

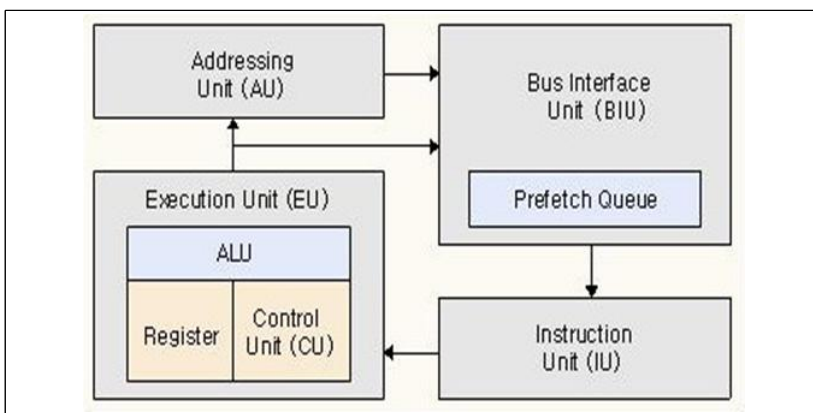
## 6주차 3차시 CPU의 기능과 성능

### 【학습목표】

1. CPU의 4가지 구성요소를 특징과 함께 설명할 수 있다.
2. CPU의 4단계 기본 동작과 제어장치의 기본 동작 과정을 설명할 수 있다.

### 학습내용1 : CPU의 기능

#### 1. CPU의 4가지 구성요소



##### ① 실행 유닛(EU : Execution Unit)

- CPU의 각 명령어를 수행하는 유닛
- 실행유닛은 ALU(Arithmetic Logic Unit)와 제어 유닛(CU:Control Unit), 레지스터로 구성
- ALU에서는 덧셈이나 비교 연산과 같은 산술 연산, 논리연산을 수행한 후 중간 결과를 레지스터에 보관

##### ② 명령어 유닛(IU : Instruction Unit)

- 메모리에서 가져온 명령어를 실행유닛에서 정확히 수행할 수 있도록 제어하는 유닛
- 이때 제어 유닛은 ALU나 레지스터들이 잘 동작되도록 하여 줌

##### ③ 어드레싱 유닛(AU : Addressing Unit)

- CPU가 메모리나 입출력 장치에서 데이터를 읽거나 쓸 때, 메모리나 입출력 어드레스를 만들어 주는 유닛

##### ④ 버스 인터페이스 유닛(BIU : Bus Interface Unit)

- 어드레스 버스, 데이터 버스, 컨트롤 버스 등의 시스템 버스를 통해서 CPU 외부와 연결시켜 주는 유닛

## 2. CPU의 기능과 동작

〈CPU가 모든 명령어들에 대하여 공통적으로 수행되는 기능은 명령어 인출(Instruction Fetch)과 명령어 해독(Instruction Decode)임〉

### 1) 추가되는 명령어의 기능들

기능	내용
데이터 인출(data fetch) 기능	<ul style="list-style-type: none"> <li>명령어 실행을 위하여 데이터가 필요한 경우, 기억장치 또는 입출력장치에서 그 데이터를 읽어오는 과정</li> <li>연산 과정에서 사용하는 데이터를 불러오는 과정이라고 할 수 있음</li> </ul>
데이터 처리(data process) 기능	읽어온 데이터에 대한 산술적 또는 논리적 연산을 수행함
데이터 쓰기(data store) 기능	데이터 처리 과정에서의 수행 결과를 저장하는 기능

