6주차 1차시 CPU와 Micro Prossor

【학습목표】

1. 인텔 CPU의 역사를 시리즈별로 설명할 수 있다.
2. CPU의 내부구조를 설명할 수 있다.

학습내용1 : CPU의 역사

<PC CPU의 양대산맥 intel 과 AMD>

1. 인텔 CPU 역사

* 인텔 초창기 CPU 1971년 노이스와 무어가 공동으로 설립한 인텔은 최초 상업용 CPU인 4004을 출시
- 4004는 1만 나노(nm)공정으로 만들어졌음
- 1971년 인텔 4004
- 1972년 인텔 8008
- 1974년 인텔 8080
- 1979년 인텔 8086
- 1979년 인텔 8088

* x86 시리즈 8088과 같은 이름보다는, 286, 386, 486과 같은 이름부터는 많은 분들에게 제법 친숙할 것
- x86이란 명칭은 8086부터 시작되었음
- 원래는 80286, 80386, 80486이 정식 명칭이나 편의상 앞에 80을 빼고 뒷 3자리만 부르게 되었다고 함
- 1982년 인텔 80286
- 1985년 인텔 80386
- 1989년 인텔 80486

* 펜티엄(Pentium)
- 시리즈1993년 인텔을 대표하는 이름이라고 해도 과언이 아닌 펜티엄이 탄생
- 486 다음 모델이니 586으로 나와야 했지만, 마침 AMD에서도 586이라는 이름을 사용하고 있어서 '다섯 번째'라는 접미사 Pent- 와 '부품'을 의미하는 '-ium'를 합쳐 Pentium으로 정했다고 함

* 펜티엄(Pentium) 시리즈
- 1995년 인텔 펜티엄 프로
- 1997년 인텔 펜티엄 2
- 1999년 인텔 펜티엄 3
- 2000년 인텔 펜티엄 4
- 2005년 인텔 펜티엄 D
- 펜티엄 4 부터 코드명을 사용하였으며 윌라멧이 최초의 코드명이며 이후로 노스우드, 프레스캇, 시더밀 등
- 흔히 펜티엄 4 하면 떠올리는 제품이 바로 노스우드와 프레스캇 제품이며 프레스캇 같은 경우는 발열이 워낙 심해 '프레스핫' 이라고 불리기도 했음

* 2006년 부터 인텔은 '펜티엄' 이란 명칭 대신에 코어 2 듀오 라는 명칭으로 바꾸고 본격적인 듀얼코어 시대
- 이전까지는 클럭을 지속적으로 끌어 올리면서 단일 코어의 성능을 밀고 나갔지만 이때 부터는 클럭속도 증가가 벽에 부딪혀 대신 두 개 이상의 복수 코어를 탑재한 CPU를 만들기 시작
- 2006년 코어 2 듀오 콘로 E6000 시리즈
- 2007년 코어 2 듀오 앨런데일 E4000 시리즈
- 2007년 코어 2 쿼드 켄츠필드 Q6000 시리즈
- 2007년 펜티엄 콘로 E2000 시리즈
- 2007년 셀러론 콘로 200/400 시리즈
- 2007년 셀러론 앨런데일 E1000 시리즈
- 듀얼 코어를 선보인 이듬해 쿼드코어 (4개의 코어)를 가진 켄츠필드도 출시
- 2000년 초반 잠시 AMD에게 밀리는 듯 했던 인텔의 시작됨
- 펜티엄 콘로와 셀러론 시리즈는 기존 코어 2 듀오에서 L2 캐쉬가 제거 되거나 단일 코어로 만들어 지는 등 성능이 더 낮은 버전임

* 펜린(Penryn)계열 데스크탑용 코어2 시리즈의 코드명(45nm공정)
- 65나노 공정에서 45 나노 공정으로 바뀌면서 더 낮은 전력과 발열에 더 높은 성능을 내는 CPU를 설계할 수 있었음
- 이때부터 지금 까지도 인텔은 AMD보다 거의 반년가까이 앞서서 더 낮은 공정에서 CPU를 설계해 올 수 있었고 이는 항상 인텔이 AMD 보다 우수하다라는 인식을 소비자들에게 심어주기에 충분했음
- 2008년 코어 2 듀오 울프데일 E7000/8000시리즈
- 2008년 코어 2 쿼드 요크필드 Q8000시리즈
- 2008년 펜티엄 울프데일 E5000/6000 시리즈
- 2008년 셀러론 울프데일 E3000시리즈

