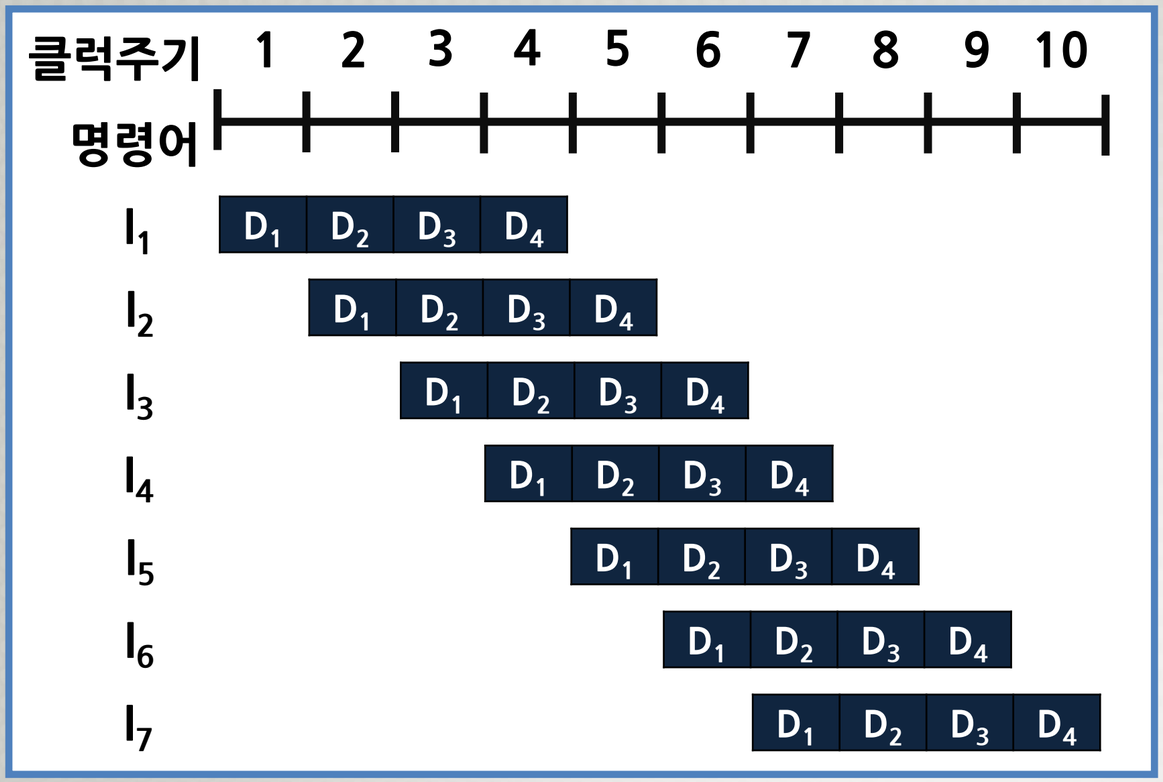
명령어 파이프라이닝(Instruction Pipelining)

CPU의 성능향상을 위하여 명령어를 실행하는데 사용되는 하드웨어를 여러 개의 독립적인 단계로 분할하여 동시에 처리하는 기술을 말한다.

명령어 파이프라이닝(Instruction Pipelining)

기본 개념



- 명령어를 균등하게 4단계로 분리하여 단계 마다 1개 클럭 동안에 실행될 수 있도록 구성한다.

  명령어 : D_1 D_2 D_3 D_4

- 실행 완료 = $D_1 + D_2 + D_3 + D_4$

명령어 파이프라이닝(Instruction Pipelining)

파이프라인에 의한 전체 명령어 실행 시간

파이프라인 단계 수 : K 단계
전체 실행될 명령어들의 수 : N 개
각 단계의 소요시간 : 1 클럭 주기
파이프라인 기술이 적용되지 않은 실행시간 : T_{np}
파이프라인 기술이 적용된 실행시간 : T_p
속도 향상(Speedup) : Sp

- 파이프라이닝 기법이 적용되지 않은 경우의 전체 실행시간

$$T_{np} = K \times N$$

- 파이프라이닝 기법이 적용된 경우의 전체 실행시간

$$T_p = K + (N - 1)$$

- 파이프라이닝 기법에 따른 속도향상(Speedup)

$$SP = \lim_{n \to \infty} \frac{T_{np}}{T_p} = \lim_{n \to \infty} \frac{K \times N}{K + (N - 1)} = K$$

명령어 파이프라이닝(Instruction Pipelining)

- 파이프라인에 의한 전체 명령어 실행 시간

→ 따라서, K-단계의 파이프라이닝 기법을 적용하였을 때, **이론적인 성능향상은 실행시간에 있어서 단계 수와 동일한 K배 만큼 빠르다**는 결론을 얻을 수 있다.

