

컴퓨터 시스템 구조

- 병렬 처리



컴퓨터소프트웨어학과
김병국 교수

학습목표

- 병렬 처리에 대한 개념을 이해한다.
- 병렬 처리 기법을 이해한다.
- 병렬 처리 시 유의 사항을 안다.
- 무어의 법칙과 암달의 법칙을 이해한다.



- 병렬 처리의 개념
- 병렬 처리 시 고려 사항
- 병렬 처리 기법
- 무어의 법칙과 암달의 법칙

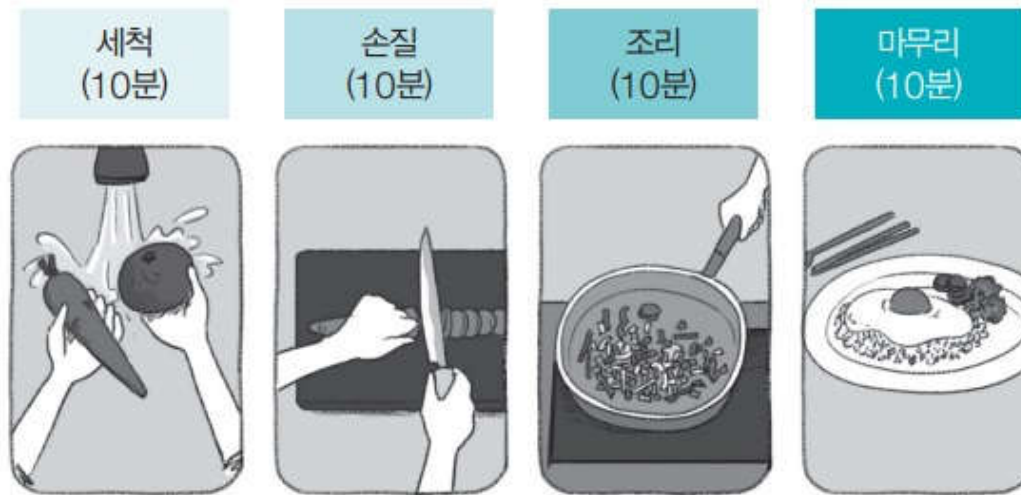


1. 병렬 처리의 개념 (1/3)

□ 병렬 처리(Parallel Processing)

- 동시에 여러 개의 명령을 처리하여 작업의 능률을 올리는 방식

□ 볶음밥 조리 예



【볶음밥 조리 과정】



1. 병렬 처리의 개념 (2/3)

□ 볶음밥 조리의 병렬 처리



【볶음밥 조리의 병렬처리】

□ 파이프라인 기법

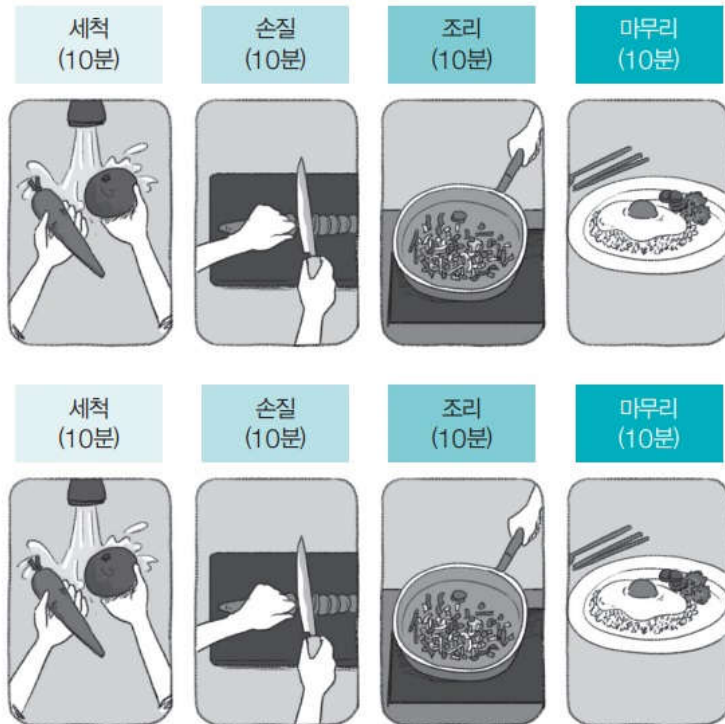
- 하나의 코어에 여러 개의 스레드(Thread)를 이용하는 방식



1. 병렬 처리의 개념 (3/3)

□슈퍼스칼라 기법

- 멀티코어 CPU를 이용해 여러 개(코어 개수)의 작업을 동시에 처리 하는 방식



[조리 시설이 2개인 경우의 병렬처리]



