

리눅스 커널 소스 분석

메모리(캐시)

커널연구회(www.kernel.bz)

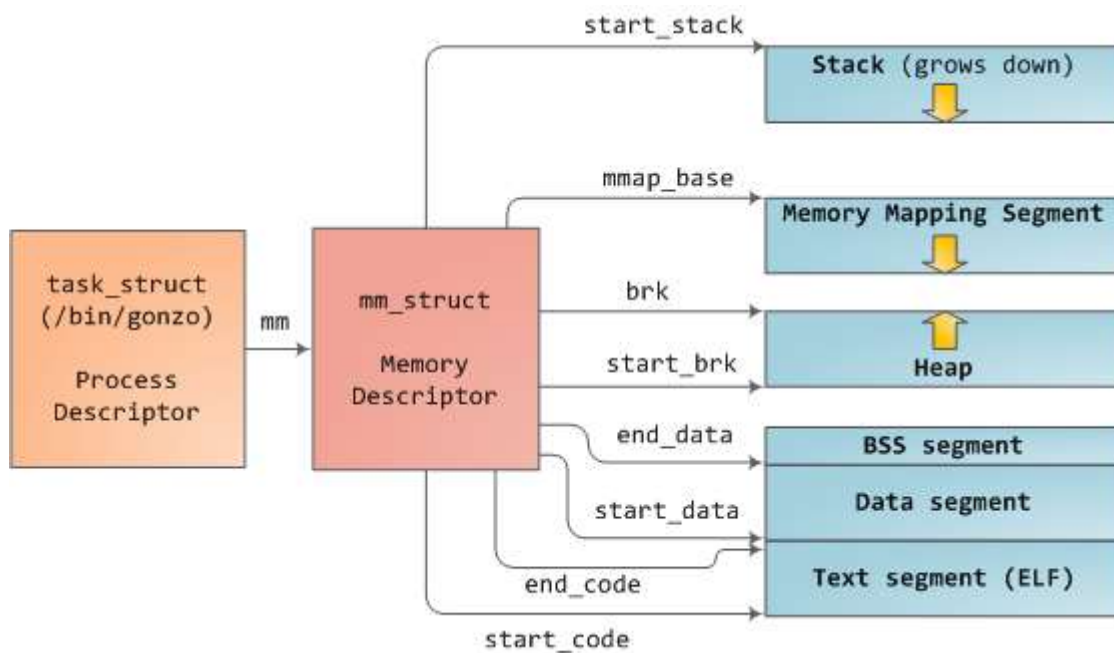
정재준(rgbi3307@nate.com)