명령어 파이프라이닝(Instruction Pipelining)

예제

명령어 1000개를 아래와 같은 조건으로 실행하는데 있어서
1) 파이프라이닝 기법을 사용하지 않을 때와
2) 파이프라이닝 기법을 사용할 때, 각각의 경우에
전체 명령어들의 실행 시간과 3) 속도향상(Sp)을 계산하시오.

- CPU 클럭 : 1GHz
- 명령어 실행 : 4 클럭 소요
- 명령어 : 4-단계 균등분할

명령어 파이프라이닝(Instruction Pipelining)

풀이

단계 수 : K = 4, 명령어의 수 : N = 1000,
명령어의 실행시간 = 4 nsec
CPU 클럭 주기 : 1sec ÷ 10^9 Hz = 10^{-9} sec = 1 nsec

1) T_{np} = K×N = 4 nsec ×1000 개 = 4 μsec
2) T_p = K + (N - 1) = 4 nsec + (1000 -1) = 1.003 μsec
3) Sp = 4 ÷ 1.003 = 3.988 배 ≒ 4 배 → 단계 수를 의미한다.

4-단계의 명령어 파이프라이닝 구성

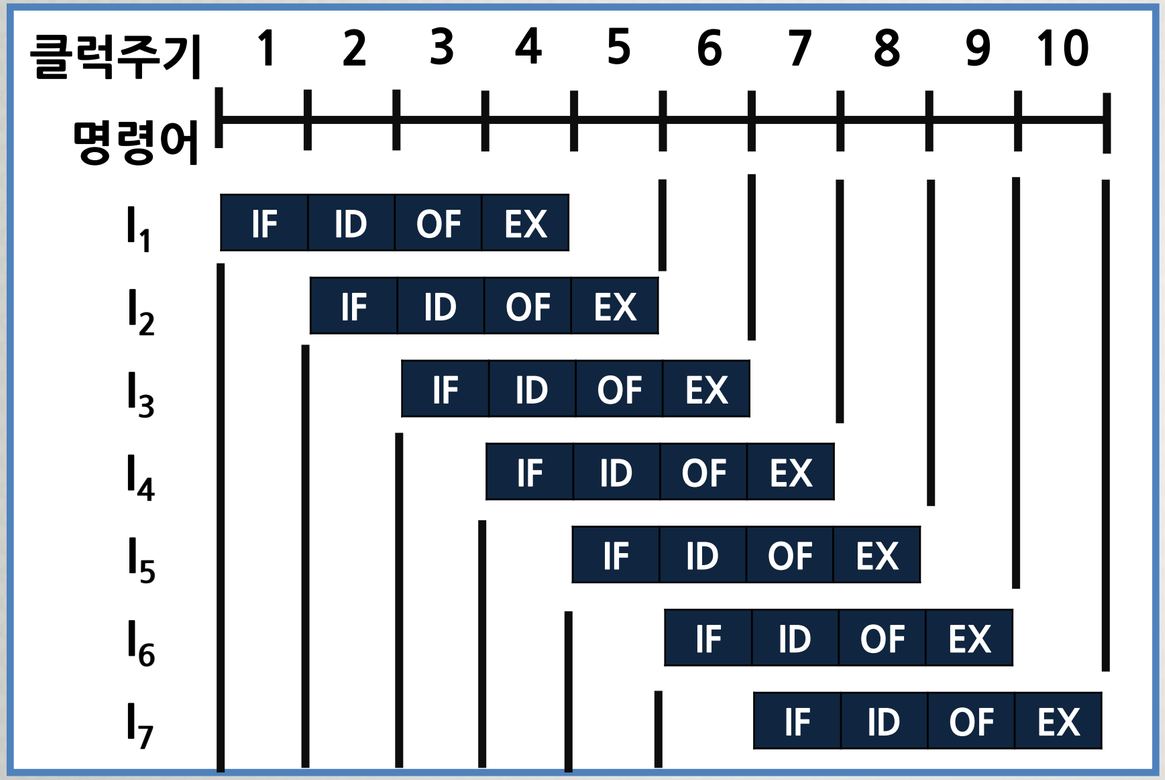
D1 : 명령어 인출(IF)	• 다음에 실행될 명령어를 기억장치로부터 인출하는 단계
D2 : 명령어 해독(ID)	• 제어 유니트에서 해독기(Decoder)를 이용하여 명령어의 op-code 를 해석하는 단계
D3 : 오퍼랜드 인출(OF)	• 연산에 필요한 오퍼랜드의 데이터를 기억장치로부터 인출하는 단계
D4 : 실행(EX)	• 해독기에서 정해진 연산을 수행하는 단계

- 단계 분할 조건
  - 각 단계의 실행 시간은 동일해야 한다.
  - 분할된 단계의 가장 많이 소요되는 시간과 동일하게 설정되어야 한다.
  - 비효율적이다.

