

컴퓨터 구조_정리노트#2
ICT융합공학부 202204010 공성택

4주차 2장

Q) '1의 보수'보다 '2의 보수'가 부호를 나타내는 데에 더 표준으로 사용된다고 배웠는데, 그렇다면 '1의 부호'가 사용되는 경우는 무엇이 있을까?

A) 조사한 결과, 일부 네트워크 프로토콜에서 데이터의 오류를 검출하는 데 1의 보수를 사용한다고 한다. 예시 중 하나로는 인터넷 제어 메시지 프로토콜의 체크섬에서 1의 보수가 사용된다. 1의 보수로 변환하여 각 바이트를 합산한 후 값을 계산해 데이터의 무결성 검사를 한다고 한다. 또한, 디지털 신호 처리를 하는 경우에도 마찬가지로 신호의 무결성을 확인하기 위해 1의 보수를 사용한다고 한다.

조사 결과, 아무래도 2의 보수가 1의 보수보다 오류를 더 효과적으로 처리하고 부호화된 수의 연산을 효율적으로 처리할 수 있기 때문에, 1의 보수는 무결성 검사 등의 처리에서만 사용하고 다른 경우는 거의 2의 보수를 사용한다고 결론내렸다.

Q) 아스키코드와 유니코드의 자세한 차이점과 각각 어떻게 사용되고 있는 지 궁금하다.

A) 아스키코드와 유니코드의 가장 큰 차이는 다국어의 지원이다. 아스키코드는 영어 알파벳과 숫자, 특수 문자 등만 포함되어 있지만, 유니코드는 숫자와 알파벳은 물론이고, 전 세계의 다양한 언어와 기호, 심지어 이모티콘 등도 포함하고 있다. 이런 점 때문에, 유니코드가 아스키코드보다 문자의 범위가 넓다.

아스키코드는 주로 영어 기반의 소프트웨어, 시스템에서 사용되고, 유니코드는 전 세계적으로 사용되어 웹 페이지, SNS, 소프트웨어, 게임 등 다양한 분야에서 사용된다. 추가적으로 유니코드가 아스키코드보다 훨씬 지원하는 범위가 넓는데 아스키코드가 사용되는 이유는 주로 영어로 작성되는 프로그램 환경에서는 크기가 작고 간결한 아스키코드가 메모리나 통신 프로토콜에서 자원을 적게 사용하고 빠르게 처리되기 때문이다. 또한, 아스키코드는 각 문자에 대해 고정된 크기의 코드를 사용해 처리 및 구현이 간단하기 때문에 제한된 환경, 특히 임베디드 시스템 등에서 유용하게 사용된다.

5주차 4장

Q) 컴퓨터에서 부스 알고리즘을 사용하는 구체적인 이유는?

A) 컴퓨터가 부스 알고리즘을 사용하는 가장 큰 이유는 속도 때문이다. 부스 알고리즘을 사용하면 일반적인 곱셈 알고리즘보다 연산 속도를 빠르게 할 수 있다. 특히 큰 숫자의 곱연산에서 두드러지게 나타난다. 속도 문제 이외에도, 부스 알고리즘은 곱셈 연산을 덧셈과 시프트 연산으로 대체하기 때문에, 전력과 속도를 절약할 수 있고, 컴퓨터의 이진수 체계와 잘 맞아서 컴퓨터에서 구현하기가 쉽다. 또한, 부호 비트를 활용하여 오버플로우를 관리하기 쉽기 때문에 곱연산에서 정확도를 유지하기 유용하다고 조사했다.

Q) 강의 시간에 CPU 프로세서의 기본 구조에 대해 배웠는데, 시중의 CPU에는 내장 그래픽이 달린 경우가 있다. CPU의 내장 그래픽에 대해 추가로 어떻게 동작하는지 궁금하다.

A) CPU에 내장 그래픽이 있는 경우, 내장 그래픽은 CPU의 전력 및 자원을 공유한다. 내장 그래픽은 시스템 메모리를 공유해 그래픽 데이터를 처리하고 화면에 출력한다. 이때, 내장 그래픽은 CPU 내부의 다른 프로세서들과 상호작용한다. 내장 그래픽은 그래픽 데이터를 연

산하는 과정에서 ALU가 연산을 수행하는데에 도움을 주고, 제어 유닛은 내장 그래픽의 명령어를 해석하고, ALU나 다른 프로세서들과 협력하여 그래픽 작업을 제어하고 실행한다. 이처럼 내장 그래픽은 CPU 프로세서들과 상호작용하여 그래픽 데이터 처리를 담당한다.

