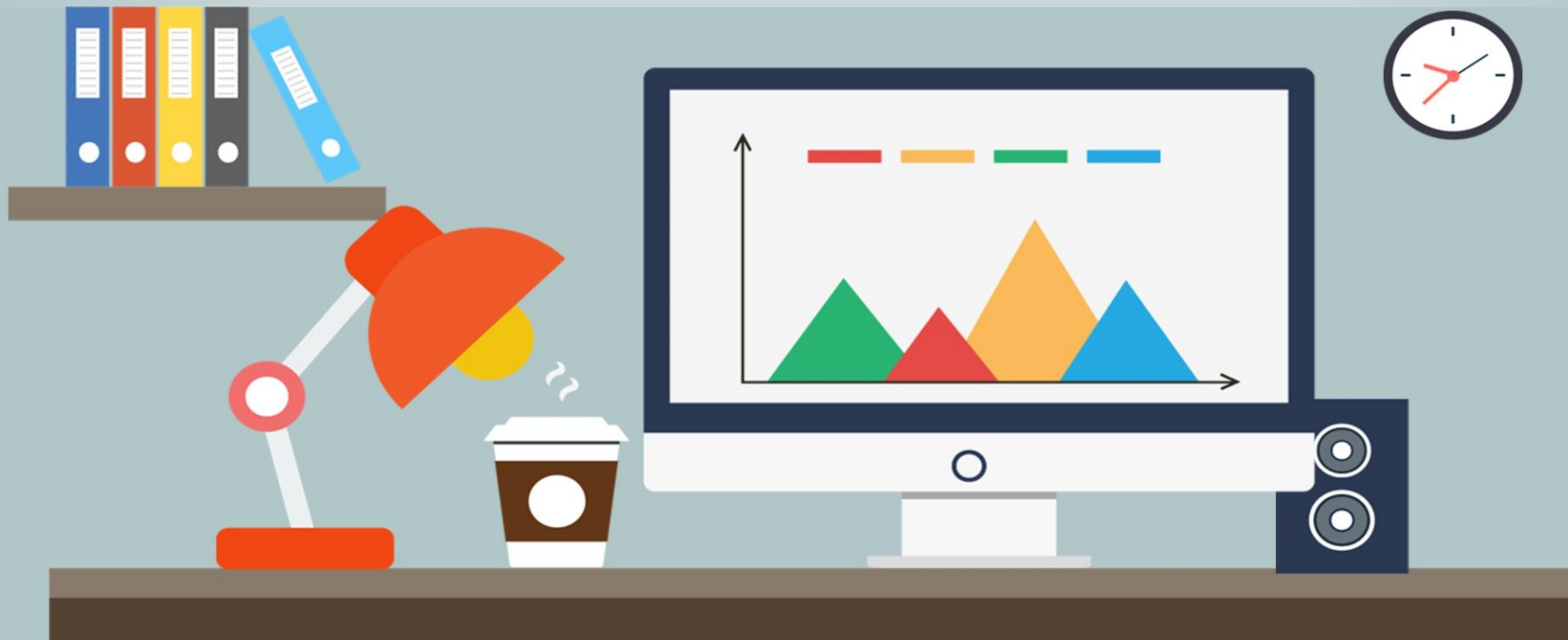


12주 3강

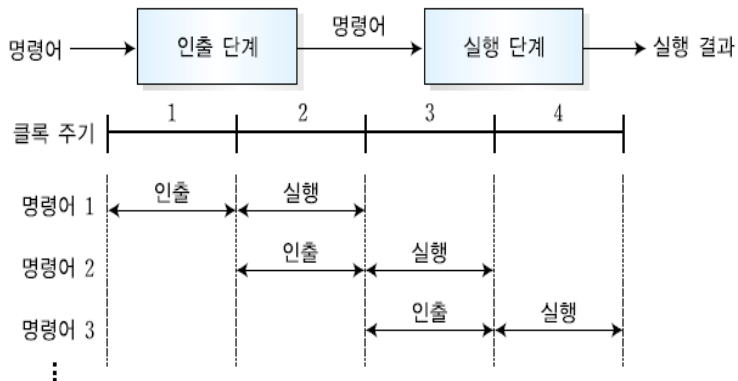
# 명령어 파이프라이닝



# 명령어 파이프라이닝



- 명령어 실행 도중에 다른 명령어 실행을 시작하여 동시에 여러 개의 명령어를 실행하는 기법
  - 한 명령어를 여러 단계로 나누고 그 명령어의 특정 단계를 처리하는 동안 다른 부분에서는 다른 명령어의 다른 단계를 처리하여 처리 속도를 향상
- 2-단계 명령어 파이프라인
  - 명령어 실행을 인출단계(fetch stage)와 실행단계(execute stage)라는 두 개의 독립적인 파이프라인 모듈로 분리하여서 수행하는 방법



