



5. CPU의 성능 향상 기법

5-1. 빠른 CPU를 위한 설계 기법

CPU의 속도를 빠르게 만드는 방법?

- ⇒ 1. 클럭 속도를 높이기
- 2. 코어 수를 늘리기
- 3. 스레드 수를 늘리기

1. 클럭 속도

클럭 속도를 빠르게 하면 ⇒ 일반적으로 CPU의 속도를 빠르게 할 수 있음.

BUT, 클럭 신호를 필요 이상으로 높이면 발열이 심해지고, 속도가 빨라지지 않음.

<클럭 속도>

- 클럭 속도 : 헤르츠(Hz) 단위로 측정, 헤르츠(Hz) : 1초에 클럭이 반복되는 횟수
- 클럭이 '똑-딱'하고 1초에 한번 반복되면 1Hz, 1초에 100번 반복되면 100Hz

2. 코어 & 멀티 코어

코어(Core)란?

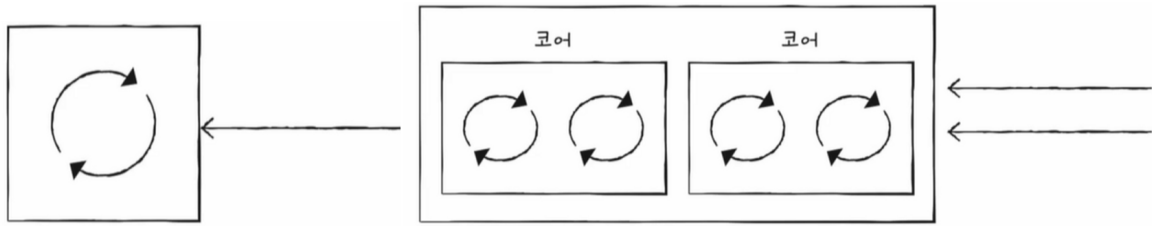
- CPU 내에서 명령어를 실행하는 부품으로, 여러 개 존재할 수 있음
- 멀티코어 프로세서 : 여러 개의 코어를 포함하고 있는 CPU

⇒ BUT, 코어 수에 비례해서 CPU 속도가 빨라지는 것은 아니고, 명령어 분배가 중요함

3. 스레드 & 멀티 스레드

스레드란?

- 실행 흐름의 단위
- 하드웨어적 스레드, 소프트웨어적 스레드로 구분
- 하드웨어적 스레드 : 하나의 코어가 동시에 처리하는 명령어 단위로, **논리 프로세서**라고도 함

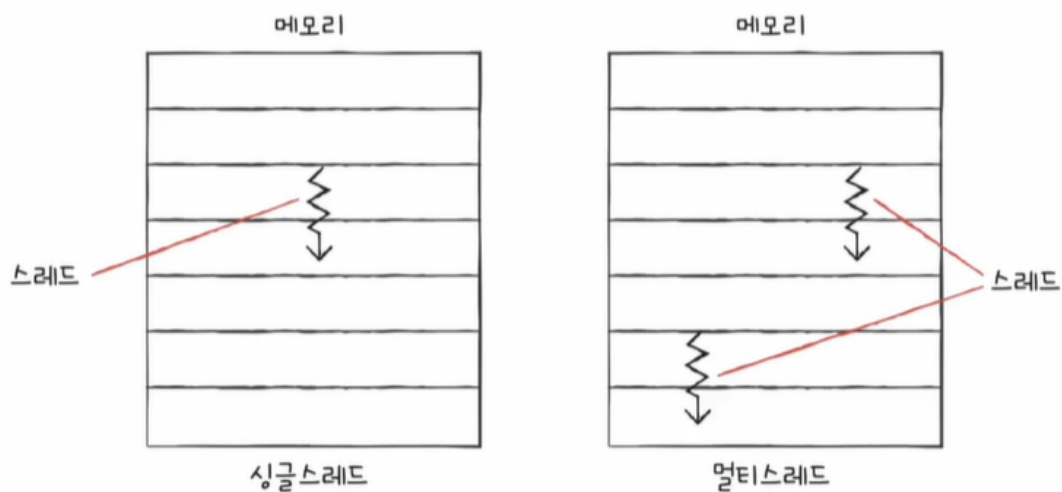


1코어 1스레드 CPU

2코어 4스레드 CPU

2코어 4스레드 CPU 같은 것을 멀티스레드 프로세서, 멀티스레드 CPU라고 부른다.

- 소프트웨어적 스레드 : 하나의 프로그램에서 독립적으로 실행되는 단위로, 1코어 1스레드 CPU도 여러 소프트웨어적스레드를 만들 수 있음.

