7주 3강

캐시기억장치의 구조

적중률(Hit Ratio)

1. 적중률은 캐시기억장치를 가진 컴퓨터의 성능을 나타내는 척도로 적중률이 높을수록 속도가 향상된다.

$$\text{적중률} = \frac{\text{적중 수}}{\text{전체 메모리 참조 횟수}}$$

2. 기억장치에서 데이터를 인출하는데 소요되는 평균 기억장치 접근시간 $T_{average}$

$$T_{average} = H_{hit\_ratio} \times T_{cache} + (1 - H_{hit\_ratio}) \times T_{main}$$

$T_{average}$ = 평균 기억장치 접근 시간　　T_{main} = 주기억장치 접근 시간

T_{cache} = 캐시기억장치 접근 시간　　H_{hit_ratio} = 적중률

- 캐시기억장치 평균 접근시간과 주기억장치 평균 접근시간에 대한 합한 것이 평균 기억장치 접근시간

- 캐시기억장치 평균 접근시간
  - 캐시기억장치 접근시간 T_{cache}와 적중률 H_{hit_ratio}와의 곱으로 얻어진다.

- 주기억장치 평균 접근시간
  - 주기억장치 접근시간 T_{main}과 실패율 $(1-_{hit_ratio})$와의 곱으로 얻어진다.
  - 여기서 실패율은 곧 주기억장치에 접근하는 율을 나타낸다.

평균 기억장치 접근시간 $T_{average}$ 계산 예

1 T_{cache} = 50ns, T_{main} = 400ns일 때, 적중률을 증가시키면서 기억장치 접근시간을 계산하면

- 적중률 70%의 경우: $T_{average}$ = 0.7 x 50ns + 0.3 x 400ns = 155ns
- 적중률 80%의 경우: $T_{average}$ = 0.8 x 50ns + 0.2 x 400ns = 120ns
- 적중률 90%의 경우: $T_{average}$ = 0.9 x 50ns + 0.1 x 400ns = 85ns
- 적중률 95%의 경우: $T_{average}$ = 0.95 x 50ns + 0.05 x 400ns = 67.5ns
- 적중률 99%의 경우: $T_{average}$ = 0.99 x 50ns + 0.01 x 400ns = 53.5ns

2 캐시기억장치의 적중률이 상승할 경우

- 평균 기억장치 접근시간은 캐시기억장치 접근시간에 근접하게 되어 컴퓨터의 처리 속도의 성능 향상을 가져온다.

캐시기억장치의 구조

1 온-칩(On-chip) 캐시기억장치

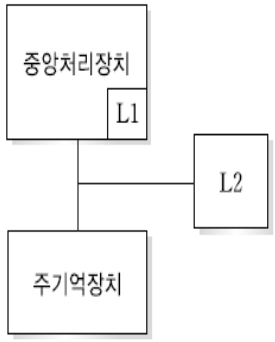
- 집적회로(Integrate Circuit)의 기술 발달로 캐시기억장치를 CPU의 내부에 포함시키는 것
- CPU의 외부 활동을 줄이고 실행 시간을 가속시켜 전체 시스템의 성능을 높여준다.

2 오프-칩(Off-Chip) 캐시기억장치 또는 외부 캐시기억장치

- 일반적인 형태로 캐시기억장치가 CPU 외부에 위치
- 외부 버스를 사용해서 CPU에 접근

단일 프로세서에서 캐시기억장치의 구조

계층적 캐시(Hierarchical Cache)기억장치



- 온-칩 캐시를 1차 캐시(L1)로 사용하고 칩 외부에 더 큰 용량의 오프-칩 캐시를 2차 캐시(L2)로 설치하는 방식
- 온-칩 캐시기억장치 L1의 크기는 제한되지만, L2의 크기는 상대적으로 L1보다 더 많은 용량을 가질 수 있다.
- L2는 주기억장치의 일부 내용을 저장, L1은 L2 내용의 일부를 저장한다. 따라서 L2는 L1의 모든 정보를 포함.
- 먼저 L1을 검사, L1에 원하는 정보가 존재하지 않으면 L2를 검사, L2에도 원하는 정보가 존재하지 않으면 주기억장치를 조사
- L1 캐시의 속도는 빠르지만 용량이 작기 때문에 적중률이 L2에 비해 낮다.

캐시기억장치의 통합과 분리

1 통합 캐시 형태

- 온-칩 캐시는 데이터와 명령어를 모두 저장하는 통합 캐시 형태
- 통합 캐시는 명령어와 데이터 간의 균형을 자동적으로 유지해주기 때문에 분리 캐시보다 적중률이 더 높은 장점이 있다.

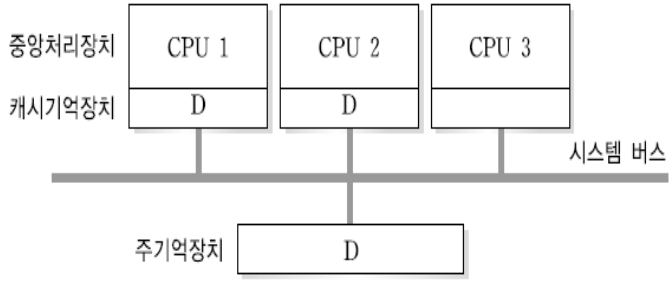
2 분리 캐시 형태

- 최신 캐시의 설계는 용도별 또는 기능별 분리 캐시 경향
- 분리 캐시는 명령어만 저장하는 명령어 캐시와 데이터만 저장하는 데이터 캐시로 분리하여 두 개의 온-칩 캐시를 두는 형태다.
- 여러 개의 명령어들이 동시에 실행되는 고성능 프로세서에서는 이러한 경향이 뚜렷하다.
- 분리 캐시의 장점은 명령어 인출과 명령어 실행 간 캐시의 충돌이 발생하지 않는다는 것이다.

