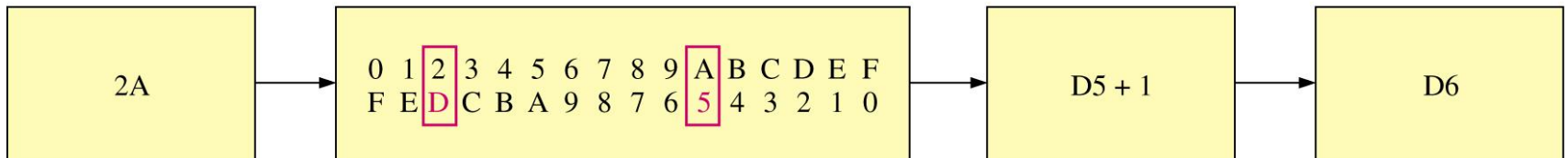
16진수 뱀셈



16진수에 대한 2의 보수 구하기 방법 3



Example:



16진수 뺄셈



예제 2-30) $84_{16} - 2A_{16}$, $C3_{16} - 0B_{16}$

방법 1)

$2A_{16} = 00101010$
 $\Rightarrow$ 보수는 $11010110 = D6_{16}$

방법 2)

$0B$ 의 보수는
 $\Rightarrow FF - 0B + 1 = F5$

8진 수 (Octal Numbers)



- 8진 수 - 10진 수 변환

8진수



◎ 10진수-8진수 변환

=> 10진수를 8로 반복적으로 나눈다.

$$359/8 = 44.875 \rightarrow 0.875 \times 8 = 7$$

(몫 : 44, 나머지 : 7)

$$44/8 = 5.5 \rightarrow 0.5 \times 8 = 4$$

(몫 : 5, 나머지 : 4)

$$5/8 = 0.625 \rightarrow 0.625 \times 8 = 5$$

(몫 : 0, 나머지 : 5)

$$\Rightarrow 547_8$$

8진수



- 8진수-2진수 변환

- 2진수-8진수 변환

2진 코드화된 10진 수(BCD)



- 10진수의 각 자리 숫자를 해당하는 2진수로 표현
- 8421 코드
- BCD 덧셈
 - 두 개의 BCD 수를 더한다.
 - 4비트 합이 9 이하이면 그대로 둔다.
 - 4비트 합이 9보다 크거나, 캐리가 발생하면 결과에 6을 더한다.
6을 더할 때, 캐리가 생기면 다음 4비트 그룹에 캐리를 더해준다.

8421 코드



예제 2-35) BCD 수 덧셈

0011 + 0100, 10000110 + 00010011

$$\begin{array}{r} 0011 \\ 0100 \\ \hline 0111 \end{array} \quad \begin{array}{l} 3 \\ + 4 \\ \hline 7 \end{array}$$

$$\begin{array}{r} 1000 \ 0110 \\ 0001 \ 0011 \\ \hline 1001 \ 1001 \end{array} \quad \begin{array}{l} 86 \\ + 13 \\ \hline 99 \end{array}$$

예제 2-36) BCD 수 덧셈

1001 + 0100, 00010110 + 00010101

$$\begin{array}{r} 1001 \\ 0100 \\ \hline 1101 \end{array} \quad \begin{array}{l} > 9 \\ + 6 \\ \hline \end{array} \quad \begin{array}{l} \text{add 6} \\ 0001 \ 0011 \Rightarrow 13 \end{array}$$

$$\begin{array}{r} 9 \\ + 4 \\ \hline 13 \end{array}$$

$$\begin{array}{r} 0001 \ 0110 \\ 0001 \ 0101 \\ \hline 0010 \ 1011 \end{array} \quad \begin{array}{l} 16 \\ + 15 \\ \hline 31 \end{array}$$

$$\begin{array}{r} 0010 \ 1011 \\ + 0110 \\ \hline 0011 \ 0001 \Rightarrow 31 \end{array} \quad \begin{array}{l} + 6 \\ \text{add 6} \end{array}$$

그레이 코드(Gray Code)