4-단계의 명령어 파이프라이닝 구성

4-단계별 실행과 흐름도

명령어 인출(IF) + 명령어 해독(ID) + 오퍼랜드 인출(OF) + 실행(EX) = 실행 결과



명령어 파이프라이닝 실행 장애(Hazard)

구조적 장애(Structural Hazards)

- 계속되는 명령어 수행에 있어서 한 명령어는 메모리부터 명령어를 가져오는 단계에 있고 또 다른 명령어는 메모리로부터 필요한 데이터를 가져오는 단계에 있다면, 메모리가 하나인 경우에 두 개의 서로 다른 명령어의 메모리 접근 충돌 때문에 동시에 처리할 수 없는 구조적 해저드가 발생한다.

명령어 파이프라이닝 실행 장애(Hazard)

데이터 장애(Data Hazards)

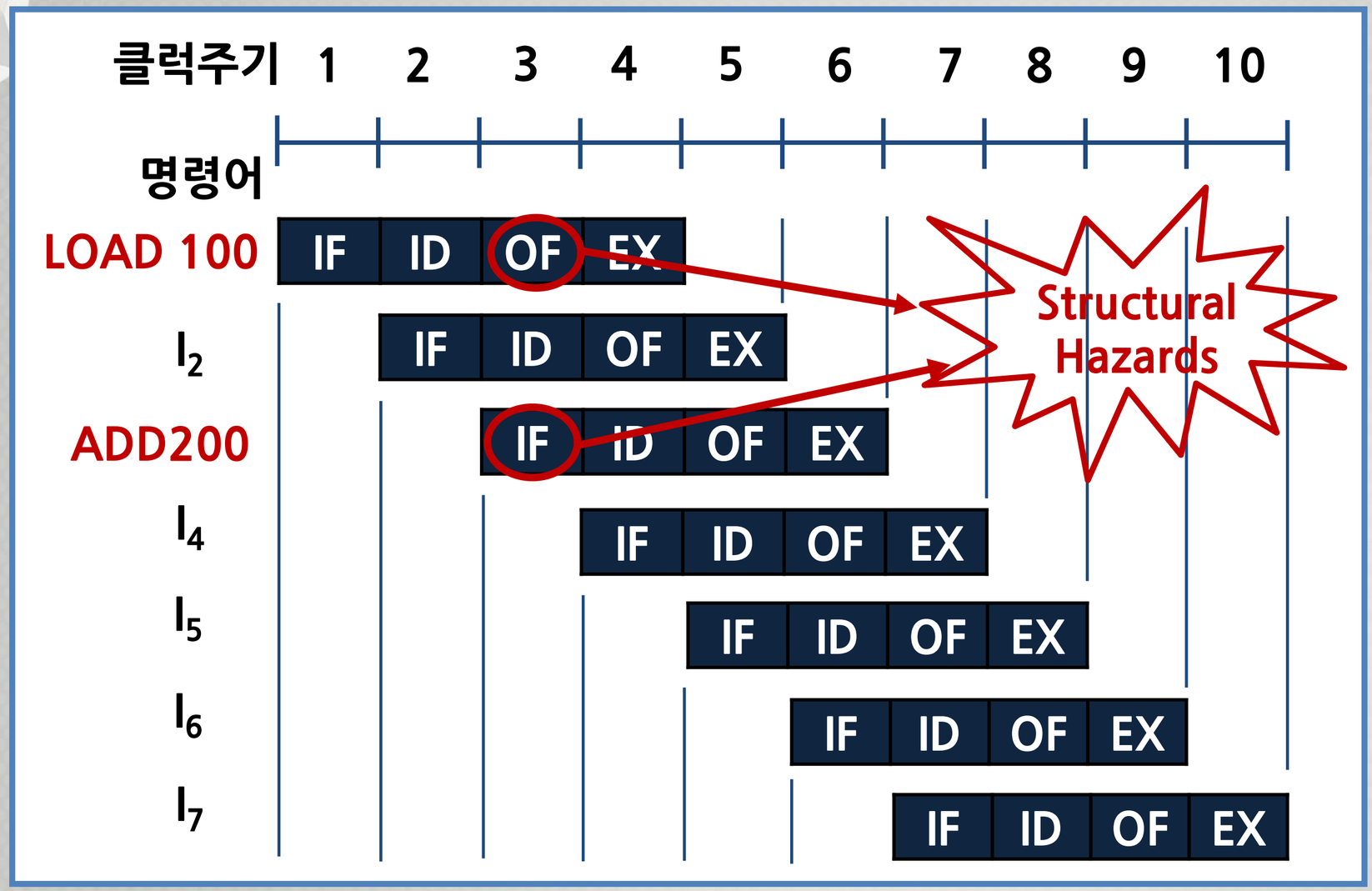
- 한 명령어 수행 결과가 다른 명령어의 연산에 사용될 때 파이프라이닝의 단계가 지연되어야만 하는 경우에 발생하는 것을 의미하는데, 이는 명령어 처리 결과 사이에 데이터의 종속성이 존재할 때 데이터 해저드가 발생한다.