내용

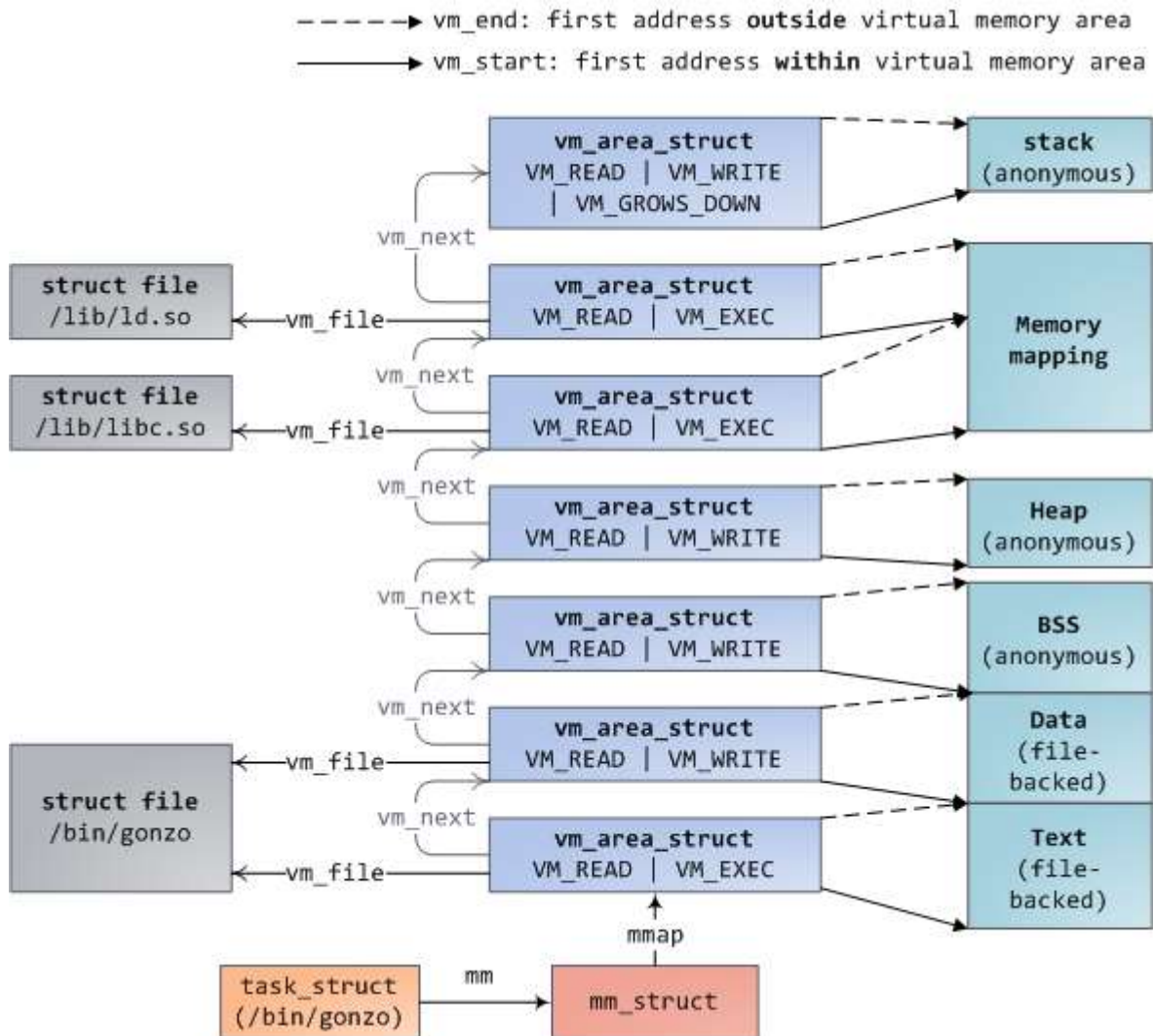
메모리(캐시).....	1
1. 메모리 주소.....	2
1.1 주소 연결	2
1.2 BUDDY MEMORY ALLOCATION.....	3
2. 캐시 메모리.....	4
2.1 캐시 어드레싱.....	4
2.2 SET-ASSOCIATIVE.....	8
2.3 CACHE COHERENCE	10
2.3.1 MSI 프로토콜.....	11
2.3.2 MESI 프로토콜.....	12
2.3.3 MOESI 프로토콜.....	14

