ARM Processor Features

ARM processor 구조 : co-processor : (MMU : cpu 안에 들어있진 않음) 캐시는 아니고 메모린데 아주 빠름

CP15 아주중요 :

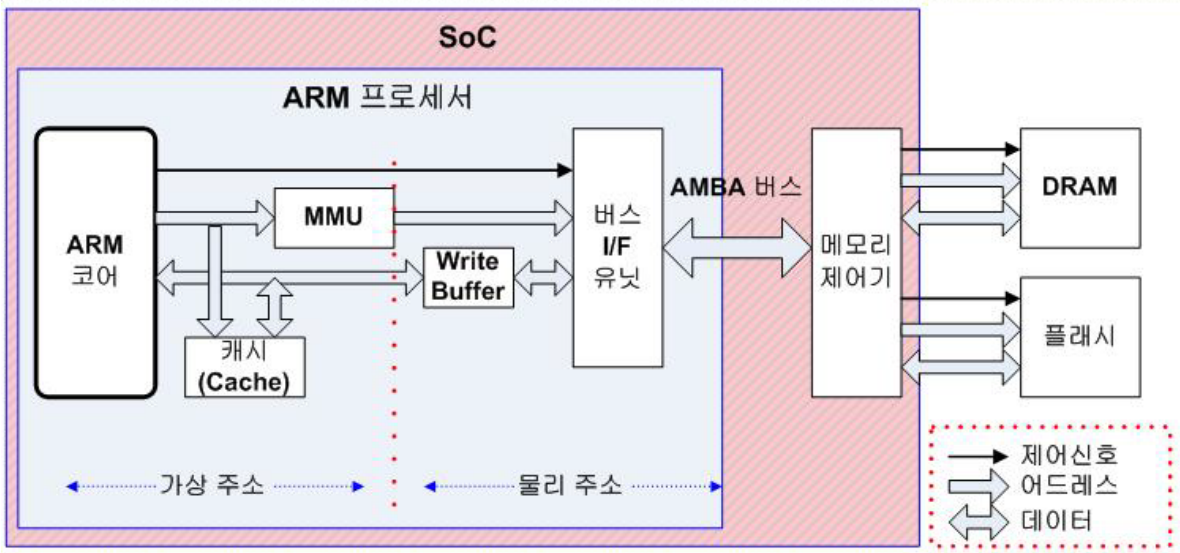
- Co-processor 기반 추가 기능
  - 기존 ARM core는 MMU가 없는 고정된 지점 계산 엔진
  - 추가적인 구조는 cache memory, MMU, write buffer, TCM(Tightly-coupled-memory가 포함된 co-processor
    - 모든 추가 구조들은 Co-processor 15에 의해 구성됨

      부동소수점 모듈로 쓰임

      캐쉬 메모리, MMU

coprocessor에게 전용reg로 말을 검

ARM Core and Memory Architecture



ARM Processor Configuration

- 구성 및 제어에 CP15 사용
  - 캐시, MMU, MPU, Endian
- Co- processor 설정
  - Co-processor 설정은 MRC 또는 MCR 지침을 통해서만 가능 (일반 레지스터에서 CP15레지스터로 데이터 전송)

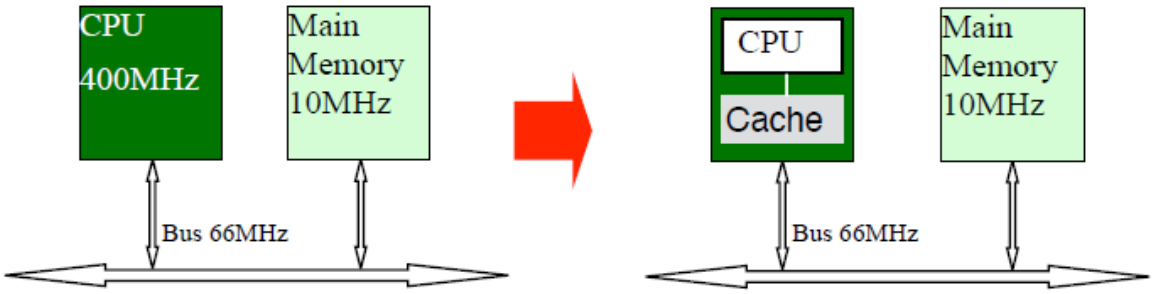
Representative CP15 Registers

이런 다양한게 있고여

Register	Usage	
0	ID code register	<opc_2>=0
	Cache type register	<opc_2>=1
1	Control Register	Cache, MMU enable, Endian Clock, etc.
2	Translation table base register	
3	Domain access control register	
5	Fault status register	
6	Fault address register	
7	Cache operation register	Cache control
8	TLB operation register	
9	Cache lockdown register	
10	TLB lockdown register	
13	FSCE PID register	Fast Context Switching Extension
14	Debug support register	DCC enabled
4, 11, 12	Reserved	

