

컴퓨터구조 정리노트 #5 - 시험 공부
ICT융합공학부 202204010 공성택

6장 이후로는 따로 필기하며 외우기보다는 큰 틀만 잡고 강의 자료 위주 이해 암기했습니다.

6장

캐시 기억 장치

목적: CPU와 주기억 장치 속도 차이로 인한 대기 시간 최소화

CPU와 주기억 장치 사이 위치

특징: 주기억 장치보다 액세스 높은 칩
가벼운 공간 때문에 용량 적음
물리

흐름: CPU가 기억 장치 접근 전 캐시에 있는지 확인 후
있으면 정보 읽고, 없으면 캐시에 정보 불러와 읽음

캐시 히트: CPU가 원하는 데이터가 캐시에 존재

"미스: " " 존재 X
⇒ 주기억 장치로부터 불러옴

히트율: $H = \frac{\text{캐시 히트 횟수}}{\text{전체 기억 장치 접근 횟수}}$

미스율: $1 - H$

평균 기억 장치 액세스 시간 (T_a)

$$\begin{aligned} T_c: \text{캐시 액세스 시간} \quad T_a &= H \times T_c + (1-H) \times (T_c + T_m) \approx T_m \\ T_m: \text{주기억 액세스 시간} \quad &\approx H \times T_c + (1-H) \times T_m \end{aligned}$$

참고 지역성
공간적 지역성: 기억 장치 인접 데이터가 액세스 가능성 높다
시간적 지역성: 최근 액세스 데이터가 액세스 가능성 높다

캐시 설계 목표: 히트율 극대화

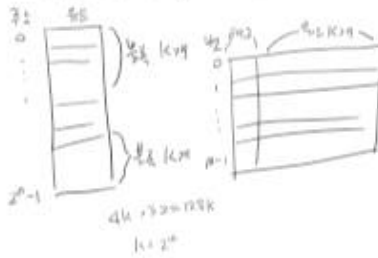
캐시 정보 읽는 시간 최소화

캐시 정보 가져오는 시간 최소화

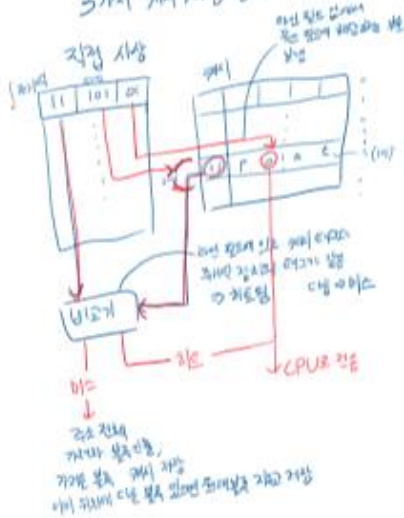
캐시 내용 변경 주기억 장치 갱신 시간 최소화

- 캐시 설계 요소

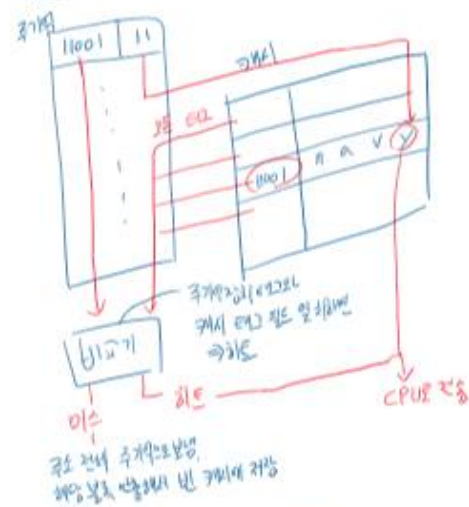
1. 캐시 용량: 커적수록 히트율↑, 비용↑, 액세스시간↑
공간 제한 있음
2. 사상 방식: 주계급 접근방식이 한 캐시 라인의 데이터결정
본래: 주계급 \rightarrow 캐시 이동하는 정도↓
주계급: 2^n 라인 \rightarrow 블록 \rightarrow 블록 수 = $2^n / K$
라인: 캐시에서 블록의 저장 감소
데이터: 라인에 적지된 블록 구분



