시스템 프로그래밍 (1.1-1.3 요약)

12161633 이영주

시스템 소프트웨어는 컴퓨터의 작동을 제공하는 다양한 프로그램들로 구성되어 있다. 사용자와 컴퓨터의 운영 시스템고 상호작용을 함으로써 어떻게 작동하는 지에 대한 세부 사항을 사용자가 알 필요 없이 응용 프로그램 또는 다른 문제의 해결에 초점을 맞출 수 있도록 도와준다. 책에서 주요하게 다루는 주제는 시스템 소프트웨어와 기계의 구조 사이에서의 관계이다.

대부분의 시스템 소프트웨어가 응용 소프트웨어와 다른 점은 기계 종속성이다. 응용 소프트웨어는 컴퓨터를 도구로 하여, 주로 문제의 해결에 관심이 있으며, 컴퓨터 시스템 자체에는 관심이 없다. 반면, 시스템 프로그램은 어떤 특별한 응용보다는 컴퓨터 자체의 운영 및 사용을 위한 지원을 목표로 한다. 이러한 이유로 시스템 프로그램은 보통 사용되는 기계의 구조에 연관되어 있다. 대부분의 시스템 소프트웨어는 기계 종속적이다. 가장 보편적인 하드웨어 특징과 개념을 보여줄 수 있도록 일반적이 아닌 기능은 포함하지 않은 단순 명령어 컴퓨터(Simplified Instructional Computer : SIC)구조를 사용하였다. 이는 소프트웨어 시스템의 각 요소에 대한 핵심 개념이 어떤 특정한 기계에서만 나타나는 기능으로부터 명확하게 분리될 수 있다.

SIC는 전형적인 마이크로 컴퓨터와 유사하다. 대부분이 그러하듯 표준모델과 확장모델인 XE 두개 버전이 존재한다. 두 버전은 표준 SIC 기계의 모든 프로그램이 SIC/XE 시스템에서 용이하게 수행될 수 있는 상향 호환성을 유지하도록 설계되어 있다.

SIC 기계 구조를 살펴보자. 우선, 메모리는 한 바이트 당 8비트로 구성되어 있다. 한 워드는 세개의 연속적인 바이트로 구성된다. SIC의 바이트 주소 워드는 최하위 바이트의 위치로서 주소 지정이 된다. 컴퓨터 메모리는 전체 2^15바이트이다. 레지스터는 특별한 목적을 갖는 5개로 구성된다. 각 레지스터는 24비트의 길이를 가진다. 번호 0인 A는 산술 연산에 사용, 번호 1 – X는 인덱스 레지스터로 주소 지정에 사용된다. L은 번호 2로 링크 레지스터를 뜻하며, 점프 명령어 수행 시 귀환 주소를 저장한다. PC는 번호 8로 다음 명령어의 주소를 가지고 있는다. 마지막으로 번호 9인 SW는 조건 코드를 포함하여 다양한 정보를 갖고 있는다. 데이터는 정수일 때 24비트의 2진수로 저장된다. 음수는 2의 보스를 사용하여 나타낸다. 문자는 8비트 ASCII코드를 사용하여 저장한다. SIC 표준형에는 부동 소수점 하드웨어가 없다. SIC의 표준형의 모든 기계 명령어는 opcode(8) + x(1) + address(15) 의 24비트 형식을 가진다. 플래그 비트 x는 인덱스 주소 지정 방식을 나타내기 위해 사용된다. x비트는 2가지 주소 지정 방식을 나타낼 수 있다. 목표 주소 TA를 계산하여 주소를 지정한다. x=0일 때는 직접 주소 지정 방식으로 TA = 주소 값을 그대로 갖는다. x=1일 때는 인덱스 주소 지정 방식으로 TA = 주소 + (레지스터/메모리의 내용)으로 표현한다. SIC는 대부분의 간단한 작업을 할 수 있는 기본적인 명령어 집합을 제공한다. 레지스터로부터 읽고 쓰는 명령어 뿐만 아니라, 정수 연산 명령어들이 포함되어 있다. SIC의 표준형에서, 입력과 출력은 한 번에 한 바이트씩 레지스터 A의 제일 오른쪽 8비트로 전송하거나, 혹은 오른쪽 8비트로부터 전송받음으로써 수행된다. 각 장치는 유일한 8비트 코드로 지정된다. 세 개의 입출력 명령어가 있는데 각각은 피연산자로서 장치 코드를 지정한다. TD명령어는 주소가 지정된 장치가 데이터의 한 바이트를 받거나 보낼 준비가 되어 있는지 테스트한다. 그 후, RD나 WD로 읽거나 쓰는 것을 실행한다.