1. 메모리 주소

1.1 주소 연결



그림출처: <https://manybutfinite.com/post/how-the-kernel-manages-your-memory/>



그림출처: <https://manybutfinite.com/post/how-the-kernel-manages-your-memory/>

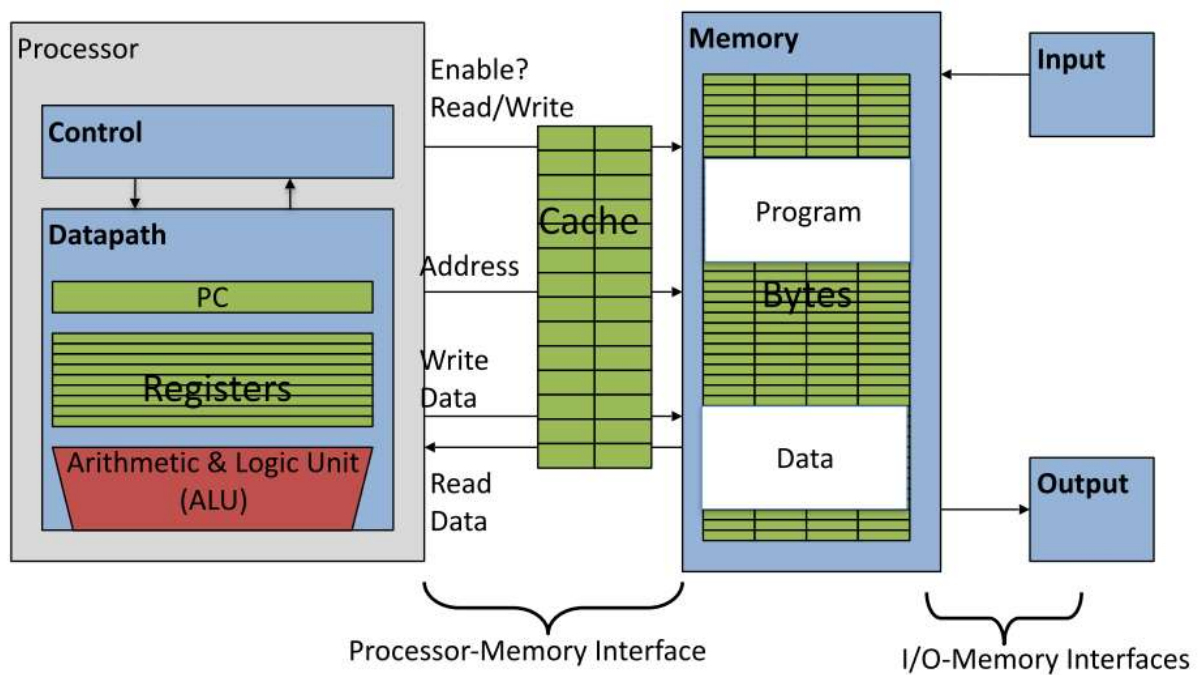
