

05

디지털공학개론

| 디지털 코드

05

디지털 코드

1.가중치 코드/비가중치 코드

2.알파뉴페릭 코드

3.에러 검출 코드

1. 디지털 코드의 정의

디지털 코드

디지털 시스템에서는 각종 정보를 2진 비트들을 이용하여 정의하며 정보란 수치뿐 만 아니라 알파벳(대·소문자), 각종 기호 및 특수 문자들을 의미함

→ 디지털 2진 코드는 2진수로 이루어진 코드

가중치 코드 /
비가중치 코드

알파 뉴메릭 코드

특수 목적용 코드

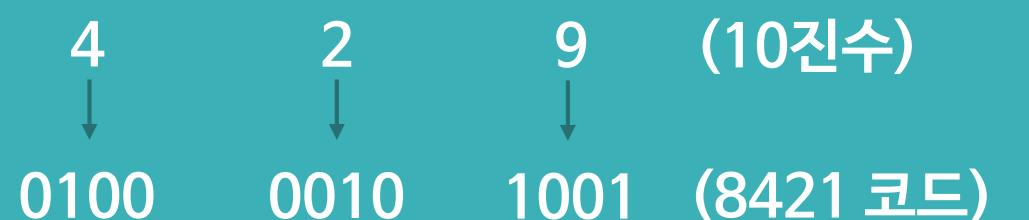
2. 10진수 표현을 위한 2진 코드

| code 10진수 | 8421 코드 | 3초과 코드 | 8421 코드 | 비퀴너리 5043210 | 존슨 코드 |
|--------------|---------|--------|---------|-----------------|-------|
| 0 | 0000 | 0011 | 0000 | 0100001 | 00000 |
| 1 | 0001 | 0100 | 0111 | 0100010 | 00001 |
| 2 | 0010 | 0101 | 0110 | 0100100 | 00011 |
| 3 | 0011 | 0110 | 0101 | 0101000 | 00111 |
| 4 | 0100 | 0111 | 0100 | 0110000 | 01111 |
| 5 | 0101 | 1000 | 1011 | 1000001 | 11111 |
| 6 | 0110 | 1001 | 1010 | 1000010 | 11110 |
| 7 | 0111 | 1010 | 1001 | 1000100 | 11100 |
| 8 | 1000 | 1011 | 1000 | 1001000 | 11000 |
| 9 | 1001 | 1100 | 1111 | 1010000 | 10000 |

3. BCD(Binary coded Decimal ; 8421) 코드

(1) BCD 코드

- 10진수를 2진 비트들로 표현하는 가장 보편적인 방법으로 네 개의 비트들이 왼쪽부터 각각 8, 4, 2, 1의 가중치를 가지도록 하는 방법
- 인코딩(Encoding) 과정이 용이함
- 사용 불가 코드가 존재 ("1010" ~ "1111")



3. BCD(Binary coded Decimal ; 8421) 코드

(2) BCD 코드 연산의 문제점

X

틀린 예

$$\begin{array}{r} 1 \ 3 \\ + \ 8 \\ \hline 2 \ 1 \end{array} \quad \begin{array}{r} 0 \ 0 \ 0 \ 1 \\ + \ 0 \ 0 \ 0 \ 0 \\ \hline 0 \ 0 \ 0 \ 1 \\ \quad (1) \end{array} \quad \begin{array}{r} 0 \ 0 \ 1 \ 1 \\ + \ 1 \ 0 \ 0 \ 0 \\ \hline 1 \ 0 \ 1 \ 1 \\ \quad (11) \end{array}$$

O

바른 예

$$\begin{array}{r} 1 \ 0 \ 0 \ 0 \\ + \ 0 \ 1 \ 0 \ 1 \\ \hline 1 \ 1 \ 0 \ 1 \text{ (13의 등가 2진수)} \end{array} \quad \begin{array}{r} + \ 0 \ 1 \ 1 \ 0 \text{ (8421 코드로 바꾸기 위해 6을 더함)} \\ \hline 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \text{ (13의 등가 8421 코드)} \end{array}$$

이로 인하여 BCD code 연산은 다소 복잡한 과정이 필요함

4. 3초과 코드(Excess-3 code)