## 2-단계 파이프라인의 특징



- 2-단계 파이프라인을 이용하면 명령어 처리 속도가 약 두 배 향상
- 이론상 단계 수가 늘어날수록 그만큼의 속도 향상을 가져온다.
- 두 단계의 처리 시간이 동일하지 않으면, 두 배의 속도 향상을 얻지 못한다
  - 각 명령어의 인출단계와 실행단계의 처리 시간이 동일해야 파이프라인에 의한 효율 향상을 얻을 수 있다.
  - 단계의 처리 시간이 동일하지 않으면 파이프라인 단계의 수를 증가시켜서 각 단계의 처리 시간을 같게 한다.
- 단계의 세분화
  - 여러 단계간의 시간차이를 거의 없게 한다.
  - 파이프라인 단계의 수를 늘리면, 전체적으로 속도 향상도 더 높아지게 된다.
  - 문제점 극복과 속도향상을 동시에 얻을 수 있게 된다.

# 4-단계 명령어 파이프라인



## ● 명령어 인출, 명령어 해독, 오퍼랜드 인출, 실행의 4단계

### ① 명령어 인출(IF, Instruction Fetch)단계

- 명령어를 기억장치에서 인출하는 과정
- PC에 제시된 기억장치 주소에 근거해서 명령어를 인출하여 IR로 이동

### ② 명령어 해독(ID, Instruction Decode)단계

- 명령어 해독기(decoder)를 이용하여 인출된 명령어를 해석한다.

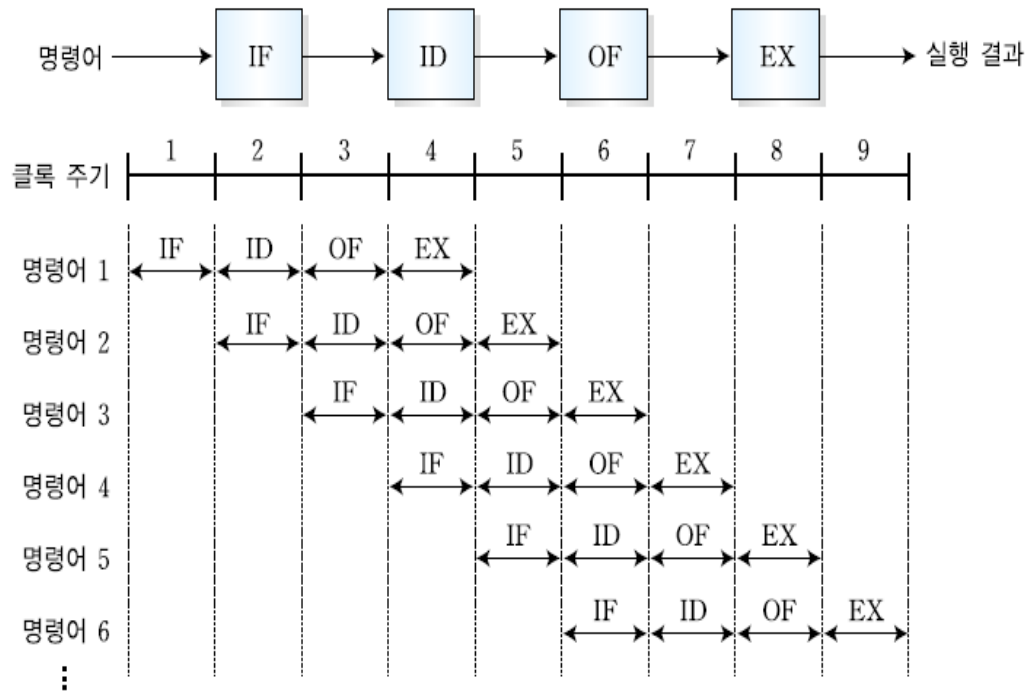
### ③ 오퍼랜드 인출(OF, Operand Fetch) 단계

- 기억장치에서 오퍼랜드를 인출하는 단계다.
- 오퍼랜드는 피연산자 부분으로 연산에 사용될 변수나 데이터
- 인출된 피연산자들은 실행단계에서 각종 연산에서 사용될 것이다.

### ④ 실행(EX, Execute)단계

- 명령어에서 지정된 연산을 수행하는 단계다.