### 2) CPU 기본 기능

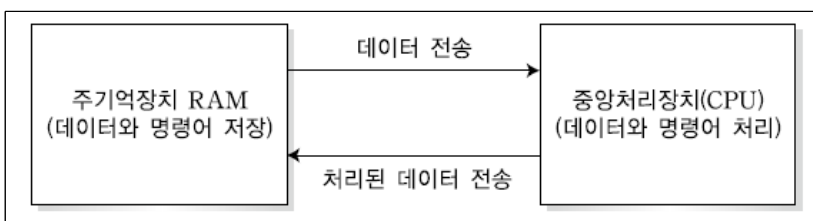
- CPU는 데이터 버스를 통해서 데이터를 메모리에 읽거나 쓸 수 있으며, 데이터를 읽거나 쓰는데 사용할 메모리 위치는 어드레스에 의해서 정해짐
- CPU는 어드레싱 유닛에서 어드레스를 계산한 후 어드레스 버스로 보내게 되며, 데이터를 읽거나 쓰는 일은 버스 유닛에서 수행함
- 버스 유닛은 메모리의 어드레스를 내보내고 메모리에 쓸 데이터를 내 보내거나, 어드레스를 내보내고 메모리에서 데이터를 읽어 들임
- 또한 버스 유닛은 CPU에서 수행할 명령어를 읽을(Fetch)때에도 데이터를 읽을 경우와 마찬가지로 동작함

\* 버스 유닛에서 데이터를 읽어 들였을 경우 : 그 데이터가 레지스터에 저장됨

- 그러나 버스 유닛이 명령어를 읽어들였을 경우는 그 명령어가 프리패치(Prefetch) 큐(queue)에 저장됨

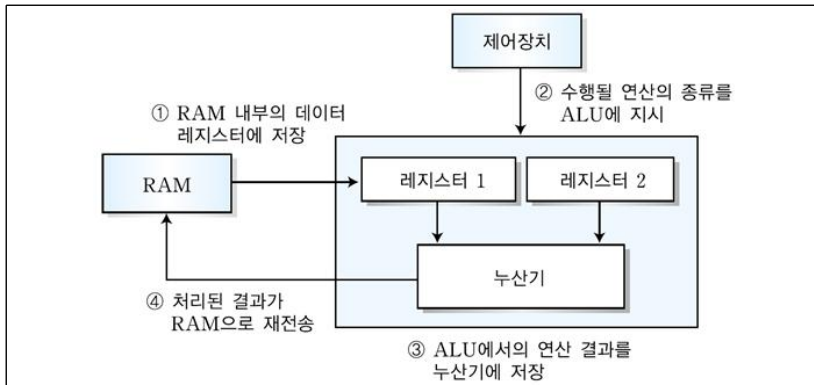
- 명령어 유닛은 프리패치 큐에서 명령어를 읽어 들여 해석하고(Decode), 해석된 명령어(Decoded Instruction)를 수행 유닛으로 보내며, 수행 유닛에서는 해석된 명령어에 따라서 ALU와 레지스터가 제어되어 명령어에 맞는 동작을 수행하게 됨

### 3) CPU와 주기억장치와의 데이터 전송



## 학습내용2 : CPU의 동작

### 1. CPU의 4단계 기본 동작



\* 누산기(Accumulator) : 데이터 레지스터로 처리 결과를 임시로 보유하는 역할

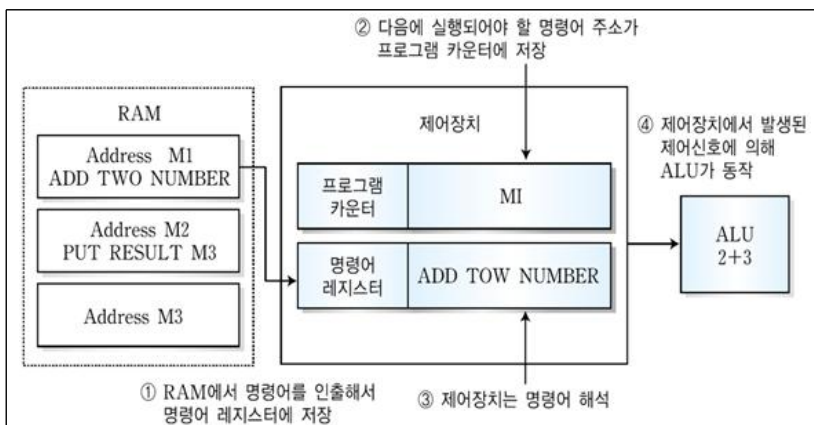
- ① 처리해야 할 데이터는 주기억장치 RAM에서 인출되고 외부 시스템 버스를 통해서 레지스터 1번으로 전달됨
- ② 제어장치는 새롭게 저장된 레지스터 1번 데이터와 이전부터 저장하고 있던 레지스터 2번의 데이터를 덧셈하라는 제어신호를 ALU로 전달됨
- ③ ALU에서는 제어신호에 의해서 덧셈을 수행하고 그 결과를 누산기에 저장
- ④ 덧셈의 계산 결과는 외부 시스템 버스를 통해서 다시 주기억장치로 전달됨