1. 캐시 lockdown 레지스터
2. TLB 작동 레지스터
3. TLB lockdown 레지스터

Cache Memory

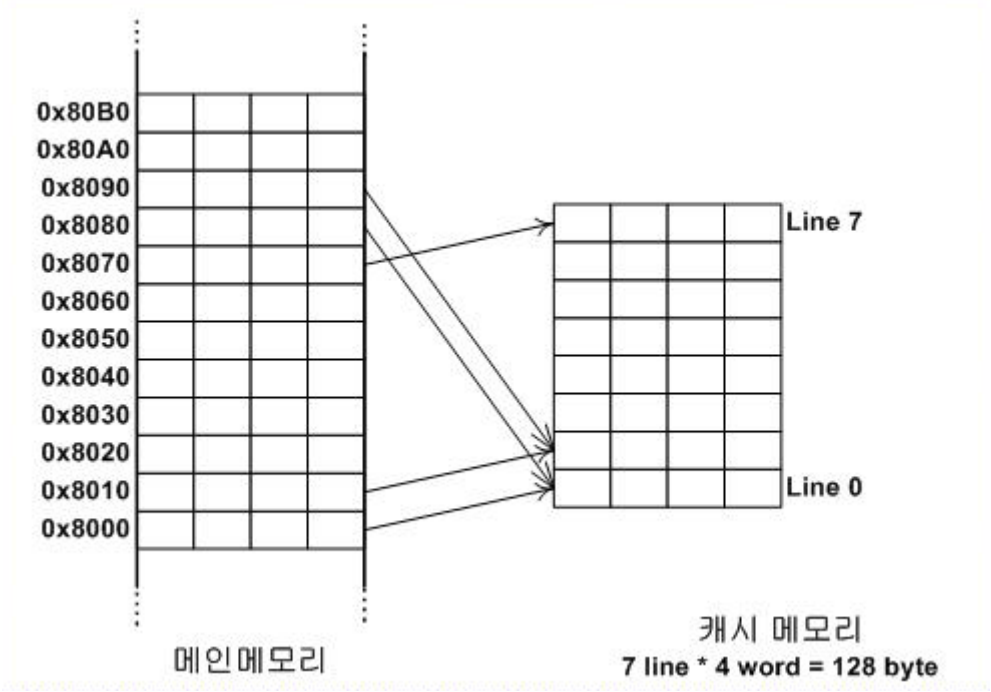


- 캐시 메모리는 CPU와 메모리 간의 속도 차이를 채움
- 캐시 구성
  - Direct mapped, set associative, fully associative
  - 캐시 line size (한번에 fetch 하는 크기, 캐시영역의 기본 단위)
  - page 의 크기는 cache의 크기와 같음
- 성능(performance) 요인(factors)
  - 캐시 hit (miss) 비율

Direct-Mapped Cache = 1 way set associative cache (복습)