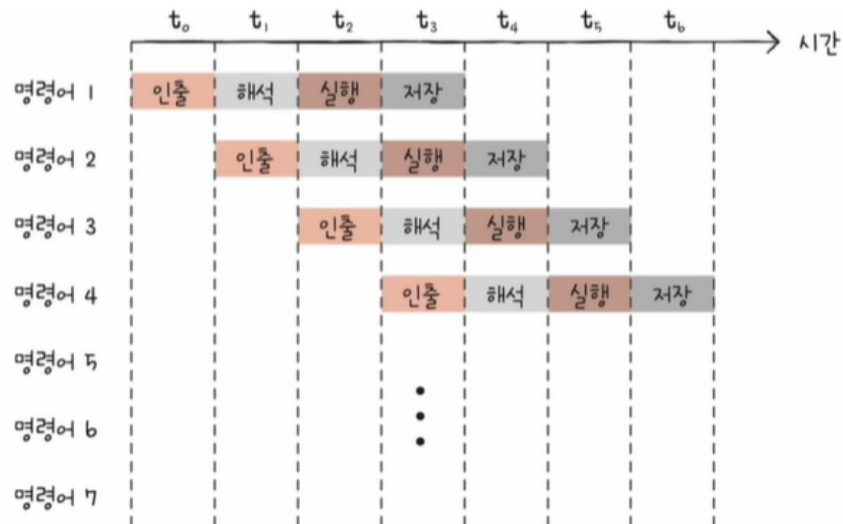


5-2. 명령어 병렬 처리 기법

1. 명령어 파이프라인

: 동시에 여러 개의 명령어를 겹쳐서 실행하는 기법으로, 성능 향상에 기여.

1. 명령어 인출(Instruction Fetch)
2. 명령어 해석(Instruction Decode)
3. 명령어 실행(Execute Instruction)
4. 결과 저장(Write Back)



⇒ 각 단계가 겹치지 않는다면 CPU는 '각 단계를 동시에 실행할 수 있다'

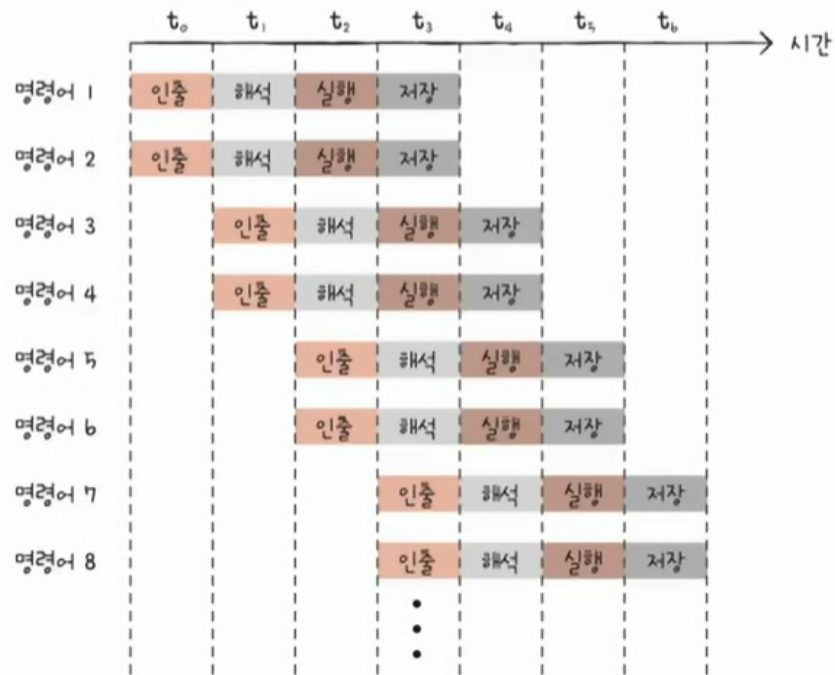
2. 파이프라인 위험

: 명령어 파이프라인이 성능 향상에 실패하는 경우

- 데이터 위험 : **명령어 간의 의존성에 의해 야기**. 모든 명령어를 동시에 처리할 수 없음 (이전 명령어를 끝까지 실행해야만 비로소 실행할 수 있는 경우)
- 제어 위험 : 프로그램 카운터의 갑작스러운 변화
- 구조적 위험 : 서로 다른 명령어가 같은 CPU 부품(ALU, 레지스터)를 쓰려고 할 때

3. 슈퍼스칼라

: CPU 내부에 여러 개의 명령어 파이프라인을 포함한 구조로, 오늘날의 멀티스레드 프로세서



⇒ 이론적으로는 파이프라인 개수에 비례하여 처리 속도가 증가하나, 파이프라인 위험도의 증가로 인하여 파이프라인 개수에 비례해 처리 속도가 증가하지는 않는다.

4. 비순차적 명령어 처리

: 파이프라인의 종단을 방지하기 위해 명령어를 순차적으로 처리하지 않는 명령어 병렬 처리 기법