### 2. 제어장치의 기본 동작

#### 1) 제어장치의 기본 동작 과정

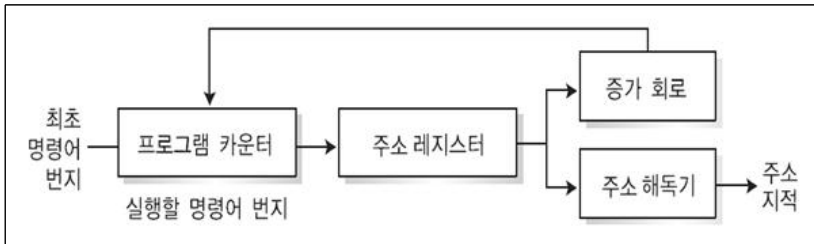
- ① 주기억장치에서 명령어를 인출해서 제어장치 내에 명령어 레지스터로 저장됨
- ② 프로그램 카운터는 다음에 실행될 명령어의 주소가 저장됨
- ③ 제어장치가 명령어 레지스터의 명령어를 해석함
- ④ 해석된 명령어는 해당되는 제어신호를 발생하게 됨

#### 2) 제어장치의 기본 동작 과정

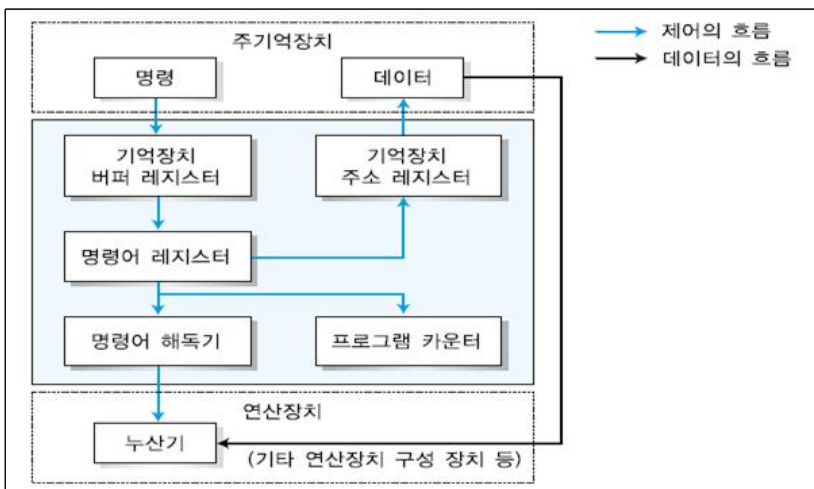


### 3) 프로그램 카운터의 역할

- 프로그램에서 항상 앞에서부터 한 명령씩 차례대로 실행되도록 함
- 조건부 분기(예: JUMP)와 같이 그 순서를 바꾸어야 하는 경우, 프로그램 카운터의 내용을 바꿈으로써 분기된 후부터의 명령들이 새로운 순서에 맞게 실행함



### 3. 제어장치의 동작을 포함한 CPU의 동작



## 학습내용3 : 성능 요소

### 1. CUP의 성능요소

<컴퓨터의 CPU가 데이터를 처리하는 속도는 컴퓨터의 성능을 평가하는 중요한 요인>

- CPU가 데이터를 처리하는 속도는 여러 가지 요소들에 의해서 좌우

\* 대표적인 요소

- 클럭(Clock) 주파수, 워드(word) 크기, 캐시 메모리, 명령어 집합의 복잡성, 파이프라이닝(Pipelining), 병렬처리(Parallel Processing)가 있음