제어 장애(Control Hazards)

- 다른 명령어들이 실행들이 앞선 명령어의 결과값에 따라 필요하지 않을 때 발생한다. 주로 분기 명령어에서 이러한 경우가 발생한다.

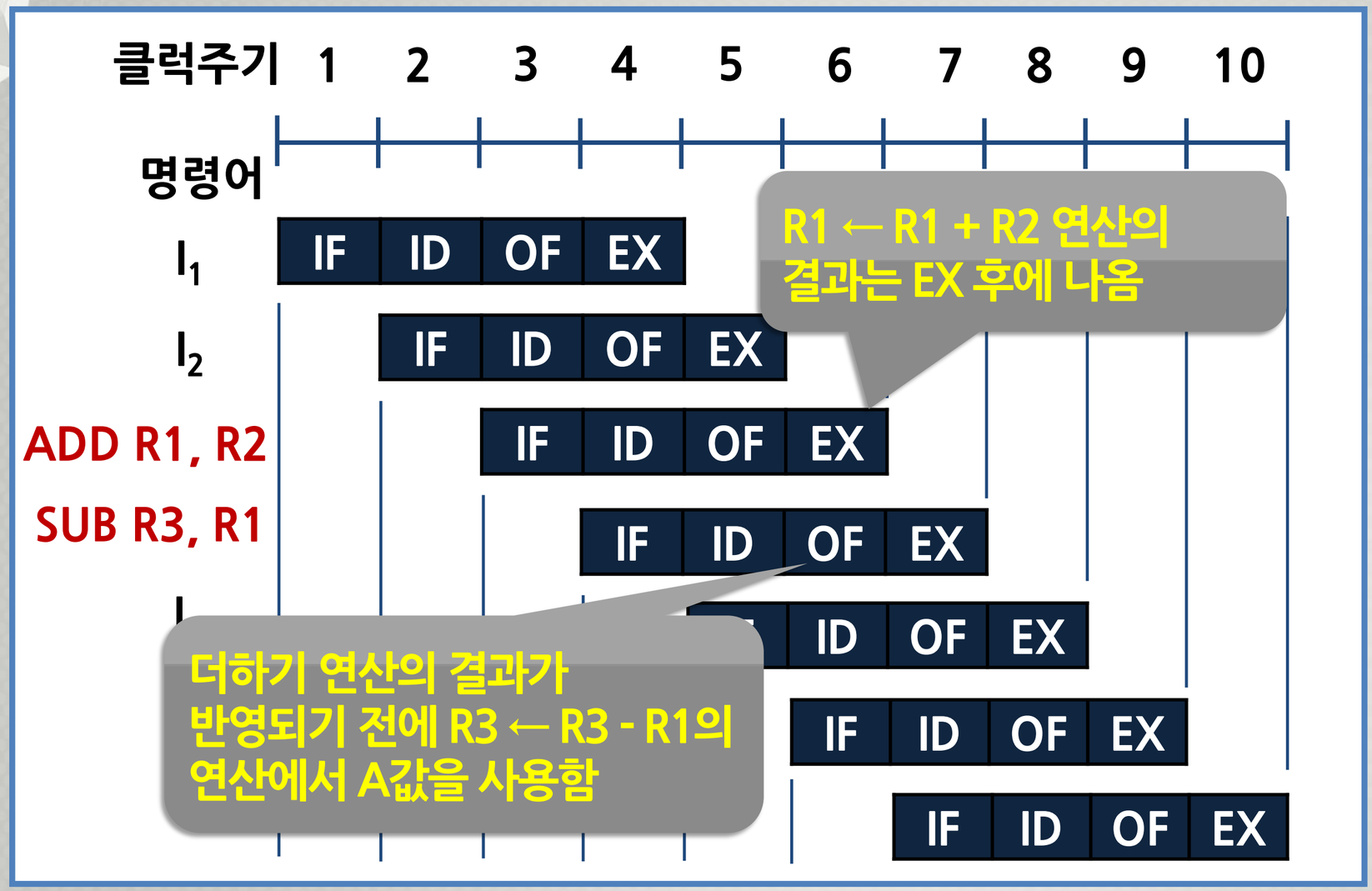
명령어 파이프라이닝 실행 장애(Hazard)

구조적 장애(Structural Hazards)



명령어 파이프라이닝 실행 장애(Hazard)

데이터 장애(Data Hazards)



명령어 파이프라이닝 실행 장애(Hazard)

제어 장애(Control Hazards)