* 네할렘(Nehalem)계열 데스크탑용 코어 i 시리즈의 코드명 (45nm공정)
- 네할렘 아키텍쳐는 기존 코어 2 듀오와 다른 다양한 특징들을 갖고 있음
- 쿼드 코어를 대중화 시켰고 L3캐쉬 추가 및 노스브릿지 내장으로 더욱 빠른 성능 향상을 가져 왔음
- 그리고 '터보 부스트'라는 기술을 통해 4개의 코어 중 작동하지 않는 코어에 들어갈 전력을 작동하는 코어에 밀어줌으로써 더욱 빠르게 동작하게끔 만들었음

- 2008년 코어 i7 블룸필드 900 시리즈
- 2009년 코어 i7 린필드800시리즈
- 2009년 코어 i5 린필드700시리즈
- 특히 린필드는 국민CPU로 칭송을 받으며 본격적으로 쿼드 코어 대중화를 이끌었고 개인 사용자, PC방, 사무실 할거 없이 매우 넓게 퍼졌음
- 2010년 인텔이 사상 최대의 실적을 낼 수 있었던 것도 바로 이 린필드 i5 750/760이 있었기에 가능

* 웨스트미어(Westmere) 계열 데스크탑용 코어 i 시리즈의 코드명 (32nm공정)
- 네할렘 아키텍쳐는 기존 코어 2 듀오와 다른 다양한 특징들을 갖고 있음
- 쿼드 코어를 대중화 시켰고 L3캐쉬 추가 및 노스브릿지 내장으로 더욱 빠른 성능 향상을 가져 왔음
- 그리고 '터보 부스트'라는 기술을 통해 4개의 코어 중 작동하지 않는 코어에 들어갈 전력을 작동하는 코어에 밀어줌으로써 더욱 빠르게 동작하게끔 만들었음
- 2010년 코어 i5 클락데일 600시리즈
- 2010년 코어 i3 클락데일 500 시리즈
- 2010년 펜티엄 클락데일 G6950
- 2010년 셀러론 클락데일 G1101
- 2010년 코어 i7 걸프타운 980X 익스트림에디션 / i7 970
- 코어 i7 걸프타운의 경우에는 CPU 최초로 헥사코어 (6개의 코어)를 탑재해 출시 되었음

* 샌디브릿지 (Sandy Bridge) 계열 코어 i 시리즈
- 2011년 샌디브릿지가 출시
- 전력 소비는 웨스트미어 제품들과 비슷하나 성능은 더욱 뛰어나고 특히 4Ghz 까지 아무런 추가 쿨링없이 쉽게 오버클럭이 된다는 점
- 2011년 코어 i3 샌디브릿지 2100 시리즈
- 2011년 코어 i5 샌디브릿지 2300, 2400, 2500 시리즈
- 2011년 코어 i7 샌디브릿지 2600 시리즈

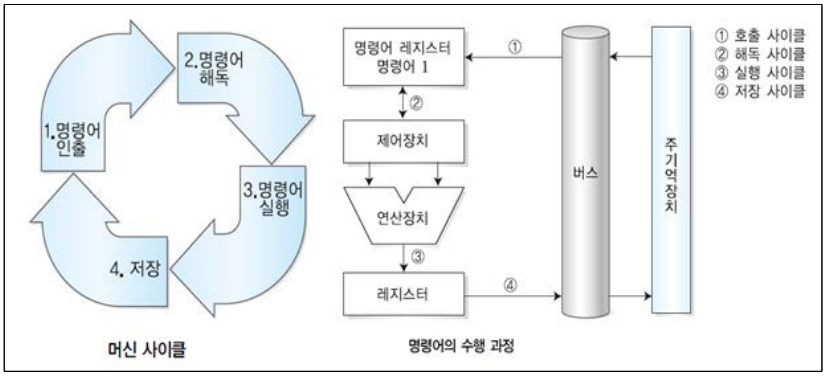
2. CPU와 CPU의 이해

1) CPU(microprocessor)