2. 병렬 처리 시 고려 사항 (1/2)

□ 상호 의존성이 없어야 병렬 처리가 가능

- 각 명령이 서로 독립적이고 앞의 결과가 뒤의 명령에 영향을 미치지 않아야 함

□ 각 단계별 처리 시간이 동일해야 함

- 오랜 시간이 걸리는 작업이 발생하면, 이 때문에 전체 작업이 밀림
- 단계별 시간의 차이가 크면 병렬 처리의 효과가 떨어짐



【단계별 시간이 다른 경우의 병렬 처리】



2. 병렬 처리 시 고려 사항 (2/2)

□ 전체 작업 시간을 몇 단계로 나눌지 잘 따져보아야 함

- 병렬 처리의 깊이 N은 동시에 처리할 수 있는 작업의 개수를 의미
- N이 커질수록 동시에 작업할 수 있는 작업의 개수가 많아짐 → 성능 향상
- 작업이 너무 많으면 각 단계마다 작업을 이동하고 새로운 작업을 불러오는 데 시간이 너무 오래 걸림 → 성능 저하



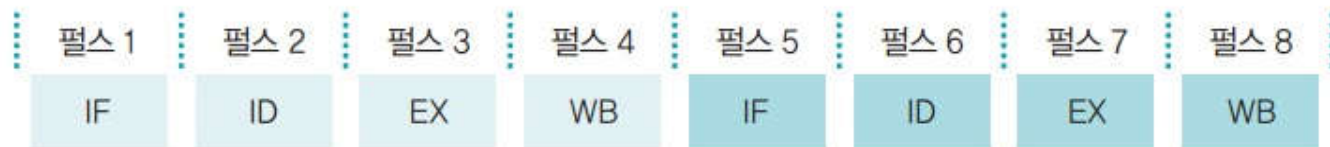
[병렬 처리의 깊이가 2인 경우]



3. 병렬 처리 기법 (1/10)

□ CPU에서 명령어가 실행되는 과정

- ① 명령어 패치(IF) : 다음에 실행할 명령어를 명령어 레지스터에 저장
- ② 명령어 해석(ID) : 명령어 해석
- ③ 실행(EX) : 해석한 결과를 토대로 명령어 실행
- ④ 쓰기(WB) : 실행된 결과를 메모리에 저장



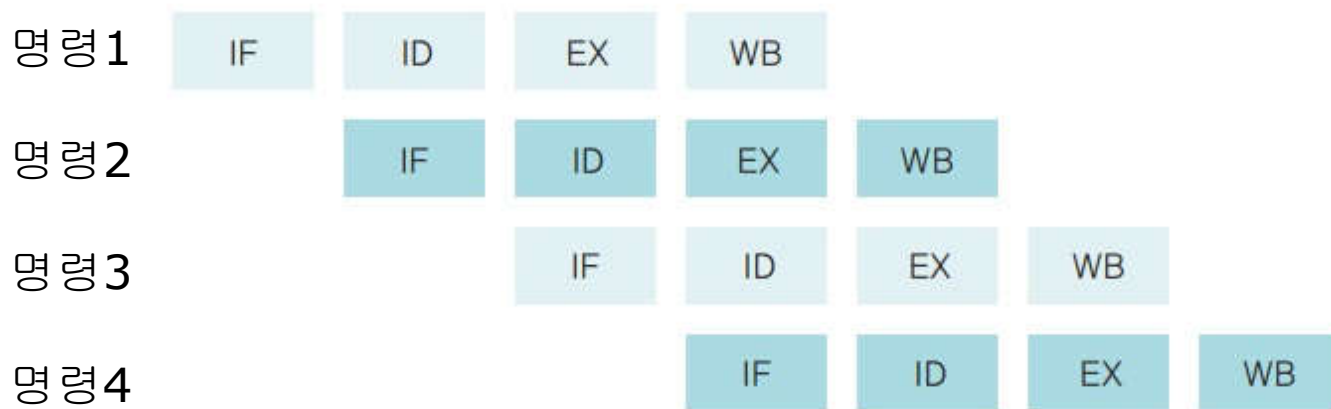
【명령어 처리 단계】



3. 병렬 처리 기법 (2/10)

□파이프라인 기법

- CPU의 사용을 극대화하기 위해 명령을 겹쳐서 실행하는 방법



[파이프라인 기법의 동작 과정]



3. 병렬 처리 기법 (3/10)

□파이프라인의 위험 (1/4)

- 데이터 위험(Data Hazard)
- 제어 위험(Control Hazard)
- 구조 위험(Structural Hazard)



