

5주 1강

CPU와 마이크로프로세서 그리고 조직



CPU와 마이크로프로세서의 이해



1 마이크로프로세서(microprocessor)

- 컴퓨터에서 명령을 수행하고 데이터를 처리하는 중앙처리장치(CPU, Central Processing Unit)를 집적회로의 칩 형태로 만든

2 폰 노이만(von Neumann) 컴퓨터 구조

- 프로그램내장 방식
- 데이터와 명령어가 주기억장치에 저장되어 있다가 CPU로 전달
- CPU는 전달된 명령어에 따라 데이터를 원하는 형태로 처리하고
처리결과는 다시 데이터버스를 통해서
주기억장치 RAM으로 보내지게 된다

3 주회로기판(main board)에서 CPU와 주기억장치의 위치



머신 사이클 (machine cycle)

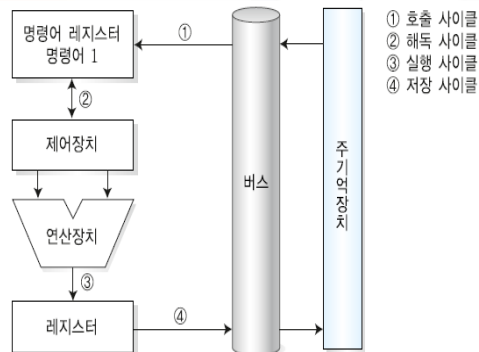
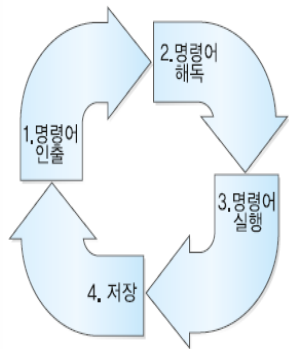


1 프로그램을 구성하는 명령어는 4단계의 과정을 통해서 수행.

이 과정은 CPU에서 동작을 하므로 머신 사이클이라고 한다.

2 각 단계별 사이클의 역할

- 인출(Fetch) 사이클 : 필요한 명령어를 주기억장치에서 불러오는 사이클
- 해독(Decode) 사이클 : 호출된 명령어를 해석하는 사이클
- 실행(Execute) 사이클 : 해석된 명령어를 산술논리연산장치를 통하여서 실행
- 저장(Store) 사이클 : 수행결과를 주기억장치에 저장하는 사이클



클럭 속도와 명령어 처리속도



1 클럭(Clock) 주파수

- CPU가 일정 속도로 동작하기 위해 공급되는 전기적 진동(pulse)
- 1초 동안에 발생한 클럭 수를 클럭 주파수라 한다. 단위는 Hz
 - 1Hz : 1초 동안에 클럭이 1번 발생
 - MHz : 1초 동안에 10^6 개의 클럭 발생할 때의 단위
 - GHz : 1초 동안에 10^9 개의 클럭이 발생할 때의 단위
 - 75MHz : 초당 7천 5백만 번의 사이클로 0과 1의 디지털 신호를 발생

2 주회로기판에 장착되어 있는 클럭 발생기에서 클럭을 생성, 클럭 수가 높을수록 컴퓨터의 처리 속도가 빠르다는 것을 의미

3 명령어 처리 속도

- 초당 처리하는 명령어의 개수
- 단위 : MIPS(Million Instruction Per Second)는 1초에 100만 개의 명령어 수행
- 처리속도가 18.5MIPS라면 1초 동안에 1,850만 개의 명령을 실행

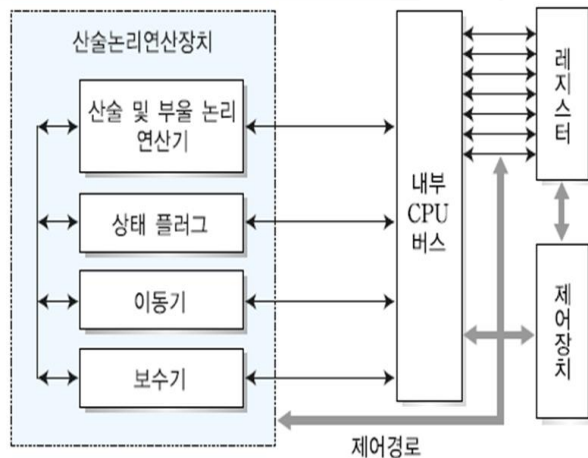
CPU의 조직



1 CPU 내부는 기본적으로 연산장치, 제어장치, 레지스터의 집합으로 구성되며, 이것들은 내부 CPU 버스로 연결되어 있다.

- 연산장치에서 각종 연산기능을 수행한다.
- 레지스터에서 데이터를 보관하는 기억기능을 수행한다.
- 제어장치는 명령을 해독하고 제어신호를 발생하여 제어기능을 수행한다.
- 버스를 통해서 데이터의 전달기능을 수행한다.
- 결과적으로 CPU는 연산, 기억, 제어, 전달 등 네 가지 기본 기능을 수행