- 컴퓨터에서 명령을 수행하고 데이터를 처리하는 중앙처리장치(CPU, Central Processing Unit)를 집적회로의 칩 형태로 만든 것임
- 폰 노이만(von Neumann) 컴퓨터 구조
    - 프로그램내장 방식임
    - 데이터와 명령어가 주기억장치에 저장되어 있다가 버스를 통해서 CPU로 전달함
    - CPU는 전달되어온 명령어를 이용 데이터를 사용자가 원하는 형태로 처리함
    - 결과는 다시 데이터버스를 통해서 주기억장치 RAM으로 보냄
- 주 회로기판(main board)에서 CPU와 주기억장치의 위치

2) 머신 사이클 (machine cycle)

- 프로그램을 구성하는 명령어는 4단계의 과정을 통해서 수행, 그런데 이 과정은 CPU에서 동작을 하므로 머신 사이클이라고 함
- 각 단계별 사이클의 역할
    - 인출(Fetch) 사이클 : 필요한 명령어를 주기억장치에서 불러오는 사이클임
    - 해독(Decode) 사이클 : 호출된 명령어를 해석하는 사이클임
    - 실행(Execute) 사이클 : 해석된 명령어를 산술논리연산장치를 통하여서 실행함
    - 저장(Store) 사이클 : 수행결과를 주기억장치에 저장하는 사이클임



3) 클록속도와 명령어 처리속도

* 클록(Clock) 주파수
- CPU는 일정한 속도로 작동하기 위해서 일정한 간격으로 공급되는 전기적 진동(pulse) 을 클록이라 함
- 1초에 클록이 몇 번 발생하는지를 나타낸 것을 클록 주파수라 하며, 단위는 Hz
- 1초에 1번 클록이 발생하는 것을 1Hz라고 함
- 1초에 106 개의 클록 발생하면 클록 주파수의 단위는 MHz로 표현함
- 1초에 109의 클록이 발생하면 단위는 GHz로 표현함
- 75MHz라면 초당 7천 5백만 번의 사이클로 0과 1의 디지털 신호를 발생함

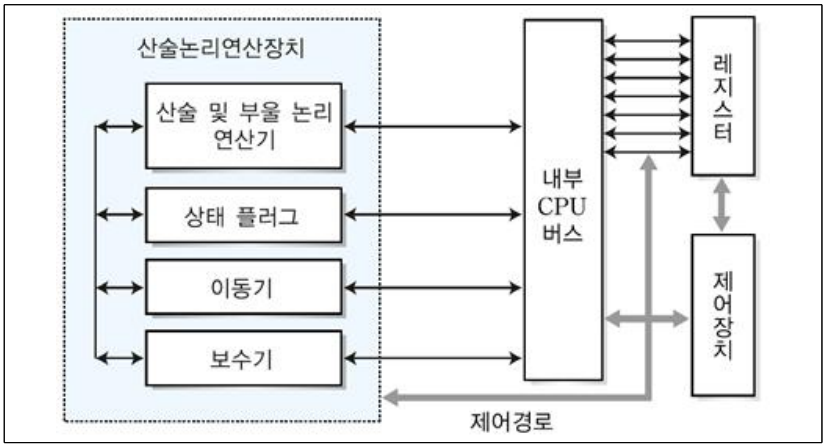
<주회로기판에 장착되어 있는 클록 발생기가 만들어 내는데, 클록 수가 높을수록 컴퓨터의 처리 속도가 빠르다는 것을 의미>

* 명령어 처리 속도
- 초당 처리하는 명령어의 개수로 단위는 1초에 100만 개의 명령어 수행을 나타내는 MIPS(Million Instruction Per Second)으로 나타냄
- 처리속도가 18.5MIPS라면 1초 동안에 1,850만 개의 명령을 실행할 수 있음