(1) 3초과 코드

- BCD 연산이 복잡한 문제를 개선하기 위해 만들어짐
- 비가중치이나 스스로 보수를 만들어 낼 수 있는 이른바 자기 보수 (self - complement)형 코드로서 감산을 수행 할 수 있다는 장점이 있음

자기 보수의 성질

| 10진수 | 3 | 6 | 9의 보수 관계 |
|----------|------|------|----------|
| 3초과 code | 0110 | 1001 | 1의 보수 관계 |

4. 3초과 코드(Excess-3 code)

(2) 3초과 코드(Excess-3 code) 연산의 예

- 두 개의 3초과 코드값을 가산 시 사용되지 않는 코드값 (0000, 0001, 0010, 1101, 1110, 1111)이 존재함
- 캐리(Carry) 발생 여부에 따라 보정 과정을 거쳐 완전한 결과값을 유도함

$$\begin{array}{r} \begin{array}{r} 1 & 5 \\ + & 1 & 5 \\ \hline 3 & 0 \end{array} & \begin{array}{r} 0 & 1 & 0 & 0 \\ + & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 \end{array} & \begin{array}{r} 1 & 0 & 0 & 0 \\ + & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array} \rightarrow 15 \text{의 3초과값} \\ & \begin{array}{r} 0 & 0 & 1 & 1 \\ + & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 \end{array} \rightarrow 15 \text{의 3초과값} \\ & \begin{array}{r} 1 & 0 & 0 & 1 \\ + & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 \end{array} \rightarrow \text{우측에 자리올림이 발생했으므로 2진값} \\ & \begin{array}{r} 0 & 0 & 1 & 1 \\ + & 0 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 \end{array} \rightarrow \text{우측에 0011을 더함} \\ & \begin{array}{r} 0 & 1 & 1 & 0 \\ + & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 \end{array} \rightarrow 30 \text{의 3초과값} \end{array}$$

5. 그레이 코드(Gray Code)

(1) 그레이 코드

| 10진수 | 2진 코드 | Gray 코드 | 10진수 | 2진 코드 | Gray 코드 |
|------|-------|---------|------|-------|---------|
| 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 |

5. 그레이 코드(Gray Code)

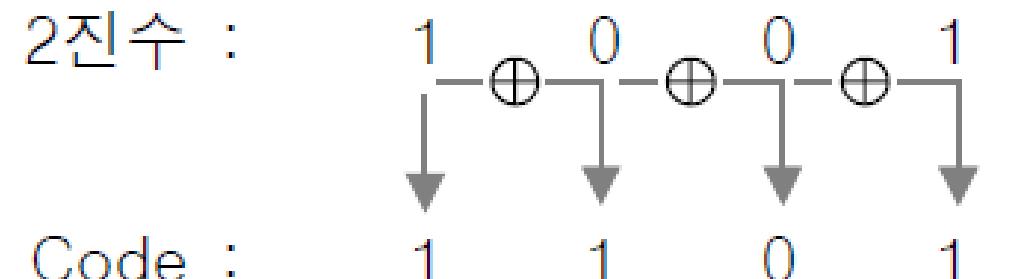
(2) 그레이 코드의 특징

- 산술적 연산이 불가능함
- 인접한 비트 중에서 그 값의 변화가 1비트만 연속적으로 변화함
- 비가중치 2진 순회 코드로 변화함
- 입출력 장치, 아날로그-디지털 변환기 등에서 사용함

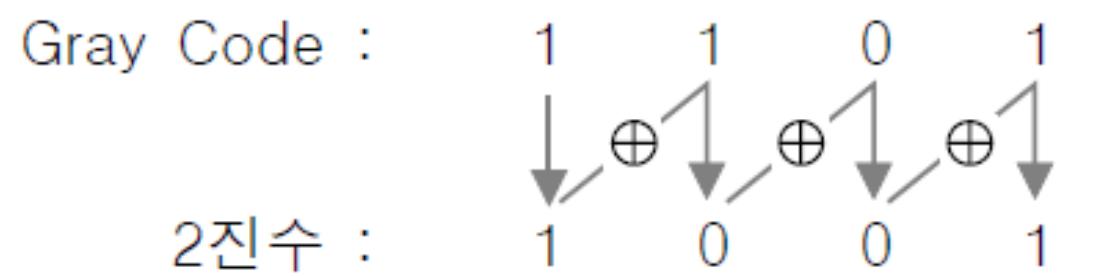
5. 그레이 코드(Gray Code)