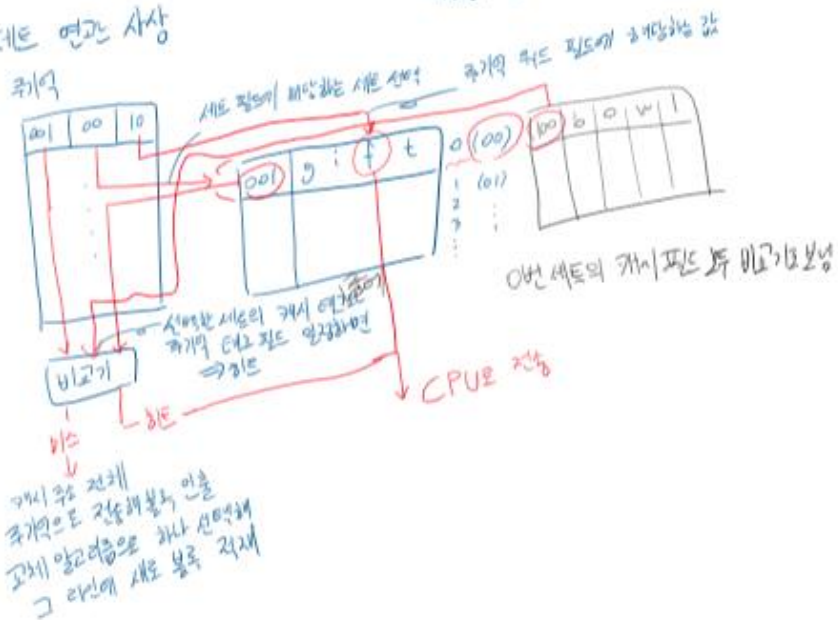
3가지 캐시 사상 정리



오래전 변환 사상



세트 연관 사상



직접 사상 : 2) 하드웨어 간단, 비용 ↓
 단) 각 주기마다 불록 교체 가능한 캐시 라인 한 개만
 그렇기 때문에 주기마다 바깥으로 캐시 라인 ↓

- 직접 사상 : 교체 행위에 관련된 비용 매우 적은 것
- 관련 연관 사상 : 주기당 블록 교체 시 연관 라인들은 적게
- 세트 연관 사상 : 주기당 블록 그룹에 하나씩 캐시 세트 블록, 세트에는 두개씩 라인 적게

사상 방식 비교	연관성	태그 연관 작성	캐시 라인	교체 방법
직접	단순	X	↓	필수
관련 연관	복잡	연관	↑	필수
세트 연관	중간	중간	중간	필수

3. 교체 알고리즘

LRU : 사용되지 않은 채로 가장 오래 있던 블록 교체

FIFO : 캐시에 적재된 지 가장 오래된 블록 교체

LFU : 참조된 횟수가 가장 적은 블록 교체

Random : 사용 횟수 고려 X, 캐시 라인 중 임의 교체

4. 쓰기 정책 : 캐시 블록 변경 시 내용을 주기에 갱신하는 시기, 방법

Write through : 모든 동작이 캐시 뿐만 아니라 주기로도 동시에 수행

장 : 캐시, 주기당 블록 내용 항상 같음

단 : 모든 쓰기 동작이 둘 다 실행되므로 쓰기 시간 길

Write back : 캐시에서 변경돼도 주기에는 갱신 X

장 : 쓰기 동작 횟수 초기화, 쓰기 시간 짧음

단 : 캐시와 주기당 내용 서로 다름

→ 블록 교체 시 캐시 상태 확인 후 주기에 갱신 동작 필요
 따라서 각 캐시 라인이 상태 비트 갖고 있어야 함.

5. 라인 크기

블록 크기에 따른 특성 : 블록 크기 커질수록 캐시에 들어올 수 있는 블록 수 ↓

블록 수 적으면 자극 교차

블록 크기 커질수록 원하는 워드에서 떨어져 있는 워드도 같이 읽음

→ 그것들은 사용 가능성 ↓ ∴ 효율 ↓

∴ 8 ~ 32 바이트가 최적

6. 캐시 속

온 칩 캐시 : CPU 칩 내에 캐시 포함 \Rightarrow 액세스 시간 \downarrow

계층적 캐시 : 온 칩을 1차 캐시, CPU 외부에 더 큰 2차 캐시

분리 캐시 : 캐시를 명령어와 데이터 캐시로 분리

- 계층적 캐시에서 히트율

L1 검색 후 원하는 정보 없으면 L2, L2에도 없으면 주기억

히트율 : $L1 < L2$

속도 : $L1 > L2$

평균 기억 장치 액세스 시간 (T_a)

$$T_a = H_1 \times T_{L1} + (H_2 - H_1) \times T_{L2} + (1 - H_2) T_m$$

T_{L1}, H_1 : L1 캐시의 액세스 시간, 히트율

T_{L2}, H_2 : L2 "