학습내용2 : CPU 내부구조 1

1. CPU의 조직

1) CPU의 내부구조는 기본적으로 연산장치, 제어장치, 레지스터의 집합으로 구성되며, 이것들은 내부 CPU 버스로 연결되어 있음
- 연산장치에서 각종 연산기능을 수행함
- 레지스터에서 데이터를 보관하는 기억기능을 수행함
- 제어장치는 명령을 해독하고 제어신호를 발생하여 제어기능을 수행함
- 버스를 통해서 데이터의 전달기능을 수행함
- 결과적으로 CPU는 기본적으로 연산, 기억, 제어, 전달 등 네 가지 기능을 수행함



1) 연산장치

① 산술 논리 연산장치(ALU, Arithmetic and Logic Unit)
- 덧셈, 뺄셈과 같은 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 회로임
    - 덧셈, 뺄셈, 곱셈, 나눗셈 등의 산술연산과 논리 연산으로는 AND, OR, NOT, XOR 등을 수행함
    - 이외에 많은 산술, 논리 연산을 수행할 수 있음

③ 상태 플래그(Status Flags)
    - 연산 중인 ALU 내의 데이터 상태를 표시함
    - 음수, 0, 오버플로우 등을 표시함

③ 이동기(Shifter)
    - 데이터 비트를 좌우로 비트 별로 이동(비트의 이동은 2로 곱셈하거나 나눗셈하는 것으로 해석)함

④ 보수기(Complmenter)
    - ALU내의 데이터에 대하여 보수연산을 수행함
    - 컴퓨터에서는 2의 보수를 주로 사용함
    - 2의 보수는 덧셈과 뺄셈 계산장치의 제작을 쉽게 함

2) 프로세서 레지스터와 스택

① 프로세서 레지스터
- CPU내에서 데이터를 저장하는 장치, 간략하게 레지스터라고도 함
- 컴퓨터의 기억장치들 중에서 속도가 가장 빠름
    - ALU에서 처리된 결과 데이터를 임시적으로 보관함
    - 주기억장치로부터 읽어온 명령어와 데이터를 임시적으로 보관함
- 레지스터의 용도별 분류
    - 데이터 레지스터 : 정수 데이터 값을 저장할 수 있는 레지스터임
    - 주소 레지스터 : 기억장치 주소를 저장하여 기억장치 액세스에 사용함
    - 범용 레지스터 : 데이터와 주소를 모두 저장할 수 있는 레지스터임
    - 부동 소수점 레지스터 : 부동소수점 데이터 값을 저장하기 위해 사용 함
    - 상수 레지스터 : 0이나 1 등 고정된 데이터 값을 저장하기 위한 레지스터
    - 특수 레지스터 : 실행 중인 프로그램의 상태를 저장하는 레지스터(프로그램 카운터, 상태 레지스터)다.
    - 명령 레지스터 : 현재 실행중인 명령어를 저장함
    - 색인 레지스터 : 실행 중에 피 연산자의 주소를 계산하는데 사용됨

3) CPU에 존재하는 레지스터

- 사용자에게 보이는 레지스터들과 제어 및 상태 레지스터들로 분류함

① 사용자에게 보이는 레지스터들
- 어셈블리 프로그래머는 프로그램에서 사용되는 변수 데이터 등의 저장을 위해 해당 레지스터를 알고 있어야 함
- 사용하는 목적에 따른 분류
    - 일반목적용 레지스터 : 프로그래머에 의해 여러 용도로 사용함
    - 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 레지스터
- 조건 코드(Condition Codes) : 저장된 데이터의 상태를 표시하는데 사용됨
    - 부호(sign) 비트 : 경우에는 양수인지 음수인지를 표시함
    - 영(0) 비트 : 해당 데이터가 0이라는 것을 표시함
    - 오버플로우 비트 : 연산의 결과 등에 오버플로우가 발생했다는 것을 표시함