① 클럭 주파수

- 컴퓨터에서 수행되는 모든 연산의 타이밍을 맞추기 위해 펄스를 시스템 클럭이라 함
- 클럭 주파수는 컴퓨터가 명령어를 수행하는 속도를 결정
  - 하나의 클럭 동안에 명령어 부 사이클이 수행됨
  - 클럭의 주기가 길면 그 만큼 처리할 수 있는 명령어 부 사이클의 시간이 지연됨
  - 클럭의 주기는 클럭의 주파수와 반비례이므로 짧은 주기는 높은 클럭 주파수를 뜻함
- 결과적으로 클럭 주파수는 특정시간 동안에 완수할 수 있는 명령어의 수를 제한
- 측정 단위는 MHz(megahertz : millions of instructions per second)

② 워드 크기

- CPU가 한 번에 읽고(read), 쓸(write) 수 있는 비트 수
- 워드의 크기는 레지스터의 크기와 버스의 데이터 선로 수에 달려 있음
- 워드 크기가 큰 컴퓨터는 한 명령어에서 더 많은 데이터를 처리할 수 있음

③ 캐시기억장치

- CPU가 데이터에 빠르게 접근할 수 있는 고속의 기억장치
- 읽기와 쓰기 동작의 속도를 향상시켜서 전체적으로 CPU 속도에 영향을 줌

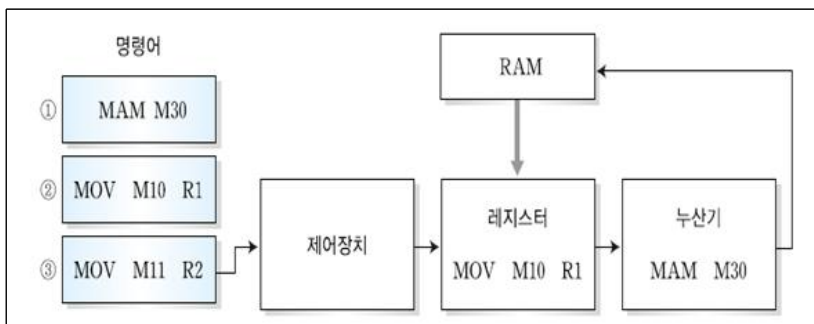
④ 명령어 집합의 복잡성

- 축소 명령어 집합 컴퓨터(RISC, Reduced Instruction Set Computer)는 연산속도를 향상 시키기 위해 제어논리를 단순화하고 단순화된 명령어 구조를 가짐
- 축약 명령어 집합 컴퓨터는 CPU에 빠르게 수행되는 제한된 수의 간단한 명령어만이 내재된 컴퓨터
- 결과적으로 특별한 설계 방법을 통해 속도를 최대한 높일 수 있는 컴퓨터

⑤ 파이프라이닝(Pipelining) 기법

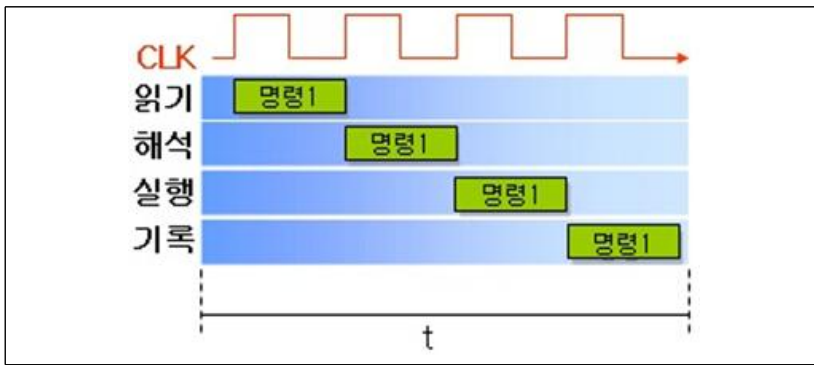
- 파이프들이 연속적으로 연결되는 개념
- CPU 또는 프로세서가 이전 명령어의 수행이 완전하게 종료되기 전에 새로운 다음 명령어 수행을 시작하는 기법임

