R E P O R T

컴퓨터공학입문 이론 과제6

과목명	컴퓨터공학입문
분반	4 분반
교수	윤 한 경
학번	2020136129
이름	최 수 연
제출일	2020년 5월 7일 목요일

목차

문제 1. n의 보수에 의한 감산에서 올림수를 어떻게 처리하는지 설명하고, 10진수로 피감수가 13 감수가 10인 감산을 10의 보수에 의한 방법으로 처리하시오.

n의 보수에 의한 감산에서는 올림수가 있는 경우에는 올림수를 버리고, 올림수가 없는 경우에는 피감수 M에 감수 N의 n의 보수(2의 보수 또는 10의 보수)를 더하여 나온 결과 값의 n의 보수를 구하고 -를 붙인다.

피감수가 13이고 감수가 10인 감산을 10의 보수에 의한 방법으로 처리하려면, 먼저 피감수 13에 10의 10의 보수인 89(∵99-10)+1=90을 더하면 103이 나오고 여기서 올림수가 존재하므로 올림수인 1을 버린다.

문제 2. 2진수 1011을 그레이 코드로 표시하시오.

2진 코드를 그레이 코드로 변환하는 방법은 첫 번째, 2진수의 최상위(MSB)는 그대로 그레이 코드의 최상위 비트가 되고, 두 번째, 2진수의 이웃한 두 비트를 자리 올림수를 제거한 나머지를 첫 번째 그레이 코드로 취한다. 그리고 두 번째 과정을 반복한다. 이 과정에 따라 2진수 1011을 그레이 코드로 변환하면 1110이 된다.

문제 3. 6비트 BCD 코드를 설명하고 6비트 BCD 코드에서 알파벳 소문자가 없는 이유를 설명하시오.

6비트 BCD 코드는 4비트 BCD 코드에 2개의 비트를 더하여 10진법의 수 이외에 문자를 표시할 수 있도록 한 것으로 기억장치의 단어의 길이가 6의 정수배인 컴퓨터에 적합한 코드이다. 6비트 BCD 코드는 알파벳 소문자를 포함할 시 6비트를 초과하기 때문에, 영문의 소문자 표시는 대응하는 코드가 없고, 또한 대문자와 소문자를 구별할 수 없기 때문에 알파벳 소문자를 표현할 수 없다.

문제 4. 해밍코드를 설명하고, 수신 비트가 1110110011111일 때 몇 번째 비트에서 에러가 발생했는지와 교정된 2진 정보 데이터(D8~ D1)를 적으시오.

해밍코드는 에러 정정 코드 중 가장 간단한 형태로 에러를 검출하고 자동으로 정정해주는 코드이다. 정보 데이터 비트 수가 D일 때 필요한 패리티 비트 수 P는 다음 조건을 만족해야 한다. $2^P >= D + P + 1$

에러 체크 비트 0110을 10진수로 바꾸면 6이 된다. 이 때, 6번째 비트에서 1이 0으로 바뀌는 에러가 발생했다는 것을 알 수 있다. 따라서 6번째 데이터 비트 0을 1로 바꾸고, 정정 후의 2진 정보 데이터 비트를 적으면 D8~D1 = 11101101이다.

문제 5. 한글코드를 종류별로 설명하고 한글코드를 8비트로 처리할 수 있는 방안이 있는지 의견을 개진하시오.

한글코드에는 완성형 코드, 2바이트 조합형 코드, 유니코드가 있다.

먼저 완성형 코드는 음절 한 문자마다 한 개의 코드를 부여하는 방법으로, 2바이트에 글자별로 코드를 부여하는 방법이다. 이 코드는 초성, 중성, 종성으로 나뉘지 않고 글자 단위로 처리하기 때문에 조합형에 비해 입출력 처리가 단순하다. 또한 한글을 나타내는 코드의 첫 번째와 두 번째 바이트의 상위 비트가 모두 1로 되어있다.

2바이트 조합형 코드는 2바이트의 연속된 16비트에 초성, 중성, 종성을 각각 5비트씩 할당하여 이를 자모의 조합에 의해 한글 문자를 표시할 수 있게 규정한 한글코드이다. 처음 한 바이트의 상위 비트에 1이 세트되면 한글코드로 인식하여 다음 연속으로 오는 한 바이트를 처음과 연결된 하나의 코드로 구성한다. 2바이트 조합형은 모든 형태의 한글 처리가 가능하며, 초성, 중성, 종성의 분리가 쉽기 때문에 음성 인식 등의 형태소로 분석할 수 있고, 여러 분야에서의 활용이 용이하다. 또한 한글 데이터를 간단한 루틴으로 정렬할 수 있다.

유니코드는 기존의 ASCII 8비트 체계를 16비트 코드체계로 확장 전환시킴으로써 전 세계의 문자 체계를 수용할 수 있게 하는 컴퓨터 표준을 마련한 코드이다. 이 코드는 UNIX, Linux, NT 이상의 윈도우 시스템에서 사용하고 있으며, Java에서도 사용되고 있다. 유니코드 2.0은 조합할 수 있는 모든 완성형 한글 글자 11,1172자가 가나다순으로 정렬되어있고, 조합형 한글도 모두 갖고 있어, 완성형뿐만 아니라 조합형의 형태도 모두 갖고 있다.

한글코드를 8비트로 처리하는 방안이라고 한다면, 초성과 중성을 합치고 종성을 따로 추가하는 방식으로 처리하면 중성에 따라 종성을 붙이지 않는 경우도 있기 때문에, 초성, 중성, 종성을 따로 분리하여 처리하는 것과 글자별로 코드를 부여하는 것보다 좀 더 적은 저장 공간을 사용할 수 있지 않을까 생각한다.