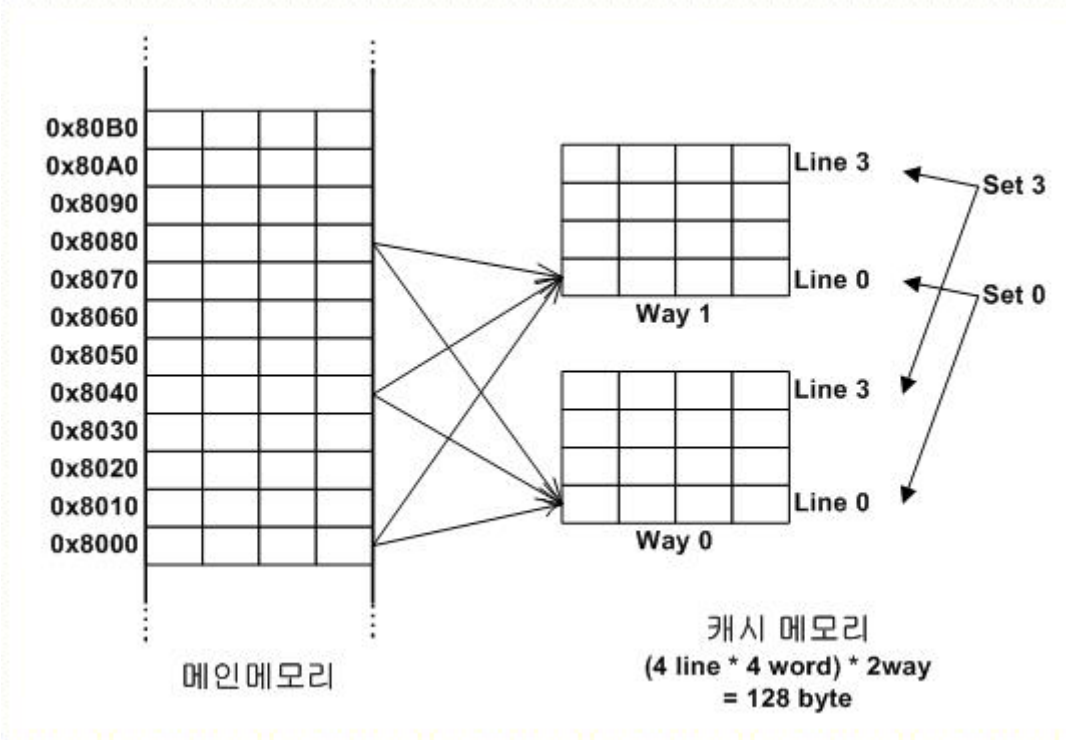
- main memory 주소당 캐시의 자리는 딱 1곳으로 unique하게 정해진다.

교통정리 확실, 효율 떨어짐(cache miss 발생할 확률 높아짐)



Set-Associative Cache(복습)

- direct처럼 한 블록이 들어갈 수 있는 자리가 고정되어 있는 것이 아닌 각 블록당 n개의 배치 가능한 위치를 갖는 n-way를 갖음



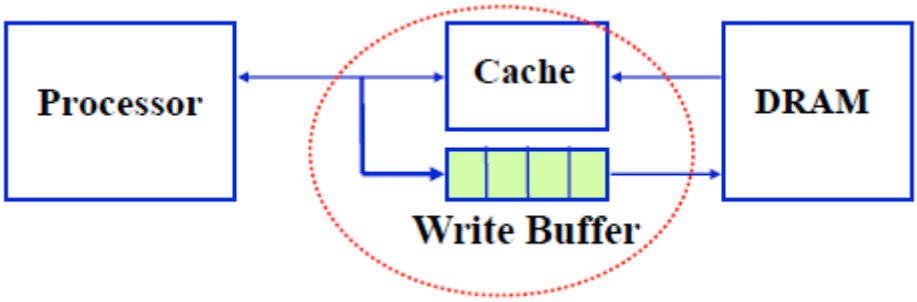
- fully-associative cache : cache의 어느 장소로의 저장이 main memory의 어느 line이나 허가된다. - cache miss 문제는 적지만 블록을 찾기 위해서는 캐시 내의 모든 엔트리를 검색

cache line size = 512bit

Write Buffer

- 프로세서 코어와 주 메모리 사이의 매우 작은 FIFO메모리, 캐시로부터 메인 메모리에 쓰기를 지원하는 매우 작은 FIFO 버퍼
- CPU에서 메모리로 쓰기 중 속도 차이를 극복하기 위해 사용

- CPU는 실제 메인 메모리에 도달하기 전에 write buffer 로 쓰기 작업 완료
- 프로세서에서 DRAM으로 request를 내려보낼 때 wrtie가 다 되었는지 기다릴 필요 없이 write buffer에 일단 보내면 background으로 씀



Special Cache Control

쫓겨나지 않게

- Cache Flush
  - 메모리에서 새로운 데이터를 fetch해오기 위해 캐시의 내용들을 정리 (cache에 수정된 내용이 무효화됨)
  - context switch 때 필요 (다른 process가 바뀌었을 때)
  - DMA로 전송할 때 필요 (이전값을 전송하지 않고, main memory에서 바로 읽어서 보내기 때문에)
- Cache lockdown(잠금)
  - 특정 캐시 line이 교체(replace) 되는 것을 방지, cache miss를 피하기위함
  - locking 해야할 항목

    vector interrupt table, ISR, 자주사용되는exteranl variables

MMU (Memory Management Unit)