\* 파이프라이닝의 동작원리

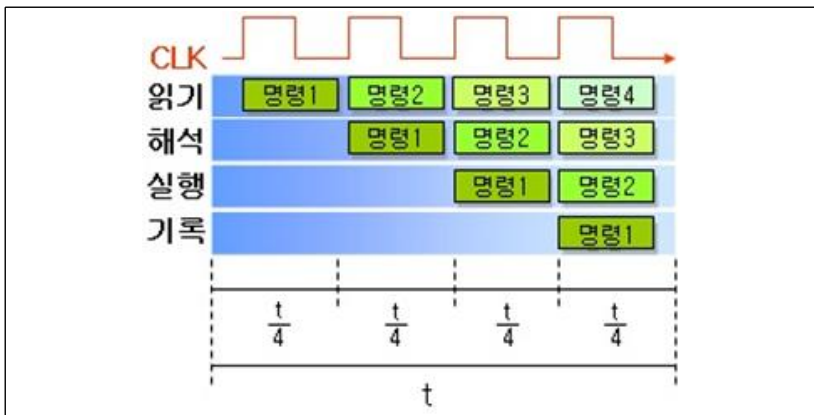


- 제어장치가 3번 명령어를 해독하는 동안 2번 명령어에 필요한 데이터는 레지스터로 가고 1번 명령어는 누산기에서 수행을 마침
- 명령어들의 부 사이클이 동시간에 처리될 수 있어, CPU의 처리 속도를 증가 시킬 수 있음
- 명령어를 수행하기 위한 각각의 스테이지를 병렬로 구동시켜 명령어의 처리 흐름이 끊이지 않도록 하여 마치 수도관을 흐르는 물처럼 끊임없이 명령어가 처리 된다고 하여 이러한 이름이 붙었음

\* 일반적인 명령어 처리 방식



\* 파이프라이닝(Pipelining) 기법 처리 방식



- 파이프라이닝 구조를 사용한 프로세서의 스테이지임

- 이제 이 프로세서의 처리율은 얼마나 되는 것일까?

- 명령1은 기록 처리 중이고, 명령2는 실행 중이며, 명령3은 해석중이고, 명령4는 읽기 단계에 있으므로 이것을 식으로 풀어보면 다음과 같음

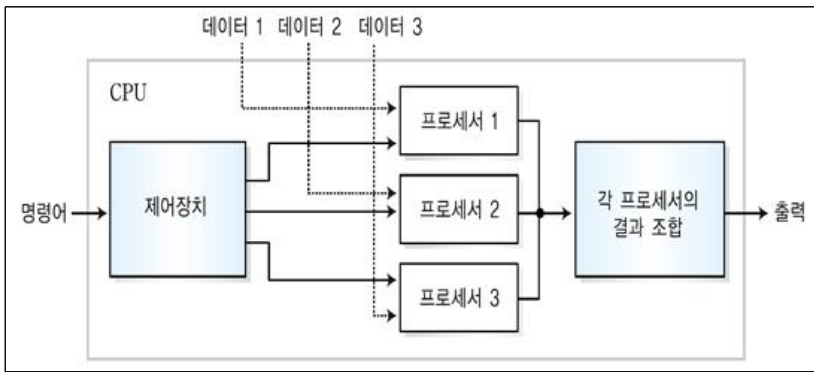
$$- 4/4\text{명령}/t + 3/4\text{명령}/t + 2/4\text{명령}/t + 1/4\text{명령}/t = 2.5\text{명령}/t$$

- 처리율이 무려 2배 이상 증가

⑥ 병렬처리

- 하나 이상의 CPU로 구성된 컴퓨터에서 한번에 여러 개의 명령어를 동시에 수행시킬 수 있는 방법을 병렬처리(Parallel processing)라고 함

\* 병렬처리의 동작원리



- 3개의 CPU가 존재함
- 명령어가 제어장치로 입력되면 제어장치는 이 명령어가 수행이 가능한 CPU로 보냄
- 여분의 CPU가 다른 명령어를 처리할 수 있으므로 대기하는 시간 없이 바로 처리됨  
→ 그래서 컴퓨터의 처리속도가 증가하게 됨
- 여러 개의 CPU가 필요하므로 비용이 상승하지만 그 만큼 컴퓨터가 처리하는 속도는 증가하게 될 것임

【학습정리】

1. 명령을 해석하고 수행하는 머신사이클은 인출-해독-실행-저장의 4단계로 구분된다.
2. CPU의 성능 요소는 클럭(clock) 주파수, 워드(word)크기, 캐시기억장치, 명령어 집합의 복잡성, 파이프라이닝, 병렬처리 등에 의해 평가된다.