② 제어 및 상태 레지스터들(Control and Status Registers)
- 프로그램 카운터(Program Counter)
    - 주기억장치에 저장된 다음에 인출할 명령어의 주소를 가지고 있는 레지스터임
- 명령어 레지스터(Instruction Register)
    - 가장 최근에 주기억장치인 RAM에서 인출한 명령어를 저장함
- 기억장치 주소 레지스터(Memory Address Register)
    - 액세스할 기억장치의 주소가 저장되는 레지스터임, 이 레지스터의 출력이 주소 버스와 직접 연결됨
- 기억장치 버퍼 레지스터(Memory Buffer Register)
    - 기억장치에 쓰여질 데이터 혹은 가장 최근에 읽은 데이터가 저장됨
- 입/출력 주소 레지스터(I/O AR: I/O Address Register)
    - 입/출력 장치의 주소를 저장하는 주소 레지스터임
- 입/출력 버퍼 레지스터(I/O BR: I/O Buffer Register)
    - 입/출력 모듈과 CPU 사이에 교환되는 데이터를 일시적으로 저장하는 레지스터임

4) 제어 및 상태 레지스터

① 프로그램 상태 단어(Program Status Word) : 저장된 데이터의 상태와 조건을 나타내기 위하여 추가된 조건 코드 비트

- 부호(sign) 비트 : 해당 레지스터내의 데이터의 부호를 표시함
- 영(zero) 비트 : 레지스터가 0이라는 것을 표시함
- 올림수(carry) 비트 : 해당 레지스터에서 자리 올림이 발생하였다는 것을 표시함
- 동등(equal) 비트 : 비교 대상과 해당 레지스터가 동일한 상태임을 표시함
- 오버플로우(overflow) 비트 : 해당 레지스터의 오버플로우 상태를 표시함
- 인터럽트 가능/불가능(interrupt enable/disable) 비트 : 인터럽트 가능 여부를 표시함
- 디렉션(direction) 비트 : 문자열 조작에서 참일 경우 주소 레지스터 값이 자동으로 감소하고, 거짓일 경우 자동으로 증가하도록 하는 비트임

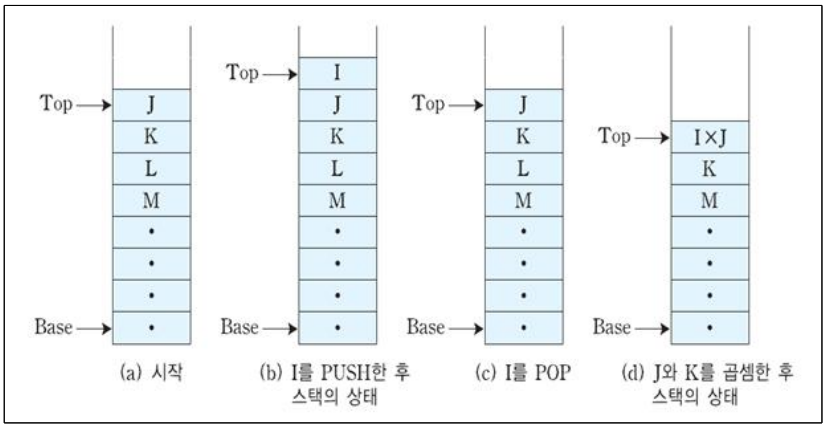
- 트랩(trap) 비트 : 참일 경우 한 명령이 실행할 때마다 인터럽트가 발생함
- 보조올림수(auxiliary carry) 비트 : 연산 결과 하위 니블(4bits)에서 비트 범위를 넘어섰을 때 참이 됨 이진화 십진법(BCD) 연산에 사용하는 비트임
- 패리티(parity) 비트 : 연산 결과에서 1의 값을 갖는 비트의 수가 짝수일 경우 참임
- 수퍼바이저(supervisor) 비트 : CPU가 수퍼바이저 모드 혹은 사용자 모드에서 실행 중인지를 나타내는 비트임

학습내용3 : CPU 내부구조 2

1. CPU의 조직

1) 스택(Stack) 저장장치
- CPU 내부의 레지스터 집합에 존재하는 저장장치
- 스택의 특징
    - 저장되는 요소들이 순차적으로 저장함
    - 요소의 개수 또는 스택의 길이는 가변적임
    - 한 번에 하나의 요소에만 액세스 가능하다.
    - 나중 입력 먼저 출력(LIFO, Last-In-First-Out)함
- 스택의 동작 표현
    - TOP : 데이터가 입력되고 출력되는 액세스 부분임
    - PUSH : 스택의 Top에 새로운 요소를 추가 저장하는 동작임
    - POP : 스택의 Top에서 하나의 요소를 꺼내는 동작임
    - 스택 포인터 : Top의 위치를 표시하는 장치다.