(3) 2진수와 그레이 코드의 상호 변환

2진수를 그레이 코드로 변환 방법



그레이 코드를 2진수로 변환 방법



6. 그 밖의 코드

| Code 10 진수 | 2421 코드 | 7421 코드 | 51111 코드 | 2-out of 5 코드 | 링 카운터 코드 |
|---------------|------------|------------|-------------|------------------|------------|
| 0 | 0000 | 0000 | 00000 | 00011 | 000000001 |
| 1 | 0001 | 0111 | 00001 | 00101 | 000000010 |
| 2 | 0010 | 0110 | 00011 | 00110 | 0000000100 |
| 3 | 0011 | 0101 | 00111 | 01001 | 0000001000 |
| 4 | 0100 | 0100 | 01111 | 01010 | 0000010000 |
| 5 | 1011 | 1010 | 10000 | 11000 | 0000100000 |
| 6 | 1100 | 1001 | 11000 | 10001 | 0001000000 |
| 7 | 1101 | 1000 | 11100 | 10010 | 0010000000 |
| 8 | 1110 | 1111 | 11110 | 10100 | 0100000000 |
| 9 | 1111 | 1110 | 11111 | 11000 | 1000000000 |

05

디지털 코드

1.가중치 코드/비가중치 코드

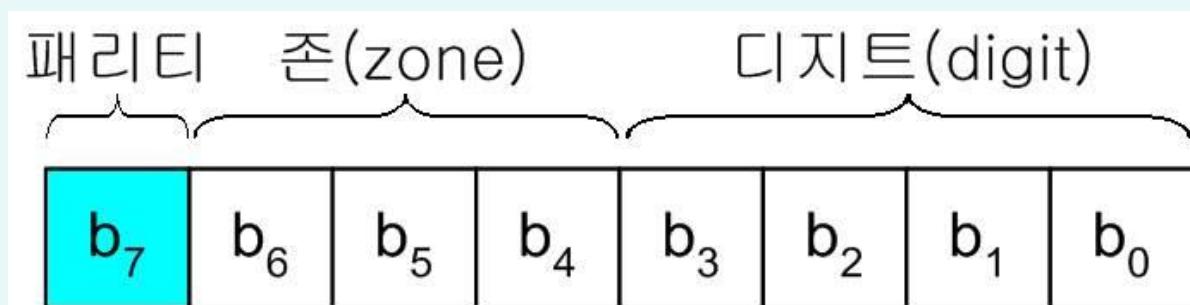
2.알파뉴페릭 코드

3.에러 검출 코드

1. ASCII [American Standard Code for Information Interchange] 코드

(1) ASCII 코드의 구성

- 미국 국립 표준 연구소(ANSI)가 제정한 정보 교환용 미국 표준 코드
- 128가지의 문자를 표현 가능



| Parity | Zone bit | | | | Digit Bit | | | |
|--------|----------|---|---|--------------------|-----------|---|---|--|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Check | 1 | 0 | 0 | 영문자 A~O(0001~1111) | | | | |
| | 1 | 0 | 1 | 영문자 P~Z(0000~1010) | | | | |
| | 0 | 1 | 1 | 숫자 0~9(0000~1001) | | | | |

1. ASCII [American Standard Code for Information Interchange] 코드

(2) 표준 ASCII 코드표

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|
| 0 | NUL | SOH | STX | ETX | EOT | ENQ | ACK | BEL | BS | TAB | LF | VT | FF | CR | SO | SI |
| 1 | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAN | EM | SUB | ESC | FS | GS | RS | US |
| 2 | | ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | . | / |
| 3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | | = | > | ? |
| 4 | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5 | P | Q | R | S | T | U | V | W | X | Y | Z | [| ₩ |] | ^ | - |
| 6 | ` | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7 | p | q | r | s | t | u | v | w | x | y | z | { | | } | ~ | |