# 4-단계 명령어 파이프라인과 시간흐름도

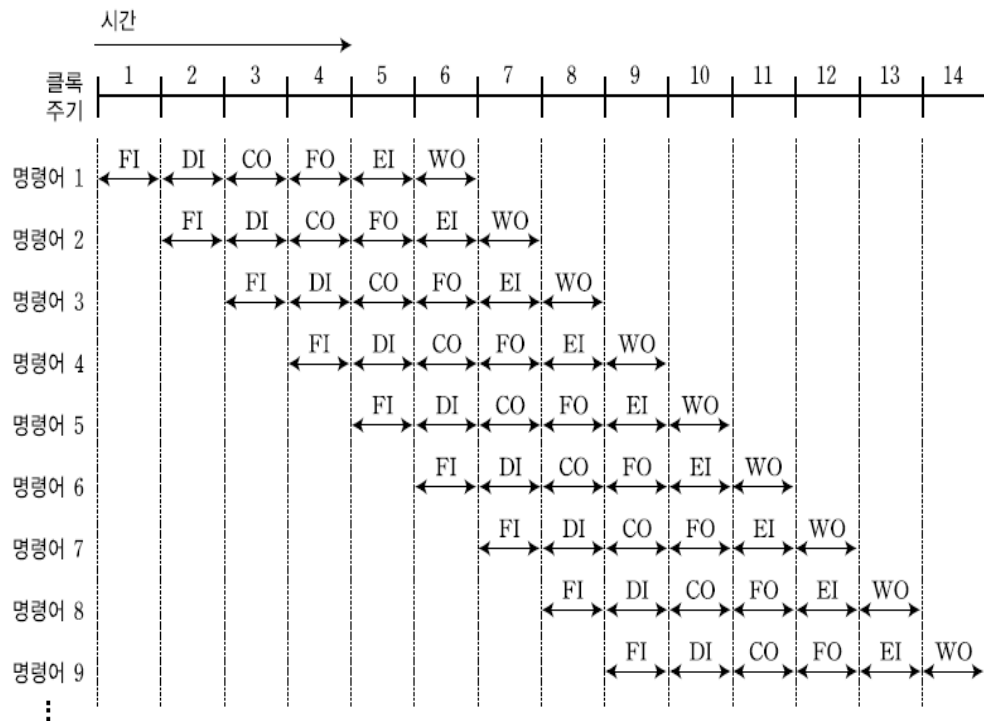


# 6-단계 명령어 파이프라인



- FI(Fetch Instruction)단계 : 명령어 인출단계
- DI(Decode Instruction)단계 : 명령어 해독단계
- CO(Calculate Operand)단계 : 오퍼랜드를 계산하는 단계
  - 오퍼랜드는 일반적인 데이터 값뿐만 아니라 주소일 수도 있다.
  - 간접주소 방식에서는 유효주소를 찾는 계산이 필요하다.
- FO(Fetch Operand)단계 : 오퍼랜드 인출단계
  - 오퍼랜드 값은 세 번째에서 계산된 값이다.
- EI(Execute Instruction)단계 : 명령어 실행단계
- WO(Write Operand)단계 : 연산된 결과인 오퍼랜드를 저장하는 단계

# 6-단계 명령어 파이프라인 시간흐름도



# 파이프라인에 의한 속도 향상



- 파이프라인의 단계 수  $k$ , 실행할 명령어들의 수는  $N$ 이고, 각 파이프라인 단계가 한 클록 주기씩 걸린다고 가정

- 파이프라인에 의한 전체 명령어 실행 시간  $T$

$$T = k + (N - 1)$$

- 첫 번째 명령어를 실행하는데  $k$ 주기가 걸리고, 나머지  $(N - 1)$ 개의 명령어들은 각각 한 주기씩만 소요
- 파이프라이닝을 적용하지 않을 경우,  $N$ 개의 명령어들을 실행하는 데에  $(k \times N)$ 주기가 소요



# 파이프라인에 의한 속도 향상의 예 (1)



- [예 1] 파이프라인 단계 수  $k=4$   
파이프라인 클럭 = 1 MHz  
(각 단 계에서의 소요시간 =  $1\ \mu\text{s}$ )인 경우
  - 첫 번째 명령어 실행에 걸리는 시간 =  $4\ \mu\text{s}$
  - 다음부터는 매  $1\ \mu\text{s}$  마다 한 개씩의 명령어 실행 완료
  - 10개의 명령어 실행 시간  $T = 4 + (10 - 1) = 13\ \mu\text{s}$
  - 파이프라이닝에 의한 속도 향상 :  $(10 \times 4) / 13 \approx 3.08\text{배}$

# 파이프라인에 의한 속도 향상의 예(2)



• [예 2] 파이프라인 단계 수  $k = 4$

명령어의 수  $N$ 을 증가시킬 때, 속도향상( $S_p$ )

- $N = 100$ 이라면,  $S_p = 400 / 103 = 3.88$
- $N = 1000$ 이라면,  $S_p = 4000 / 1003 = 3.99$
- $N = 10000$ 이라면,  $S_p = 40000 / 10003 = 3.998$
- $N \rightarrow \infty$ ,  $S_p = 4$
- 결과적으로 명령어의 수가 많으면 파이프라인 단계 수만큼 속도가 증가한다.

다음 시간

## 13주. 마이크로 연산과 제어장치