1.2 Buddy memory allocation

https://en.wikipedia.org/wiki/Buddy_memory_allocation

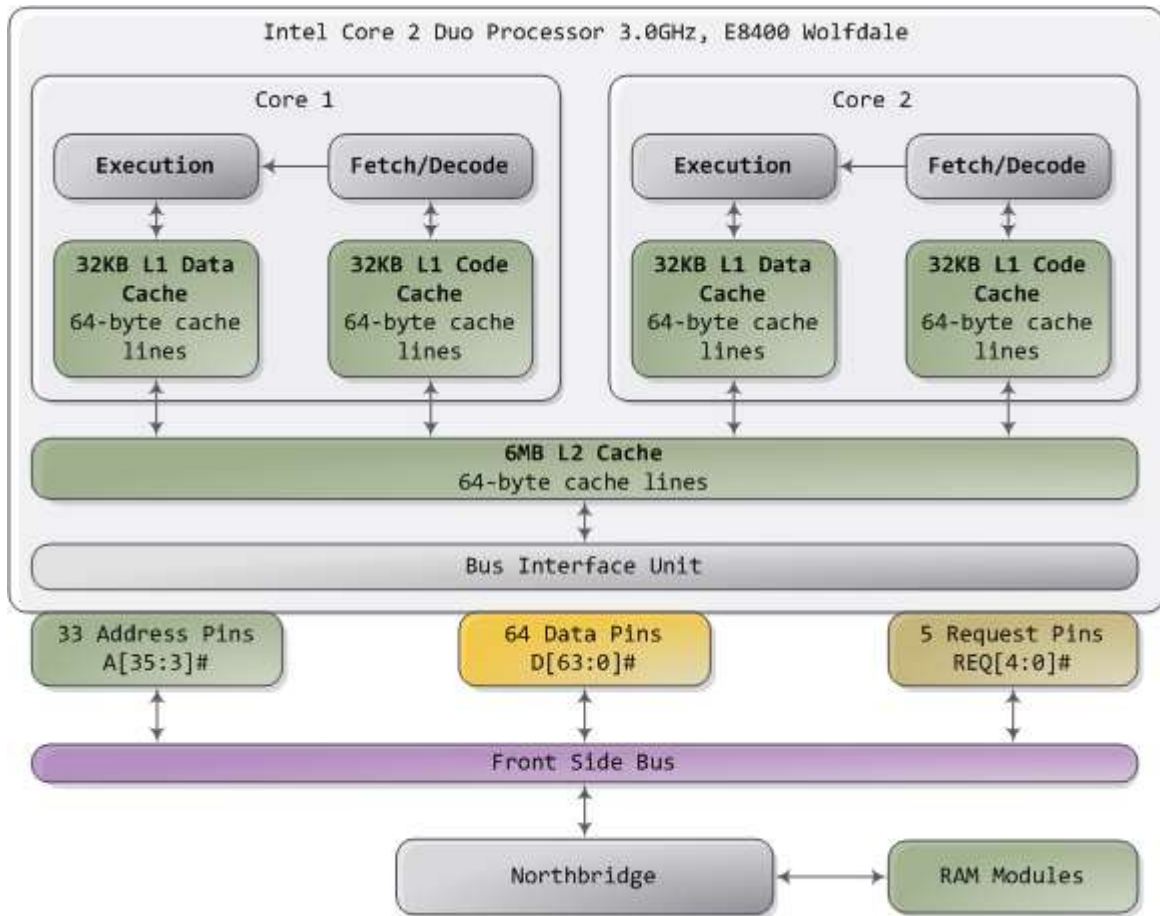
2. 캐시 메모리

2.1 캐시 어드레싱

캐시와 메모리 접근 경로



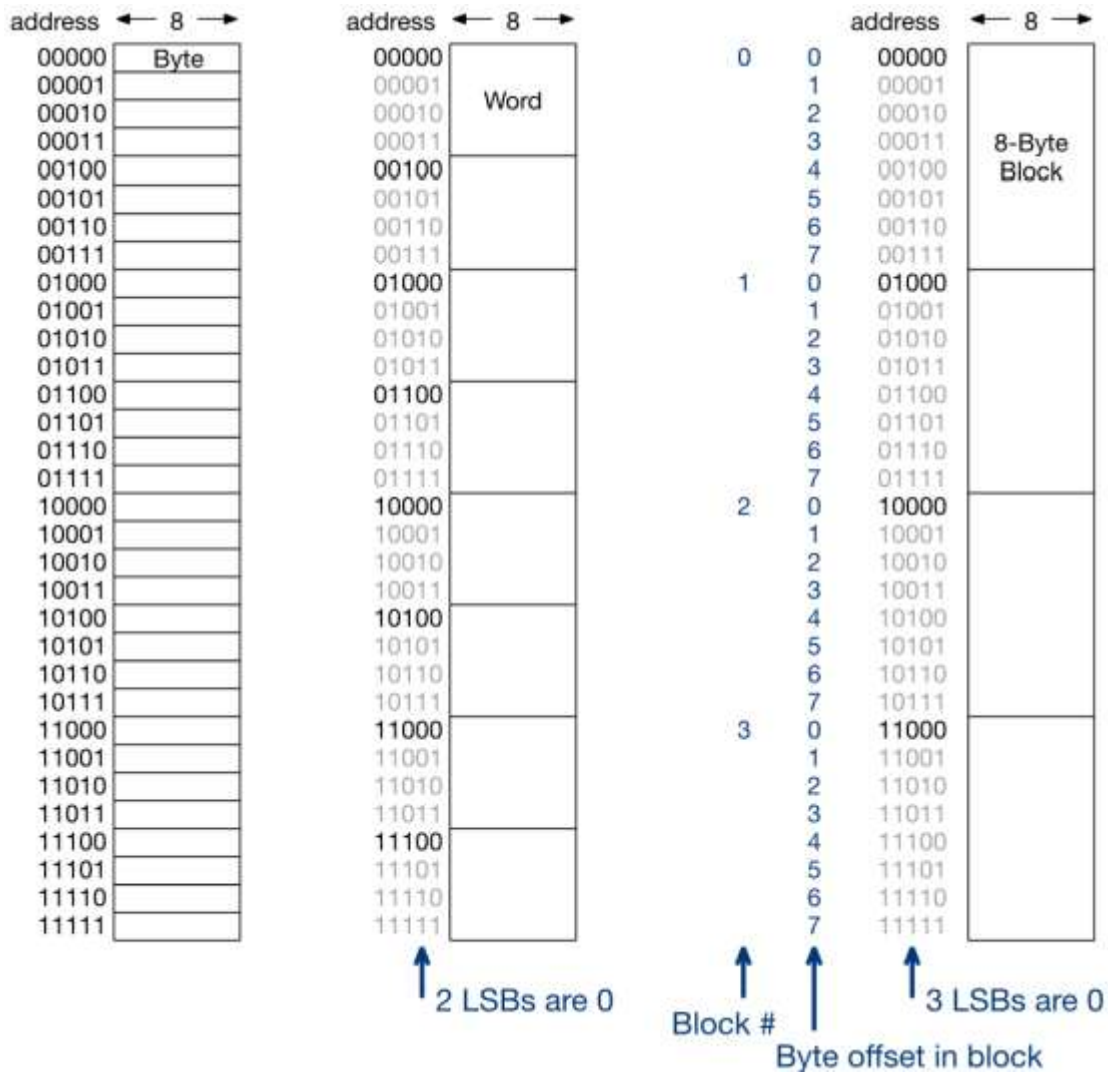