2. 표준 BCD 코드

(1) 표준 BCD의 구성

- 6비트로 하나의 문자를 표현
- 최대 64문자까지 표현 가능한 코드

| Parity | Zone bit | | Digit Bit | | | | |
|--------|----------|---|-------------|--------------------|---|---|---|
| | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C | 1 | 1 | 1 | 영문자 A~I(0001~1001) | | | |
| | 1 | 0 | 1 | 영문자 J~R(0001~1001) | | | |
| | 0 | 1 | 1 | 영문자 S~Z(0010~1001) | | | |
| | 0 | 0 | 1 | 숫자 0~9(0001~1010) | | | |
| | 혼용 | | 특수문자 및 기타문자 | | | | |

2. 표준 BCD 코드

(2) 표준 ASCII 코드표

| 문자 | C ZZ8421 |
|----|----------|----|----------|----|----------|----|----------|----|----------|
| A | 0 110001 | J | 1 100001 | S | 1 010010 | 1 | 0 000001 | = | 0 001011 |
| B | 0 110010 | K | 1 100010 | T | 0 010011 | 2 | 0 000010 | > | 1 001100 |
| C | 1 110011 | L | 0 100011 | U | 1 010100 | 3 | 1 000011 | + | 0 010000 |
| D | 0 110100 | M | 1 100100 | V | 0 010101 | 4 | 0 000100 | , | 1 011011 |
| E | 1 110101 | N | 0 100101 | W | 0 010110 | 5 | 1 000101 |) | 0 011100 |
| F | 1 110110 | O | 0 100110 | X | 1 010111 | 6 | 1 000110 | % | 1 011101 |
| G | 0 110111 | P | 1 100111 | Y | 1 011000 | 7 | 0 000111 | ? | 0 011111 |
| H | 0 111000 | Q | 1 101000 | Z | 0 011001 | 8 | 0 001000 | - | 1 100001 |
| I | 1 111001 | R | 0 101001 | | | 9 | 1 001001 | @ | 1 111010 |
| | | | | | | 0 | 1 001010 | \$ | 1 111111 |

3. EBCDIC [Extended Binary Coded Decimal Interchange Code] 코드

(1) EBCDIC 구성

- 대형 컴퓨터와 IBM 계열 컴퓨터에서 많이 사용되고 있는 8비트 코드(IBM에서 개발)
- 256종류의 문자 코드를 표현할 수 있는 영숫자 코드

| b ₉ | b ₈ b ₇ b ₆ b ₅ | b ₄ b ₃ b ₂ b ₁ |
|----------------|---|---|
| 패리티 | 존(zone) | 디지트(digit) |
| 1 | 4 | 4 |

| b ₈ b ₇ | | b ₆ b ₅ | |
|-------------------------------|--------|-------------------------------|-------|
| 0 0 | 통신제어문자 | | |
| 0 1 | 특수문자 | | |
| 1 0 | 소문자 | 0 0 | a ~ i |
| | | 0 1 | j ~ r |
| | | 1 0 | s ~ z |
| | | 1 1 | |
| 1 1 | 대문자/숫자 | 0 0 | A ~ I |
| | | 0 1 | J ~ R |
| | | 1 0 | S ~ Z |
| | | 1 1 | 0 ~ 9 |

3. EBCDIC [Extended Binary Coded Decimal Interchange Code] 코드

(2) EBCDIC 코드표

| 16진 | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-----|------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 2진 | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| 0 | 0000 | NUL | SOH | STX | ETX | | HT | | DEL | | | | VT | FF | CR | SO | SI |
| 1 | 0001 | DLE | | | | | | BS | | CAN | EM | | | IFS | IGS | IRS | IUS |
| 2 | 0010 | | | | | LF | ETB | ESC | | | | | | ENQ | ACK | BEL | |
| 3 | 0011 | | | SYN | | | | EOT | | | | | | NAK | | SUB | |
| 4 | 0100 | space | | | | | | | | [| . | | (| + | | | |
| 5 | 0101 | & | | | | | | | | ! | \$ | * |) | | ^ | | |
| 6 | 0110 | - | / | | | | | | | | , | % | _ | > | ? | | |
| 7 | 0111 | | | | | | | | ' | : | # | @ | ' | = | " | | |
| 8 | 1000 | | a | b | c | d | e | f | g | h | i | | | | | | |
| 9 | 1001 | | j | k | l | m | n | o | p | q | r | | | | | | |
| A | 1010 | | ~ | s | t | u | v | w | x | y | z | | | | | | |
| B | 1011 | | | | | | | | | | | | | | | | |
| C | 1100 | { | A | B | C | D | E | F | G | H | I | | | | | | |
| D | 1101 | } | J | K | L | M | N | O | P | Q | R | | | | | | |
| E | 1110 | \ | S | T | U | V | W | X | Y | Z | | | | | | | |
| F | 1111 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | | | | |

4. 유니코드(Unicode)

- ASCII 코드의 한계성을 극복하기 위하여 개발된 인터넷 시대의 표준 유니코드.
현재 버전 5.1.0.0(2008. 4. 4)까지 개발
- 미국, 유럽, 동아시아, 아프리카, 아시아 태평양 지역 등의 주요 언어들에 적용
- 유니코드는 유럽, 중동, 아시아 등 거의 대부분의 문자를 포함하고 있으며,
10만개 이상의 문자로 구성
- 아시아의 중국, 일본, 한국, 타이완, 베트남, 싱가포르에서 사용하는
- 표의 문자(한자) 70,207개를 표현
- 구두표시, 수학기호, 전문기호, 기하학적 모양, 딩벳 기호 등을 포함
- 앞으로도 계속해서 산업계의 요구나 새로운 문자들을 추가

05

디지털 코드

1.가중치 코드/비가중치 코드

2.알파뉴페릭 코드

3.에러 검출 코드

1. 패리티 비트

(1) 패리티 비트의 특징

- 7비트 ASCII 코드에 패리티 비트를 추가한 코드
- 짝수패리티(even parity)
- 홀수패리티(odd parity)
- 데이터 전송과정에서 에러 검사를 위한 추가비트
- 에러 검출만 가능하며, 여러 비트에 에러가 발생할 경우에는 검출이 안될 수도 있음

1. 패리티 비트

(2) 패리티 비트의 표현

| 데이터 | 짝수패리티 | 홀수패리티 |
|-----|-----------|-----------|
| ... | ... | ... |
| A | 0 1000001 | 1 1000001 |
| B | 0 1000010 | 1 1000010 |
| C | 1 1000011 | 0 1000011 |
| D | 0 1000100 | 1 1000100 |
| ... | ... | ... |

1. 패리티 비트

(3) 병렬 패리티

- 패리티를 블록 데이터에 적용
- 가로와 세로 데이터들에 대해서 패리티를 적용하면 에러를 검출
- 그 위치를 찾아 정정

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |

2. 해밍 코드

(1) 해밍 코드의 특징

- 패리티 비트는 에러 검출만 할 수 있지만 해밍 코드는 에러를 검출할 뿐만 아니라 수정까지 가능함
- 이 코드는 패리티 비트를 정보 비트 수에 따라 필요한 만큼 사용하고 그 코드 그룹의 적당한 위치에 배치함
- 만약 정보 비트의 수를 d 로 표시하면 그때 패리티 비트 수 p 는 아래의 식에 의해 결정함
- $2^p \geq d + p + 1$ ($k = d + P$ (단, k 는 해밍 비트 수))

2. 해밍 코드

(2) 해밍 코드 생성 방법

ex) 정보 데이터가 1011일 때 우수 패리티를 사용하는 해밍 코드 생성

$$d = 4, 2^p \quad 4 + p + 1 \text{을 만족하는 } p = 3 \quad (k = d + P = 7)$$

| 전송 bit의 위치 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------|-------|-------|---|-------|---|---|---|
| 패리티 비트의 위치 | P_0 | P_1 | 1 | P_2 | 0 | 1 | 1 |
| 전송 bit의 위치 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 결정 | P_0 | P_1 | 1 | P_2 | 0 | 1 | 1 |
| 결정 | P_0 | P_1 | 1 | P_2 | 0 | 1 | 1 |
| 결정 | P_0 | P_1 | 1 | P_2 | 0 | 1 | 1 |

우수 패리티를 사용하므로 $P_0 = 0, P_1 = 1, P_2 = 0$ 결정

2. 해밍 코드

(2) 해밍 코드 생성 방법

ex) 수신된 해밍 코드가 “0110001”일 때 에러 여부 검출

(단, 패리티 방식은 우수 패리티 방식 사용)