2 CPU의 내부구조



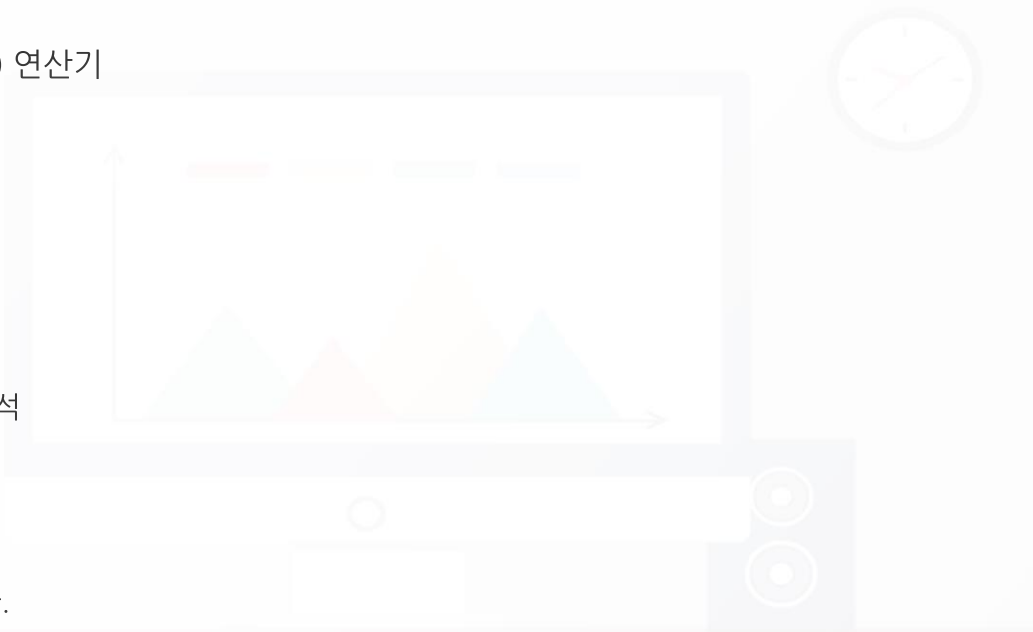


1 산술 논리 연산장치(ALU, Arithmetic and Logic Unit)

- 산술연산과 AND, OR, XOR 등의 논리연산을 계산하는 디지털 회로

2 ALU의 구성

- 산술 및 부울 논리(Arithmetic and Boolean Logic) 연산기
 - 실제적인 산술 연산과 논리 연산을 수행하는 회로다.
- 상태 플래그(Status Flags)
 - 연산 중인 ALU 내의 데이터 상태를 표시
 - 음수, 0, 오버플로우 등을 표시한다.
- 이동기(Shifter)
 - 데이터 비트를 좌우로 비트 별로 이동
 - 비트의 이동은 2로 곱셈하거나 나눗셈하는 것으로 해석
- 보수기(complementer)
 - ALU내의 데이터에 대하여 보수연산을 수행한다.
 - 컴퓨터에서는 2의 보수를 주로 사용한다.
 - 2의 보수는 덧셈과 뺄셈 계산장치의 제작을 쉽게 한다.





1 CPU내에서 데이터를 저장하는 장치, 간략하게 레지스터라고도 한다.

2 컴퓨터의 기억장치들 중에서 속도가 가장 빠르다

- ALU에서 처리된 결과 데이터를 임시적으로 보관하는 역할
- 주기억장치로부터 읽어온 명령어와 데이터를 임시적으로 보관

3 레지스터의 용도별 분류

- 데이터 레지스터 : 정수 데이터 값을 저장할 수 있는 레지스터다.
- 주소 레지스터 : 기억장치 주소를 저장하여 기억장치 액세스에 사용
- 범용 레지스터 : 데이터와 주소를 모두 저장할 수 있는 레지스터다.
- 부동 소수점 레지스터 : 부동소수점 데이터 값을 저장하기 위해 사용
- 상수 레지스터 : 0이나 1 등 고정된 데이터 값을 저장하기 위한 레지스터다.
- 특수 레지스터 : 실행 중인 프로그램의 상태를 저장하는 레지스터
(프로그램 카운터, 상태 레지스터)
- 명령 레지스터 : 현재 실행 중인 명령어를 저장한다.
- 색인 레지스터 : 실행 중에 피연산자의 주소를 계산하는데 사용된다.