○ 그레이 코드

DECIMAL	BINARY	GRAY CODE	DECIMAL	BINARY	GRAY CODE
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

그레이 코드



- 2진수-그레이 코드 변환
- 그레이 코드- 2진수 변환

그레이 코드

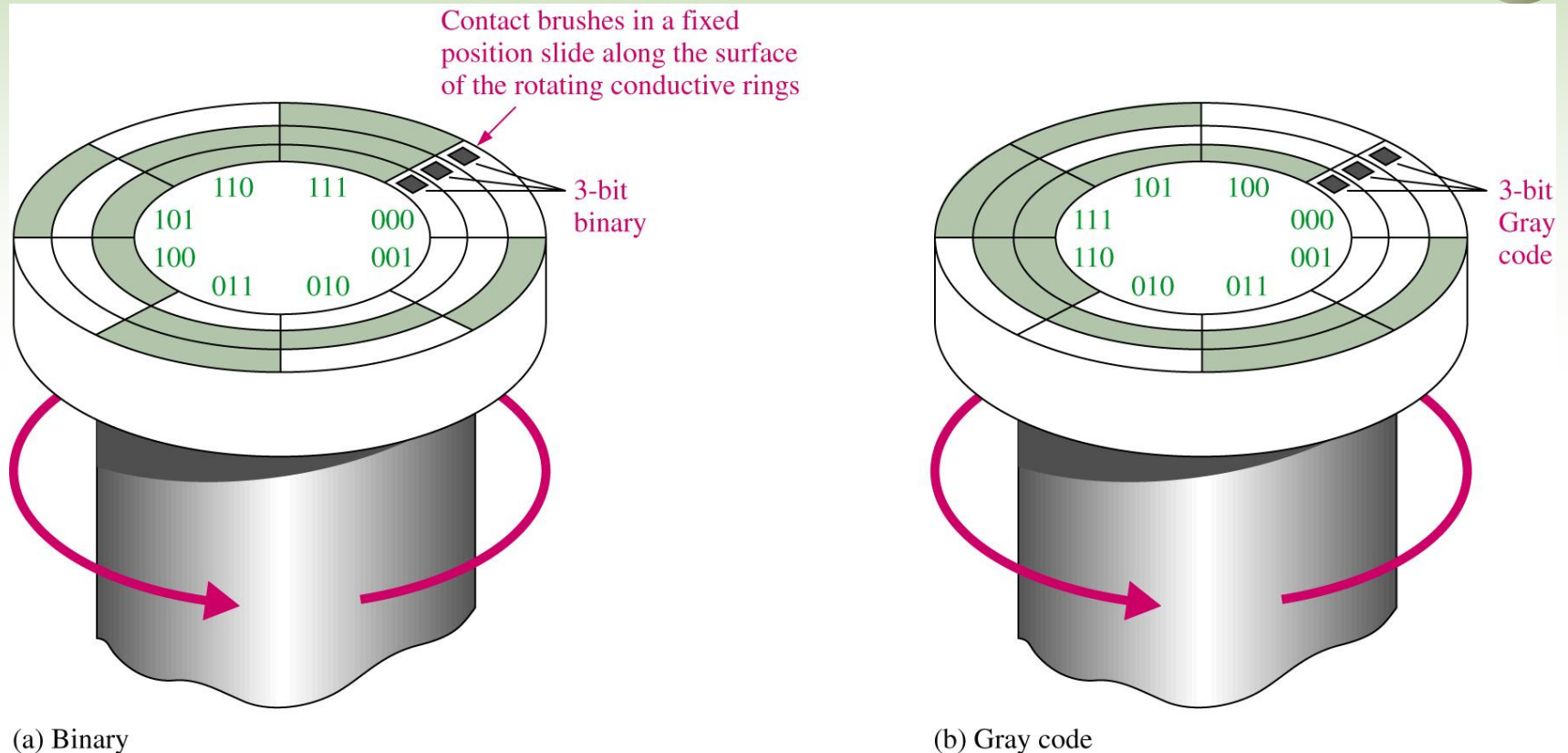


Figure 2-7 A simplified illustration of how the Gray code solves the error problem in shaft position encoders.

알파뉴메릭 코드(Alphanumeric Codes)



- ASCII(American Standard Code for Information Interchange)

=> 7비트 2진 코드로 표시되는 128개의 부호와 기호(0-127); 표 2-7

- ASCII 제어 문자

=> 컴퓨터, 프린터 등이 정보 교환에 필요한 코드(0-31)

- 확장된 ASCII 제어 문자

⇒ 그리스 문자, 비-영어 알파벳 등을 표시하기 위한 코드의 추가

⇒ 8비트 2진 코드 (128-255)

패리티 방법 (Parity Method)



● 에러 검출을 위한 패리티 방법

예제 2-39) 짝수 패리티 할당,
1010, 101101011111

EVEN PARITY		ODD PARITY	
P	BCD	P	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001