$K = 7, \therefore 2^p - 7 + 1$ 이므로 $p = 3$ (P_0, P_1, P_2)
check 비트를 C_0, C_1, C_2 라 할 때

| 전송 bit의 위치 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------------|---|---|---|---|---|---|---|
| C_0 결정 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| C_1 결정 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| C_2 결정 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

우수 패리티를 사용하므로 $C_0 = 0, C_1 = 1, C_2 = 1$ 결정
 $C_2 C_1 C_0 = 100_{(2)} = 6_{(10)}$ 이므로 6번 비트에서 에러 발생

05

디지털 코드

- 학습정리

● 2진 코드

(1) BCD

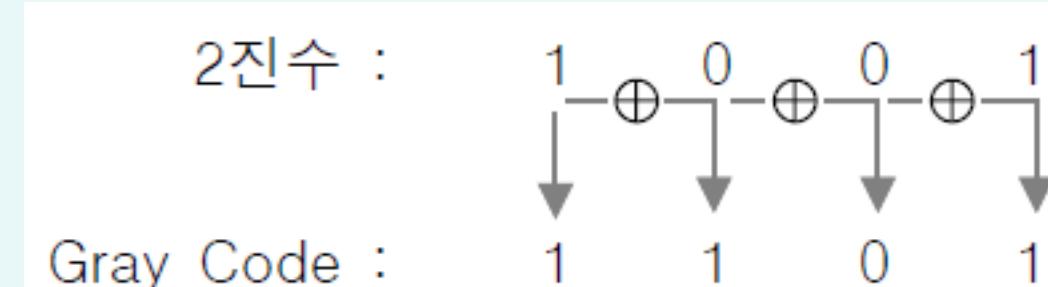
- 구성이 간단하나, 연산 과정이 복잡
- 사용 불가 코드가 존재

(2) 3초과 코드

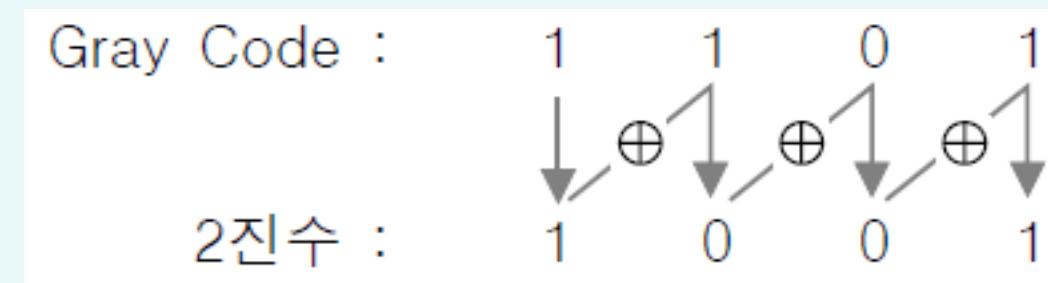
- BCD 코드를 이용한 연산이 어려움을 개선
- BCD + '0011'
- 비가중치 코드
- 자기 보수 코드이므로 연산이 가능

(3) 그레이 코드

- 진수를 그레이 코드로 변환 방법



- 그레이 코드를 2진수로 변환 방법



● 알파뉴메릭 코드(alphanumeric code)

(1) ASCII Code(American Standard Code for information interchange Code)

- 정보 교환용(Data 통신용) 미국 표준코드
- 7비트로 (3개의 Zone Bit + 4개의 Digit Bit)로 구성
- 최대 2^7 개(128개)의 문자 표현이 가능한 코드

(2) EBCDIC Code(Extended Binary Coded Decimal Interchange Code)

- 정보 표현의 제약점을 극복하기 위해 BCD 코드를 확장한 코드
- 8비트로 (4개의 Zone Bit + 4개의 Digit Bit)로 구성
- 최대 2^8 개(256개)의 문자 표현이 가능한 코드
- 대형 시스템의 정보 처리용으로 사용

● 패리티 비트와 해밍 코드

(1) 패리티 bit 기법

- 에러 검출에 이용(기수 또는 우수)

(2) 해밍 코드

- 에러 검출 및 수정 코드
- $2^p \geq d + p + 1$ $k = d + P$ (단, k 는 해밍 비트 수)