멀티 프로세서의 캐시기억장치 구조

1 최신의 컴퓨터 시스템은 여러 개의 중앙처리장치(CPU)를 장착하여 처리성능을 향상시키고 있는데, 이것을 멀티 프로세서 시스템이라고 한다.

2 시스템 버스에 온-칩 캐시의 CPU 3개가 연결된 멀티 프로세서 시스템

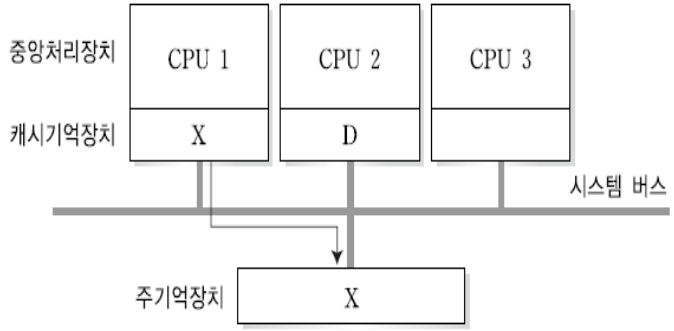


- 멀티 프로세서 시스템에서는 주기억장치와 각 중앙처리장치 내의 캐시기억장치들 사이에서 데이터의 불일치 현상이 발생
- 이러한 데이터의 불일치 현상은 프로그램이 올바르게 동작하지 않는 원인이 된다.

즉시 쓰기방식에서의 데이터 불일치 상태

멀티 프로세서 시스템에 즉시 쓰기 정책

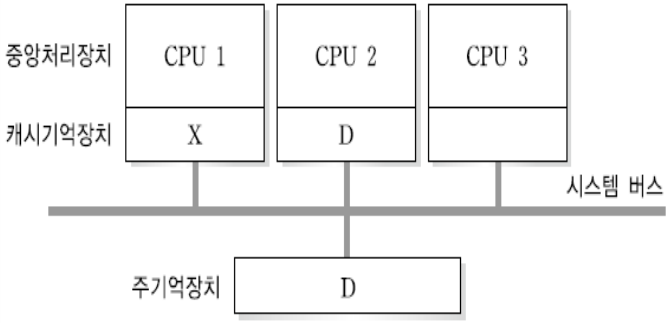
- CPU 1과 CPU 2는 주기억장치에서 D라는 데이터를 읽어온다. 이렇게 되면 CPU 1, CPU 2, 주기억장치는 D라는 동일한 데이터를 갖게 된다.
- CPU 1이 프로그램을 실행하여 D라는 데이터를 X로 수정하게 되면 CPU 1에 속한 캐시기억장치는 데이터를 X로 변경하고 즉시 쓰기 정책에 따라 주기억장치에도 수정된 데이터인 X를 저장하게 된다.
- 이 경우 CPU 1에 속한 캐시기억장치와 주기억장치의 데이터는 X로 수정이 되지만 CPU 2에 속한 캐시기억장치는 D라는 데이터로 남아있게 되기 때문에 데이터의 불일치가 발생하게 된다.



나중 쓰기 방식에서의 데이터 불일치 상태

멀티 프로세서 시스템에 나중 쓰기 정책

- 주기억장치에서 D라는 데이터를 CPU 1과 CPU 2의 캐시가 읽어와서, CPU 1, CPU 2, 주기억장치는 D라는 동일한 데이터를 갖게 된다.
- CPU 1이 프로그램을 실행하여 D라는 데이터를 X로 수정하게 되면 나중 쓰기 정책에 의해 CPU 1에 속한 캐시기억장치는 수정된 데이터 X가 저장된다.
- 주기억장치와 CPU 2에 속한 캐시기억장치는 D라는 데이터로 남아있게 되기 때문에 데이터의 불일치가 발생하게 된다.



캐시기억장치의 데이터 일관성 유지 방법

1 공유 캐시기억장치를 사용하는 방법

- 다수의 프로세서가 하나의 캐시기억장치만을 공유
- 캐시의 데이터들이 항상 일관성 있게 유지하는 장점이 있으나 다중 프로세서가 동시에 캐시에 접근하면 프로세서들 간의 충돌이 발생
- 온-칩 캐시기억장치의 경우 CPU의 외부 활동을 줄여 실행 시간을 가속시키고 전체 시스템 성능을 높이는 원칙에 위배되는 단점을 가진다.

2 공유 변수는 캐시기억장치에 저장하지 않는 방법

- 수정 가능한 데이터는 캐시기억장치에 저장하지 않는 방법
- 수정될 데이터는 캐시에 저장하지 않고 주기억장치에 바로 저장
- 캐시기억장치에 저장 가능한지 불가능 한지를 사용자가 선택하여 선언해 주어야 하는 단점이 있다.

3 버스 감시 시스템을 사용하는 방법

- 감시 기능을 가진 장비를 시스템 버스상에 추가로 설치하는 방법
- 한 캐시가 데이터를 수정하면 그 정보를 다른 캐시와 주기억장치에 전달.
- 시스템 버스에 통신량이 증가하는 단점이 있다.

다음 시간

8주. 보조기억장치