3. 병렬 처리 기법 (4/10)

□파이프라인의 위험 (2/4)

- 데이터 위험(Data Hazard)
 - 데이터의 의존성 때문에 발생하는 문제
 - 의존성을 갖는 데이터가 병렬처리되면 안 됨
 - 해결 방법: 파이프라인의 명령어 단계를 지연하여 해결

$$A = C + D$$

$$E = A * 2$$

【데이터 위험의 예】



3. 병렬 처리 기법 (5/10)

□파이프라인의 위험 (3/4)

- 제어 위험(Control Hazard)
 - 프로그램 카운터 값을 갑자기 변화시켜 발생하는 위험
 - 첫 명령어를 실행 후 다음 문장이 아닌, 다른 문장으로 이동하게 될 경우 후속 명령이 쓸모 없어짐
 - 해결 방법: 분기 예측이나 분기 지연 방법으로 해결



【제어 위험의 예】



3. 병렬 처리 기법 (6/10)

□파이프라인의 위험 (4/4)

- 구조 위험(Structural Hazard)

- 서로 다른 명령어가 같은 자원에 접근하려 할 때 발생하는 문제
- 명령어 A가 레지스터 RX를 사용하고 있는데 병렬 처리되는 명령어 B도 레지스터 RX를 사용해야 한다면 서로 충돌
- 구조 위험은 해결하기 어렵다고 알려져 있음



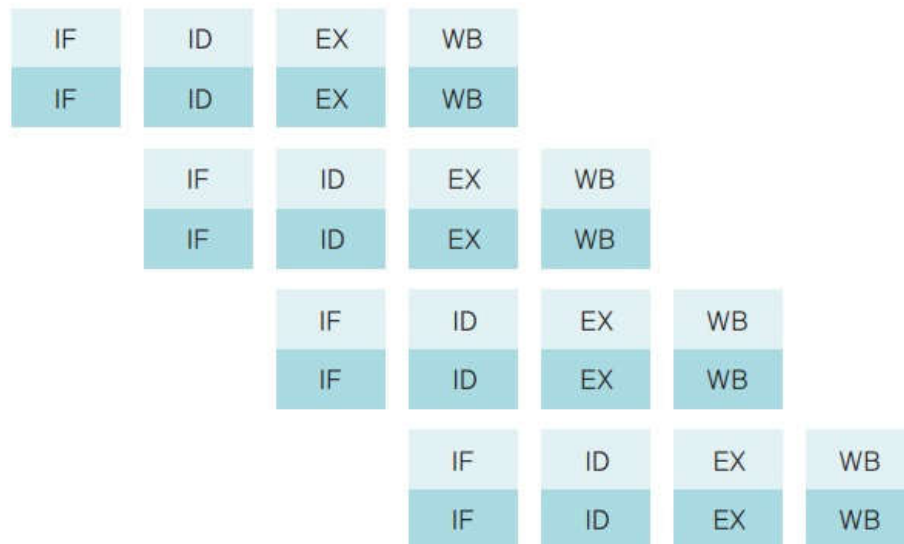
【구조 위험의 예】



3. 병렬 처리 기법 (7/10)

□ 슈퍼스칼라 기법(Super-Scalar)

- 파이프라인을 처리할 수 있는 코어를 여러 개 구성하여 복수의 명령어가 동시에 실행되도록 하는 방식
- 대부분은 파이프라인 기법과 동일하지만 코어를 2개 구성하여 각 단계에서 동시에 실행되는 명령어가 2개라는 점이 다름



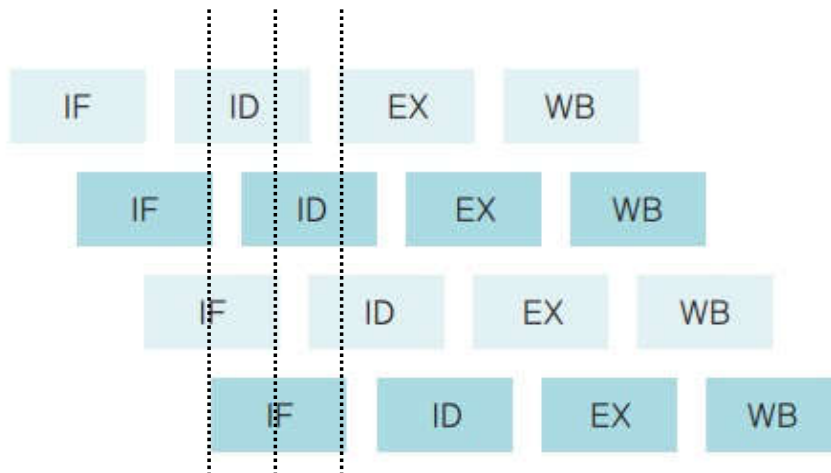
【슈퍼스칼라 기법의 동작 과정】



3. 병렬 처리 기법 (8/10)