T_m : 주기억 장치 액세스 시간

H_2 는 전체 기억 장치 액세스에 대한 L2의 히트율

통합 캐시

온 칩 캐시 처음 구현했을 때, 명령어와 데이터 양의 캐시에 저장

분리 캐시

명령어/데이터 캐시로 분리

명령어 인출/실행 무난한 액세스 횟수 제거

과잉용 포인터라인에서 사용

가상 기억 장치 : 하드 디스크처럼 용량이 큰 보조 기억 장치를 주기억 장치처럼 사용

주소 매핑 : CPU가 참조하는 가상 주소를 주기억 장치 주소로 변환

페이지에 의한 매핑

페이지 : 주소 공간을 고정 크기로 나눔

블록 : 메모리 공간을 고정 크기로 나눔

페이지 오류 : 가상한 페이지가 주기억 장치에 존재 X

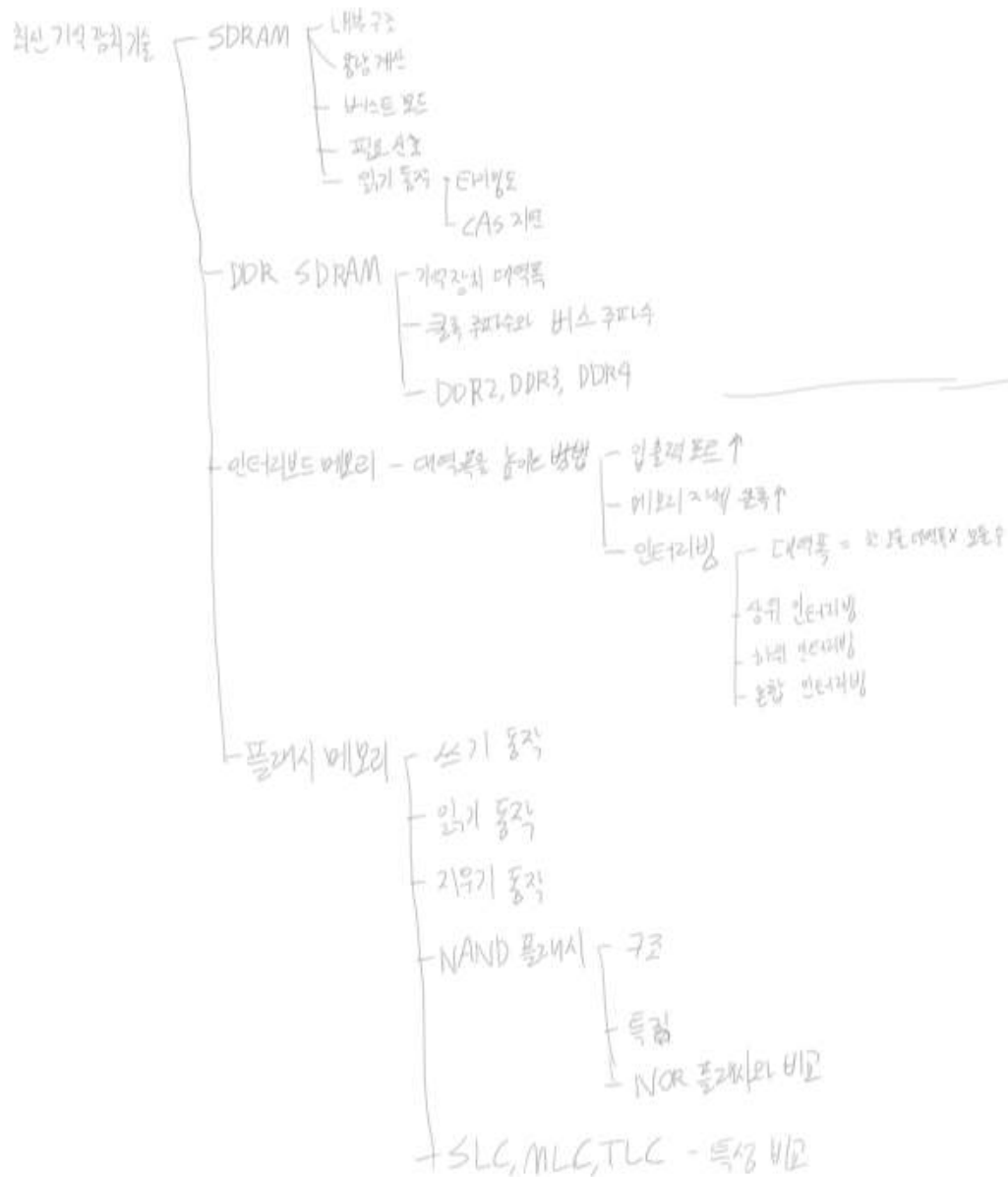
스왑잉 : 페이지 오류 처리 방법

페이지에 대한 기억 장치 매핑표를 가지고 페이지를 블록으로 변환

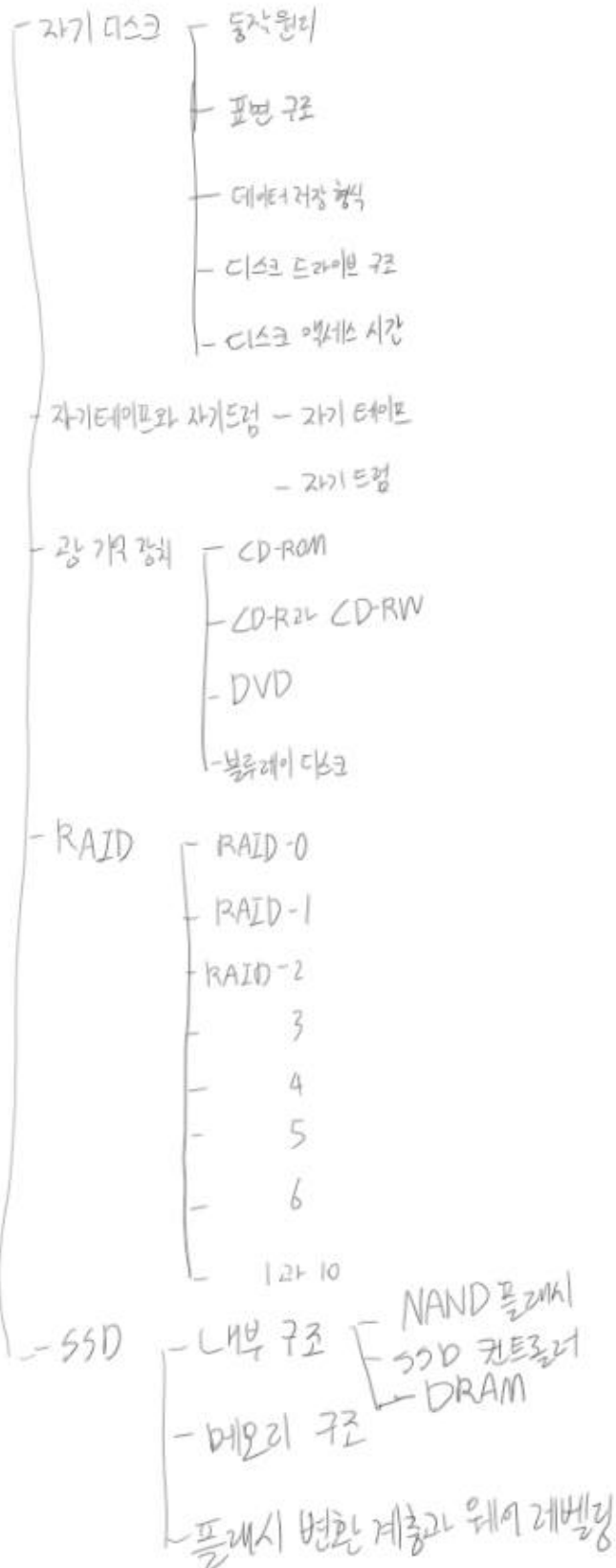
기억 장치 매핑표 : 페이지 번호에서 블록 번호로 변환하는 방법 됨

경우 내트가 이면 2개 페이지 주기억 장치

권장함 의미



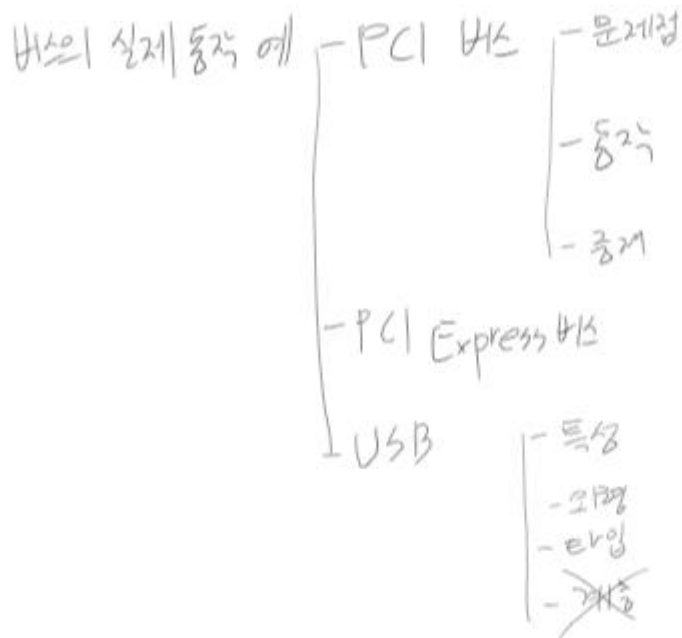
7장 보조 기억장치



8장 버스와 입출력