사용자에 보이는 CPU 레지스터



1 어셈블리 프로그래머는 프로그램에서 사용되는 변수 데이터 등의 저장을 위해 해당 레지스터를 알고 있어야 한다.

2 사용하는 목적에 따른 분류

- 일반목적용 레지스터 : 프로그래머에 의해 여러 용도로 사용
- 데이터 레지스터 : 데이터 저장에만 사용할 수 있는 레지스터(누산기)
- 주소 레지스터 : 특정 주소지정 방식을 위해 사용하는 레지스터
 - 스택 포인터(stack pointer) : 스택이라는 저장장치의 최상위(top of stack) 주소를 저장하는 레지스터다.
- 조건 코드(Condition Codes) : 저장된 데이터의 상태를 표시하는데 사용된다.
 - 부호(sign) 비트 : 경우에는 양수인지 음수인지를 표시한다.
 - 영(0) 비트 : 해당 데이터가 0이라는 것을 표시한다.
 - 오버플로우 비트 : 연산의 결과 등에 오버플로우가 발생했다는 것을 표시한다.

CPU의 제어 및 상태 레지스터



● 프로그램 카운터(PC, Program Counter)

- 주기억장치에 저장된 다음에 인출할 명령어의 주소를 가지고 있는 레지스터다.

● 명령어 레지스터(IR, Instruction Register)

- 가장 최근에 주기억장치인 RAM에서 인출한 명령어를 저장

● 기억장치 주소 레지스터(MAR, Memory Address Register)

- 액세스할 기억장치의 주소가 저장되는 레지스터다. 이 레지스터의 출력이 주소 버스와 직접 연결된다.

● 기억장치 버퍼 레지스터(MBR, Memory Buffer Register)

- 기억장치에 쓰여질 데이터 혹은 가장 최근에 읽은 데이터가 저장

● 입/출력 주소 레지스터(I/O AR: I/O Address Register)

- 입/출력장치의 주소를 저장하는 주소 레지스터다.

● 입/출력 버퍼 레지스터(I/O BR: I/O Buffer Register)

- 입/출력 모듈과 CPU 사이에 교환되는 데이터를 일시적으로 저장

스택 (Stack) 저장장치



1 스택의 특징

- CPU 내부의 레지스터 집합에 존재하는 저장장치
- 스택의 길이는 가변적
- 한 번에 하나의 요소에만 액세스 가능
- 나중 입력 먼저 출력(LIFO, Last-In-First-Out)의 특징

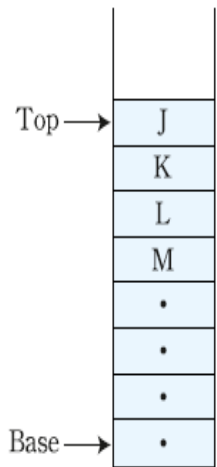
2 스택의 동작 표현

- TOP : 데이터가 입력되고 출력되는 액세스 부분
- PUSH : 스택의 Top에 새로운 요소를 추가 저장하는 동작
- POP : 스택의 Top에서 하나의 요소를 꺼내는 동작
- 스택 포인터 : Top의 위치를 표시하는 장치

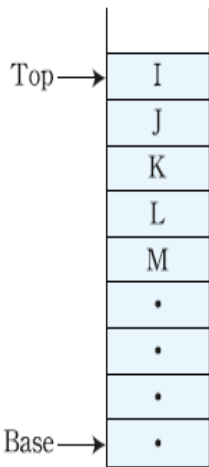
3 스택의 동작 과정

- Top에 위치한 요소는 POP동작을 통해서 스택에서 인출
- 연산을 수행하고, 연산의 결과를 다시 스택에 저장되도록 PUSH
- 결과 데이터가 저장되었으므로 Top은 위쪽으로 이동하게 된다.

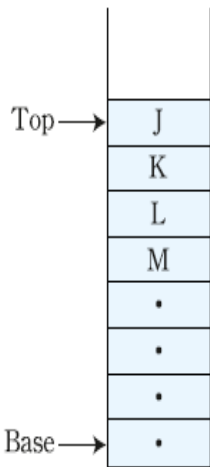
스택의 기본적인 동작



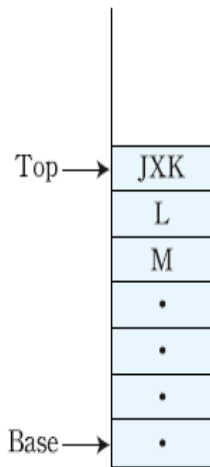
(a) 시작



(b) I를 PUSH한 후
스택의 상태



(c) I를 POP



(d) J와 K를 곱셈한 후
스택의 상태

다음 시간

5주 2강. CPU의 조직과 CPU 논리회로