□슈퍼파이프라인 기법(Super-Pipeline)

- 파이프라인의 각 단계를 세분하여 한 클록 내에 여러 명령어를 처리
- 한 클록 내에 여러 명령어를 실행하면 다음 명령어가 빠른 시간 안에 시작될 수 있어 병렬 처리 능력이 높아짐



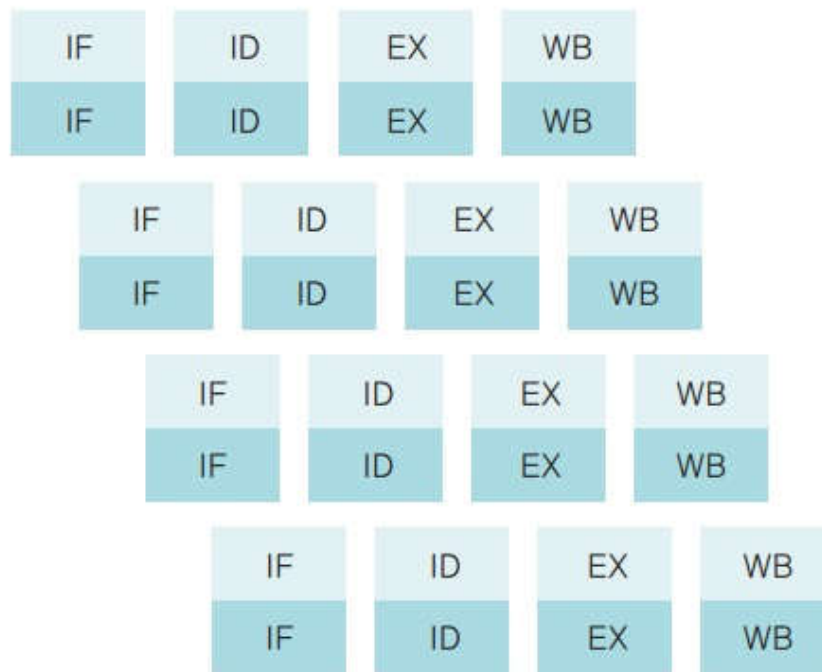
【슈퍼파이프라인 기법의 동작 과정】



3. 병렬 처리 기법 (9/10)

□슈퍼파이프라인 슈퍼스칼라 기법(Super-Pipelined Super-Scalar)

- 슈퍼파이프라인 기법을 여러 개의 코어에서 동시에 수행하는 방식



【슈퍼파이프라인 슈퍼스칼라 기법의 동작 과정】



3. 병렬 처리 기법 (10/10)

□ VLIW 기법(Very Long Instruction Word)

- CPU가 병렬 처리를 지원하지 않을 경우
- 소프트웨어적으로 병렬 처리를 하는 방법
- 동시에 수행할 수 있는 명령어들을 컴파일러가 추출하고 하나의 명령어로 실행
 - 일반적인 병렬 처리 기법들에 비해 동시에 처리하는 명령어의 개수가 적음
 - 컴파일 시 관련 처리 명령이 생성됨



4. CISC와 RISC

□ CISC

- Complex Instruction Set Computer
- 4싸이클에 1개의 명령문을 처리함(가변)
- 명령어가 많음
- 메모리 참조 연산이 상대적으로 많음
- 레지스터 개수가 적음

□ RISC

- Reduced Instruction Set Computer
- 4싸이클에 4개의 명령문을 처리함(고정)
- 컴파일러의 최적화가 필요함
- Load/Store 방식으로 메모리 참조 연산의 개수가 적음
- 레지스터 개수가 많음



5. 무어의 법칙과 암달의 법칙

□ 무어의 법칙(Moore's law)

- CPU의 속도가 24개월마다 2(1.8)배 빨라진다는 내용
- 초기의 CPU에만 적용되며 지금은 그렇지 않음

□ 암달의 법칙(Amdahl's law)

- 컴퓨터 시스템의 일부를 개선할 때 전체 시스템에 미치는 영향과의 관계를 수식으로 나타낸 법칙
- 이 법칙에 따르면 주변장치의 향상 없이 CPU의 속도를 2GHz에서 4GHz로 늘리더라도 컴퓨터의 성능이 2배 빨라지지 않음



수고하셨습니다.

