Report 1

소프트웨어학부 20194651 조나단

# 아래 크기만 표현된 2진수의 4칙연산(+, -, \*, /)을 수행하시오. 그리고 각 2진수를 10진수로 변환하여 계산 결과가 맞는지 확인하시오.

## 110111 + 1010

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

각 2진수는 110111(2) = 55(10), 1010(2) = 10(10) 이고

55(10) + 10(10) = 65(10) = 1000001(2) 임으로 계산 결과가 맞다.

## 110111 – 1010

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 2 |  |  |  |
|  |  | 1 | 1 | 0 | 1 | 1 | 1 |
| - |  |  |  | 1 | 0 | 1 | 0 |
|  |  | 1 | 0 | 1 | 1 | 0 | 1 |

각 2진수는 110111(2) = 55(10), 1010(2) = 10(10) 이고

55(10) – 10(10) = 45(10) = 101101(2) 임으로 계산 결과가 맞다.

## 10011010 \* 1010

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| \* |  |  |  |  |  |  |  | 1 | 0 | 1 | 0 |
|  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  |  |  | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |  |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |
|  | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |  |  |  |
|  | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

각 2진수는 10011010(2) = 154(10), 1010(2) = 10(10) 이고

154(10) \* 10(10) = 1540 = 11000000100(2) 임으로 계산 결과가 맞다.

## 10111101 / 1001

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

각 2진수는 10111101(2) = 189(10), 1001(2) = 9(10) 이고

189(10) / 9(10) = 21(10) = 10101(2) 임으로 계산 결과가 맞다.

# 아래 10진수들은 2진수로 변환(8 비트 2의 보수 표현) 하고, 아래 산술 연산을 수행하시오. 각 결과 값을 10진수로 변환하여 답이 맞는지 확인하시오.

## 57 + 23

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

57(10) = 00111001(2), 23(10) = 00010111(2) 이고

57(10) + 23(10) = 80(10) = 01010000(2) 임으로 계산 결과가 맞다.

## 57 – 23

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 1 |  | 1 |  |  |  |  |
|  | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| + | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| (1) | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

57(10) = 00111001(2), 23(10) = 00010111(2), -23(10) = 11101001 (2) (2의 보수) 이고

57(10) - 23(10) = 34(10) = 00100010(2) 임으로 계산 결과가 맞다.

## -57 + 23

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

57(10) = 00111001(2), -57(10) = 11000111(2) (2의 보수), 23(10) = 00010111(2) 이고

-57(10) + 23(10) = -34(10) = 11011110(2) (2의보수) 임으로 계산 결과가 맞다.

## -57 – 23

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 |  |  |  | 1 | 1 | 1 |  |
|  | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| + | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| (1) | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

57(10) = 00111001(2), -57(10) = 11000111(2) (2의 보수),

23(10) = 00010111(2), -23(10) = 11101001(2) (2의 보수) 이고

-57(10) - 23(10) = -80(10) = 10110000(2) 임으로 계산 결과가 맞다.

# 기억장치에 12-비트 데이터 ‘011101011001’ 이 저장되어 있다. 만약 그 데이터가 아래와 같은 2진 코드로 표현된 값이라면, 저장된 값이 가리키는 10진수가 무엇인지 해석하라.

## BCD 코드

0111 0101 1001(BCD) = (7 \* 10^2) + (5 \* 10^1) + (9 \* 10^0) = **759(10)**

## 3증수 코드

0111 0101 1001(excess-3) =

(0111(2) – 0011(2)) \* 10^2 + (0101(2) – 0011(2)) \* 10^1 + (1001(2) – 0011(2)) \* 10^0 = **426(10)**

## 84-2-1 코드

0111 0101 1001(84-2-1)

= ((1\*4 + 1\*-2 + 1\*-1) \* 10^2) + ((1\*4 + 1\*-1) \* 10^1) + ((1\*8 + 1\*-1) \* 10^0) = **137(10)**

## 그레이 코드

011101011001(gray) = 010110010001(2) = **1425(10)**

# ASCII 코드로 표현된 어떤 정보의 각 단어에 Even Parity 비트를 추가한 다음에, 16진수로 변환한 결과가 아래와 같다.

47 E1 74 65 A0 B2 B7

## 2진 비트 패턴으로 변환하고, ASCII 코드를 해독하여 나타내시오.

0 100 0111 = ‘G’

1 110 0001 = ‘a’

0 111 0100 = ‘t’

0 110 0101 = ‘e’

1 010 0000 = ‘ ‘ (SPACE)

1 011 0010 = ‘2’

1 011 0111 = ‘7’

47 E1 74 65 A0 B2 B7 = “Gate 27”

## Odd parity 방식으로 변경한 다음에 다시 16진수로 표현하시오.

0 100 0111 → 11000111 = C7

1 110 0001 → 01100001 = 61

0 111 0100 → 11110100 = F4

0 110 0101 → 11100101 = E5

1 010 0000 → 00100000 = 20

1 011 0010 → 00110010 = 32

1 011 0111 → 00110111 = 37

47 E1 74 65 A0 B2 B7 → C7 61 F4 E5 20 32 37

# 8-비트 데이터를 저장하는 컴퓨터 기억장치에서 오류 검출 및 정정을 위하여 해밍 코드가 사용되고 있다. 저장할 데이터가 ‘10010100’ 일 때 아래 물음에 답하시오.

## 검사비트들의 값을 구하고, 데이터 비트들과 함께 배치한 해밍 코드를 구하라.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 번호 | 12 | 11 | 10 | 9 | **8** | 7 | 6 | 5 | **4** | 3 | **2** | **1** |
| 값 | 1 | 0 | 0 | 1 | **0** | 0 | 1 | 0 | **0** | 0 | **1** | **1** |
| 배치 | D8 | D7 | D6 | D5 | **P8** | D4 | D3 | D2 | **P4** | D1 | **P2** | **P1** |

P1: 비트 번호 (1), 3, 5, 7, 9, 11의 짝수 패리티 → 1

P2: 비트 번호 (2), 3, 6, 7, 10, 11의 짝수 패리티 → 1

P4: 비트 번호 (4), 5, 6, 7, 12의 짝수 패리티 → 0

P8: 비트 번호 (8), 9, 10, 11, 12의 짝수 패리티 → 0

## 데이터가 저장되어 있는 동안에 오류가 발생하여 해밍 코드의 MSB 가 0으로 변경되었다면, 오류 검사 과정에서 생성되는 검사 비트들은 어떤 값이 되는가?

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 번호 | 12 | 11 | 10 | 9 | **8** | 7 | 6 | 5 | **4** | 3 | **2** | **1** |
| 값 | 0 | 0 | 0 | 1 | **0** | 0 | 1 | 0 | **0** | 0 | **1** | **1** |
| 배치 | D8 | D7 | D6 | D5 | **P8** | D4 | D3 | D2 | **P4** | D1 | **P2** | **P1** |

C1: 비트 번호 1, 3, 5, 7, 9, 11의 짝수 패리티 정상 여부 → 정상 (0)

C2: 비트 번호 2, 3, 6, 7, 10, 11의 짝수 패리티 정상 여부 → 정상 (0)

C4: 비트 번호 4, 5, 6, 7, 12의 짝수 패리티 정상 여부 → 비정상 (1)

C8: 비트 번호 8, 9, 10, 11, 12의 짝수 패리티 정상 여부 → 비정상 (1)

## 오류 비트를 찾아내기 위한 신드롬 단어(syndrome word)를 구하라.

C8 C4 C2 C1: 1100(2) = 12(10) 임으로 12번째 비트를 반전하면 오류를 정정할 수 있다.