CPU 칩과 시스템 버스

- 시스템 버스
 - CPU 칩
 - 버스의 크기
 - 다양한 핀 - 인터럽트 핀
 - 버스 중재 핀
- 시스템 버스
 - 메모리 버스 / I/O 버스가 각각 1개씩 있는 시스템
 - 일반적인 버스 구조
 - 버스 프로토콜
 - 다양한 버스
 - 버스 동작
 - 버스 폭
 - 버스 대역폭 확장
 - 제어 버스
 - 버스 타이밍
 - 동기식 버스
 - 비동기식 버스
- 버스 중재
 - 병렬 버스 중재 방식
 - 직렬 버스 중재 방식
 - 토큰 버스 중재 방식
 - 분산 버스 중재 방식
- 인터럽트 I/O



입출력 장치 인터페이스 - 피싱

- 프로그램 입출력 - 분리 입출력
- 메모리 사상 입출력

- 인터럽트 I/O

- DMA

- 입출력 인터페이스
실제 예

9장 인터럽트

인터럽트 개념과 종류 - 개념 : 프로그램 실행 중 예기치 않은 상황 발생, 실행 중인 작업 중단 후 상황 처리 후 복귀 후 계속 처리

- 종류 - 외부 인터럽트
- 내부 인터럽트
- 소프트웨어 인터럽트

우선순위 : 외부 > 내부 > 소프트웨어

1. 동작원리 - 동작 과정

- 프로그램 상태 보존
 - 인터럽트 서비스 루틴에 저장
 - 주기의 장치의 특정 영역인 스택에 저장