4GB의 메모리 space를 한정된 몇 천개의 영역으로 나눔, 영역 하나 : 섹션(1MB) -> 4GB에 4095개의 섹션

섹션 하나는 1MB

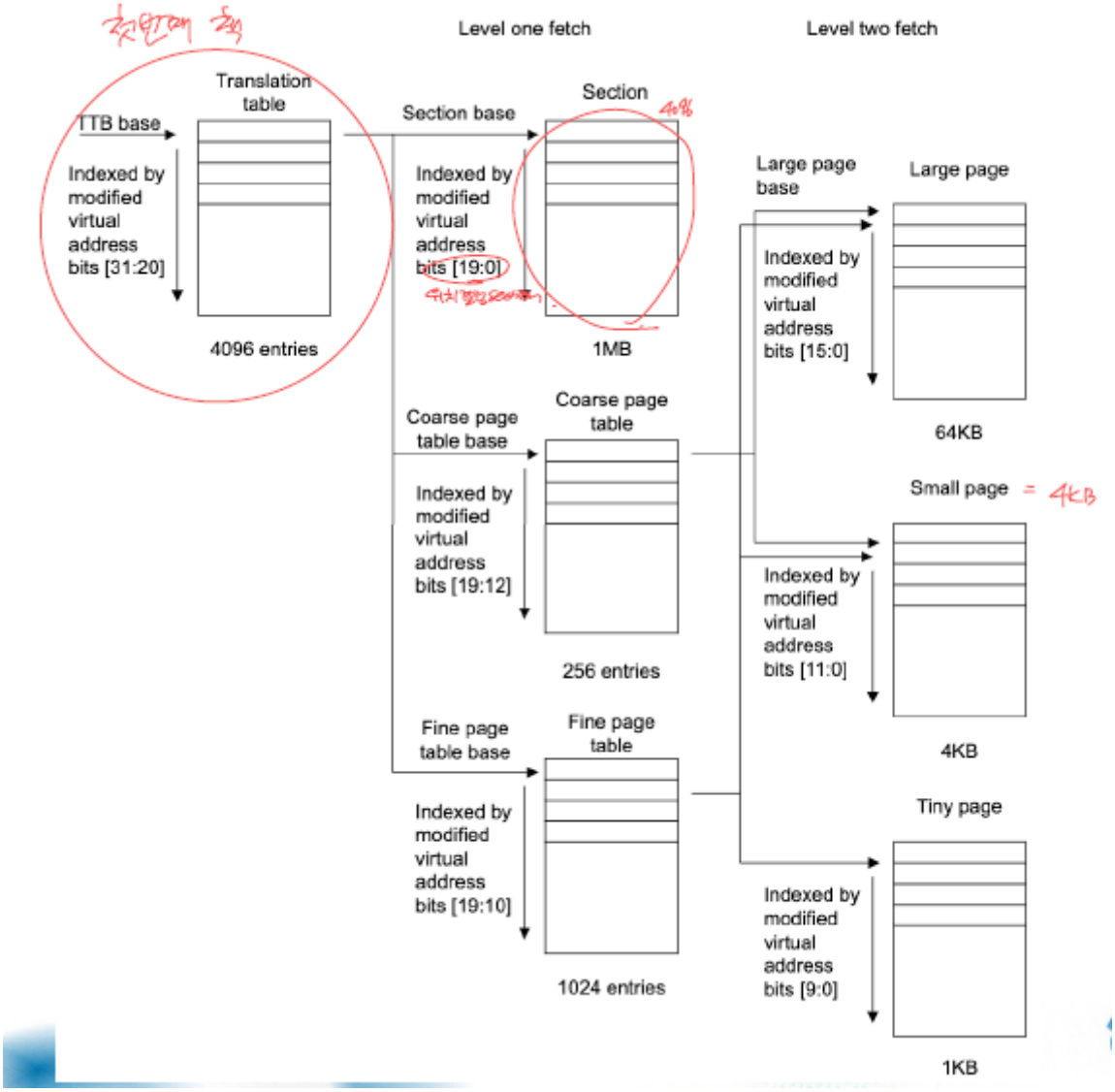
4KB로 align

Address Translation by MMU

TLB (Translation Lookaside Buffer)