2의 거듭제곱 테스트

01. 진법과 진법 변환



2 진법

과제는 반드시 p.14로 제출

정리노트

$2^0=1$ $2^5=32$
 $2^1=2$ $2^6=64$
 $2^2=4$ $2^7=128$
 $2^3=8$ $2^8=256$
 $2^4=16$ $2^9=512$
 $2^{10}=1024$

❖ 10진법

- 10진수: 기수가 10인 수
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9의 10개 수로 표현

$$\begin{aligned}
 9345.35 &= 9 \times 1000 + 3 \times 100 + 4 \times 10 + 5 \times 1 + 3 \times 0.1 + 5 \times 0.01 \\
 &= 9 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 5 \times 10^{-2}
 \end{aligned}$$

❖ 2진법

- 기수가 2인 수
- 0, 1 두 개의 수로 표현

$$\begin{aligned}
 1010.1011_{(2)} &= 1 \times 1000_{(2)} + 0 \times 100_{(2)} + 1 \times 10_{(2)} + 0 \times 1_{(2)} \\
 &\quad + 1 \times 0.1_{(2)} + 0 \times 0.01_{(2)} + 1 \times 0.001_{(2)} + 1 \times 0.0001_{(2)} \\
 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}
 \end{aligned}$$

2진 정수 연산_ 58 -49 계산해보기

58 - 49 계산

58: 00111010

49: 00110001

49의 2의 보수: 11001110 + 1
= 11001111

2의 원형수

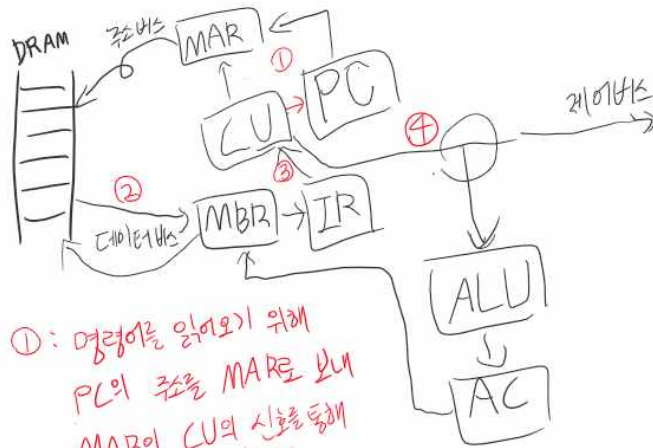
$$\begin{array}{r}
 00111010 \quad \leftarrow \text{양수} \\
 + 11001111 \quad \leftarrow \text{음수} \\
 \hline
 100001001
 \end{array}$$

2의 보수

$2^0 + 2^3 = 1 + 8 = 9$

$58 - 49 = 9$

그림 4-3 CPU 프로세스 기본 구조에서 명령 실행 순서 그림 그리고 설명해보기



- ① : 명령어를 읽어오기 위해 PC의 주소를 MAR로 보내 MAR이 CU의 신호를 통해 메모리 주소를 읽는다.
 - ② 메모리에서 데이터버스를 통해 MBR로 데이터를 가져온다. 그리고 PC를 1 증가한다.
 - ③ 가져온 명령어를 IR로 해석하고, 제어 장치를 통해 유형을 결정한다.
 - ④ 명령을 실행하여 제어부로 보내 컴퓨터 장치에 제어신호를 보내거나 명령을 실행하여 ALU로 산술+논리 연산하고 그 연산을 AC에 저장해서 MBR을 통해 주기의 장치로 전송한다.
- ★ 이 과정을 반복한다.

부스 알고리즘 $5 \times (-4)$ 직접 계산 해보기

$$\begin{aligned}
 5 \times (-4) \\
 M = 5 = 0101 \\
 Q = -4 = 4\text{의 2의 보수} \\
 = 1011 + 1 \\
 = 1100 \\
 -M = 1010 + 1 \\
 = 1011
 \end{aligned}$$

n	A	Q	q ₀	설명
4	0000	1100	0	초기 00이므로 right shift 오른쪽 시프트
3	0000	0110	0	00이므로 right shift 오른쪽 시프트
2	0000	0011	0	00이므로 A = A - M = A + (-M) 오른쪽 시프트
1	1011	0011	0	A = A - M 오른쪽 시프트
0	1101	1001	1	1이므로 right shift 오른쪽 시프트
0	1110	1100	1	오른쪽 시프트

$$\begin{aligned}
 \therefore & 1110 \ 1100 \text{ (음수)} \\
 & = 0001 \ 0011 + 1 \text{ (2의 보수)} \text{ (정확값 구함)} \\
 & = 0001 \ 0100 = 2^2 + 2^4 = 4 + 16 = 20 \\
 \therefore & -20 = 5 \times (-4)
 \end{aligned}$$