2) 스택의 기본적인 동작
- 스택의 동작 과정
    - Top에 위치한 요소는 POP동작을 통해서 스택에서 인출함
    - 연산을 수행하고, 연산의 결과를 다시 스택에 저장되도록 PUSH함
    - 결과 데이터가 저장되었으므로 Top은 위쪽으로 이동하게 됨

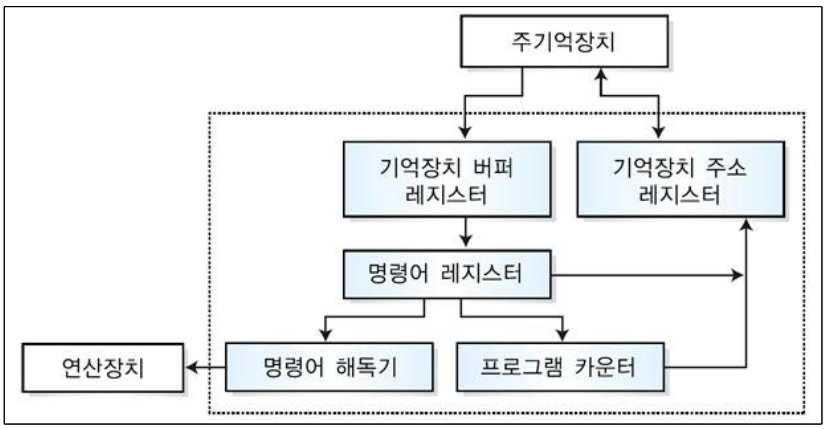


3) 제어장치
- 명령어를 해독하는 기능과 제어 신호를 해당장치에 전달하는 역할을 수행
- 명령어의 형식

  - 연산 코드필드는 수행되어야 할 연산이 지정되어 있는 필드임
  - 기억장치의 주소 필드는 해당 연산을 수행할 때 데이터가 저장되어 있는 주소임

- 제어장치의 구성요소
  - 기억장치 버퍼 레지스터 : 주기억장치에서 읽어온 명령어를 임시적으로 저장하는 곳임
  - 명령어 레지스터 : 명령어를 저장하는 곳임
  - 명령어 해독기 : IR에 저장된 명령어의 연산 코드 필드를 전달받아서 명령어를 해독하여 수행할 연산을 결정함
  - 기억장치주소 레지스터 : 명령어 레지스터에 저장된 명령어의 주소 번지를 저장함
  - 프로그램 카운터 : 다음에 수행할 명령어의 주소 번지를 저장하고 있는 곳임

4) 제어장치 구성



5) 내부 CPU 버스(Internal CPU Bus)
- CPU 내의 ALU와 레지스터들 간의 데이터 이동과 ALU와 제어장치 간의 데이터 이동 그리고 제어장치와 레지스터들 간의 데이터 이동을 위한 통로임
- 실질적인 데이터를 전달하는 데이터 버스와 제어장치에서 발생되는 제어 신호를 전달하는 제어버스로 구성됨
- CPU 밖의 시스템 버스들과는 직접 연결되지 않으며, 반드시 버퍼 레지스터들 혹은 시스템 버스 인터페이스 회로를 통하여 시스템 버스와 접속하는 특징을 가지고 있음
- 기억장치 버퍼 레지스터와 기억장치 주소 레지스터는 CPU내부와 외부 장치 간에 속도 차이를 극복하기 위한 버퍼 역할을 수행함

【학습정리】

1. CPU의 역사는 하드웨어 기술의 발전으로 분류하는 것과 소프트웨어의 발전으로 분류할 수 있다.

2. CPU 내부 구조를 이해하면 컴퓨터 구조 전체를 이해하는데 도움이 된다.