Page Sizes in ARM

- Section
  - 1MB page
- Tiiny page
  - 1KB page
- Small page
  - 4KB page
    - Typical page size
- Lage page
  - 64 KB page



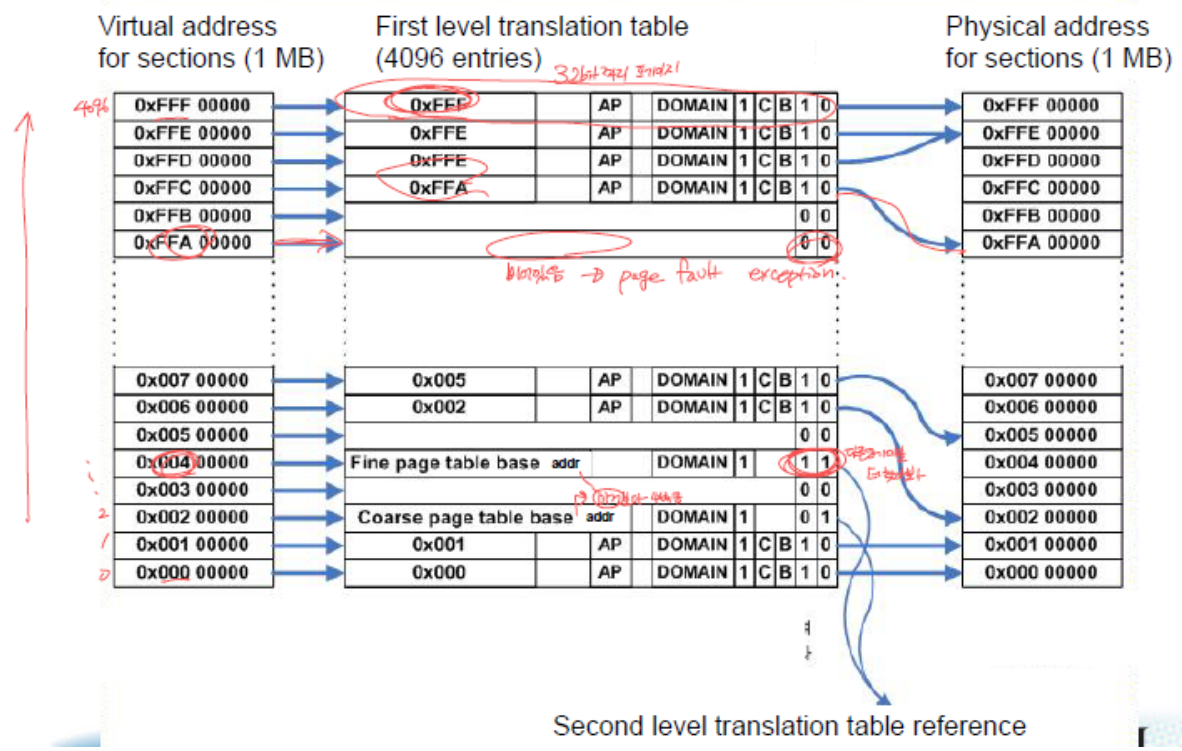
사용하는 책만 더 구체적으로 펼침

cp15 inderface : 시작 point 세팅

special register : start address값 저장

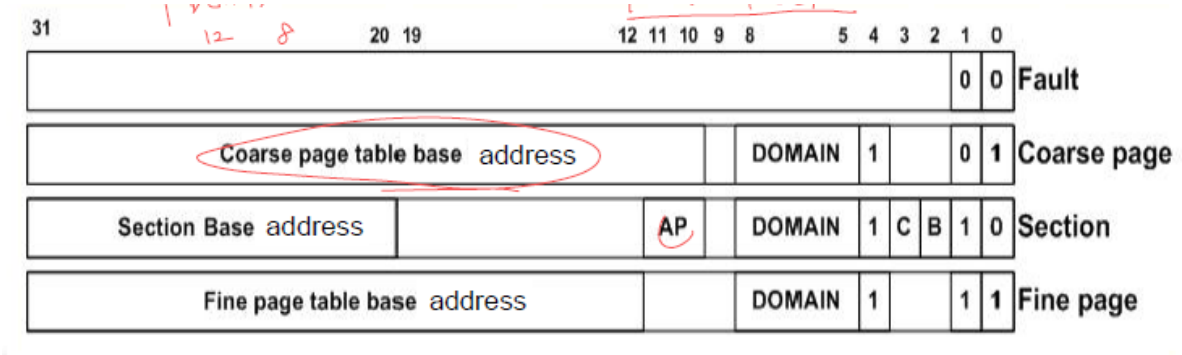
MMU라는 하드웨어가 참조하여 page fault나 page변환이 일어나야할 때 access함