 - 별도의 레지스터 세트에 저장
 - 인터럽트 벡터에 저장

- 인터럽트 요청선 연결 방법

인터럽트 처리 과정 - 소프트웨어 이용 (동작) - 인터럽트 요청

- 우선 판별

- 우선 순위

- 하드웨어 이용 (데이터 확인) - 인터럽트 요청

- 인터럽트 벡터

- 우선 순위

- 다중 인터럽트 요청선 - 인터럽트 요청

- 우선 순위

- 인터럽트 게이트

- 인터럽트 레지스터

- 마스크 레지스터

- 종류 - PIC

- APIC

1028 병렬 컴퓨터 구조

병렬 처리 동시 배정

- 칩 단위 병렬 처리
 - 명령 레벨 병렬 처리
 - 온칩 멀티스레딩 - 하이퍼스레딩
 - 심층-칩 다중 프로세서
 - 동중 멀티-코어
 - 이중 멀티-코어

- 공유 메모리 다중 프로세서 - 폴린의 컴퓨터 분류법
 - SISD
 - SIMD
 - MISD
 - MIMD
 - 공유 메모리 시스템
 - UMA
 - NUMA
 - COMA

- 공유 메모리 다중 프로세서
 - UMA 다중 프로세서
 - NUMA 다중 프로세서
 - COMA ' '

다중 컴퓨터

- [MPP
 - PC, 워크스테이션 or 서버로 구성

- [상호 연결망
 - MPP
 - 클러스터 컴퓨팅
 - 다중 컴퓨터의 통신 소프트웨어
 - 스케줄링