확장모델 SIC/XE는 표준과는 약간 다른 구조를 가지고 있다. 메모리 구조는 표준형과 같으나 최대로 사용 가능한 메모리는 1메가 바이트이다. 이런 메모리의 증가는 주소 지정 방식에 변화를 가져온다. 부가적인 레지스터가 존재한다. B는 베이스 레지스터로 주소 지정에 사용한다. 번호는 3이다. 번호 4인 S와 5인 T는 범용 레지스터로 특수 용도가 없다. 번호 6인 F는 부동 소수점 누산기로 사용된다. 데이터는 표준형과 같은 데이터 형식을 제공하나 추가적으로 S(1) + 지수(11) + 소수(36)의 구조인 48비트 부동 소수점 데이터 형태가 존재한다. S로 부호를 결정한다. SIC/XE의 큰 메모리 용량은 주소가 15비트 필드에 더 이상 맞지 않으므로 표준형에서 사용되었던 명령어 형식은 적합하지 않다. 두 가지로 나뉘어 선택할 수 있는데 하나는 상대 주소 지정 방식을 사용하는 방법이고 나머지는 주소 필드를 20비트로 확장하는 방법이다. 새로운 명령어 형식은 4 종류이다. op코드만을 나타내는 형식1, op(8) + r1(4) + r2(4)로 나타내는 형식2와 op(6)과 n,i,x,b,p,e와 주소 필드로 이루어진 형식 3과 4가 존재한다. 형식 3과 4는 비트 e를 통해 구별된다. e가 0이면 3, 1이면 4이다. 두 가지의 새로운 상대 주소 지정 방식이 형식 3으로 어셈블되는 명령어들을 위해 사용된다. b=1,p=0인 베이스 상대 주소 지정에서 형식 3 명령어의 변위 필드는 12비트 무부호 정수로, b=0,p=1인 PC 상대 주소 지정에서 변위 필드는 2의 보수로 표현된 12비트의 부호정수로 해석된다. b와 p 모두 0이면 형식 3의 명령어의 변위 필드가 목표 주소로 사용된다. 형식 4 명령어의 비트 b와 p는 모두 0이 되어야 하고, 목표 주소는 주소 필드에서 얻어진다. 이들 주소 지정 방식들은 인덱스 주소 지정 방식과 같이 사용될 수 있다. 즉, 비트 x가 1이면 (x)항이 목표 주소 계산에 더해진다. 형식 3와 4의 비트 i와 n은 목표 주소가 어떻게 사용되는가를 정하는데 사용된다. i=1이고 n=0이면 목표 주소 자체가 피연산자 값으로 쓰인다. 이것은 즉시 주소 지정 방식이라고도 한다. i=0이고 n=1이면, 목표 주소에 의해 지정된 위치의 워드를 가져오며, 이 워드에 있는 값이 피연산자 값의 주소로 사용된다. 이것을 간접 주소 지정 방식이라고 한다. 비트 i와 n이 모두 0이거나 1이면 목표 주소는 피연산자의 위치로서 주어진다. 이 것을 단순 주소 지정 방식이라고 한다. 인덱스는 즉시나 간접 주소 지정 방식과는 함께 쓰일 수 없다. 비트 n과 i가 모두 0이라면 비트 b, p, e는 주소 지정 방식을 나타내는 플래그 비트가 아닌 명령어의 주소 필드의 부분으로 인식된다. 이것은 형식 3 명령어를 표준형의 명령어 형식과 동일하게 하며, 원하는 호환성을 지원하게 하는 것이다. SIC/XE는 표준형에서 사용하는 명령어들을 모두 지원한다. 이에 추가하여 새로운 레지스터에 로드 및 저장되는 명령들과 부동 소수점 산술연산을 수행하는 명령들이 있다. 레지스터로부터 그들의 피연산자를 가져오는 명령어와 감독자 호출 명령어도 제공된다. 감독자 호출 명령어는 OS와의 통신에 사용되는 interrupt를 발생시킨다. SIC/XE는 CPU가 다른 명령을 수행하는 동안 입력과 출력을 수행할 수 있는 입출력 채널을 추가하여 계산과 입출력을 동시에 가능하게 하여 더 효과적인 시스템 운영이 되도록 도와준다.