Address Translation



0xFFA : 갔는데 완전 빈페이지임 : page fault 일어남

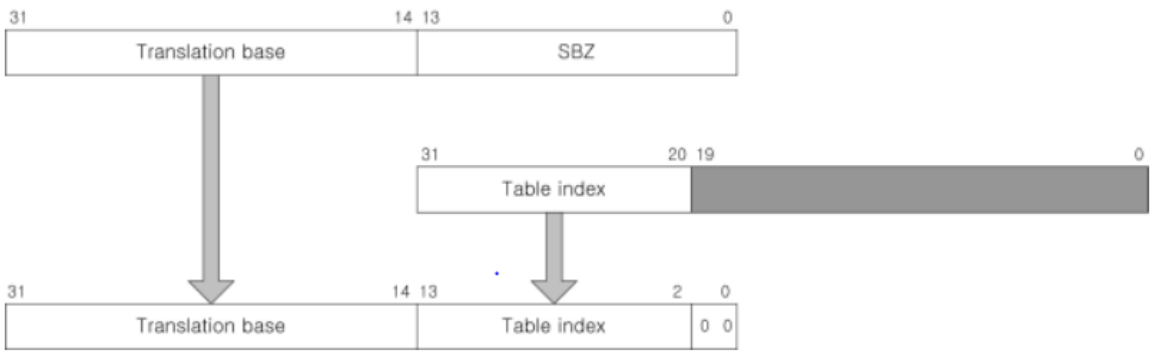
Level 1 Descriptor



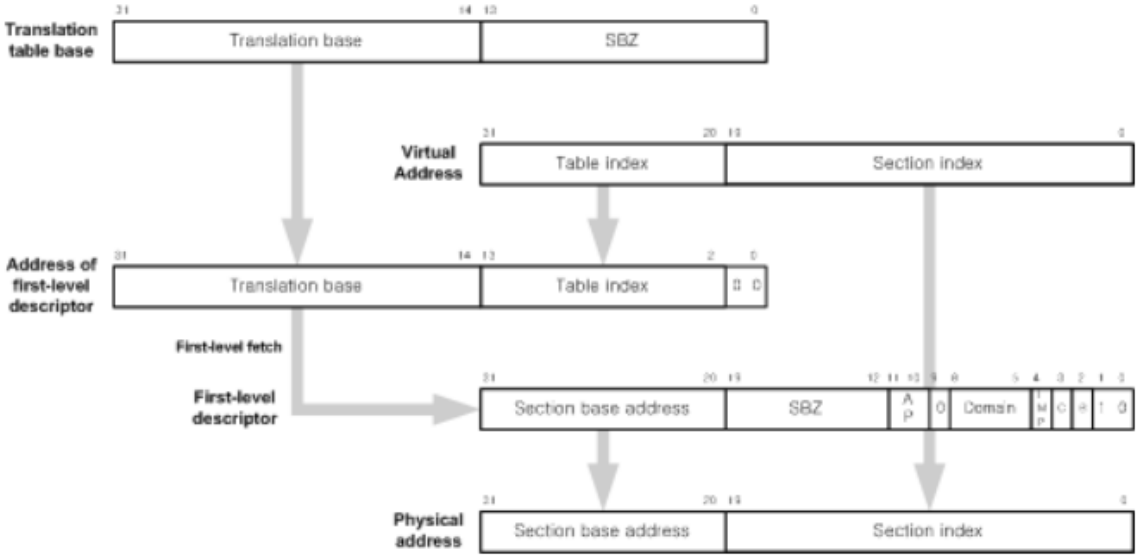
Level 2 Descriptor

Caching and Write Buffer Control

First-Level Translation



Section Translation Sequence



Small Page Translation