그림출처: 버클리 대학교 인터넷 강의자료



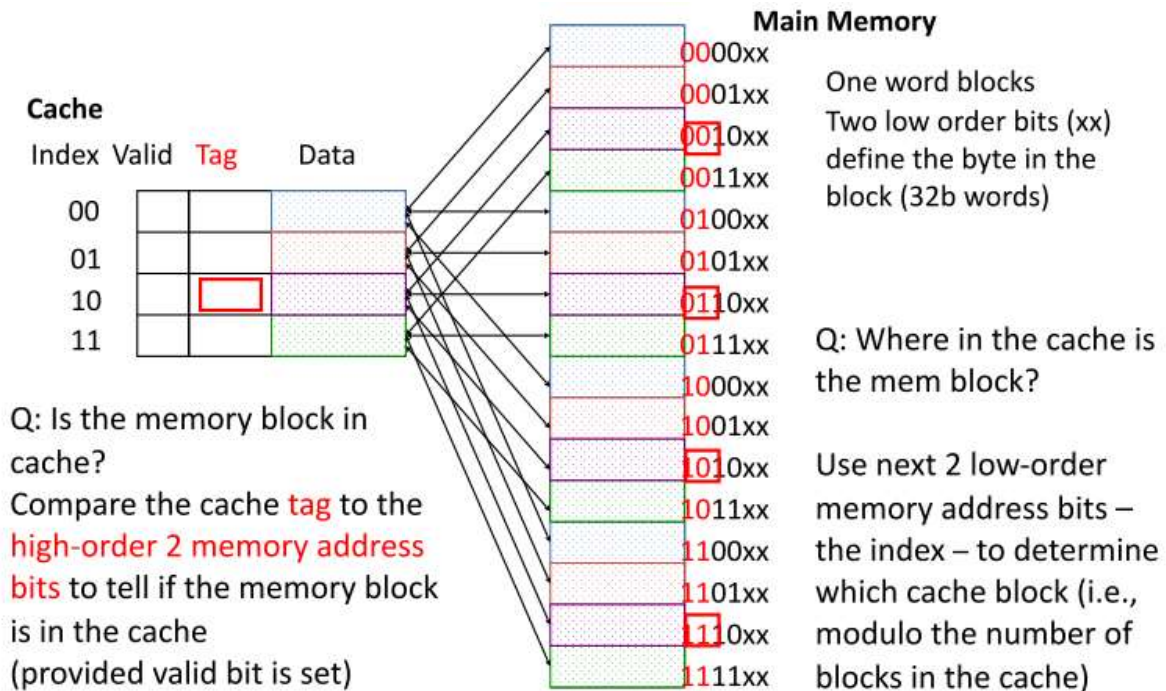
<http://duartes.org/gustavo/blog>

그림출처: <http://www.cnblogs.com/blockcipher/archive/2013/03/27/2985115.html>

메모리 블록 어드레싱

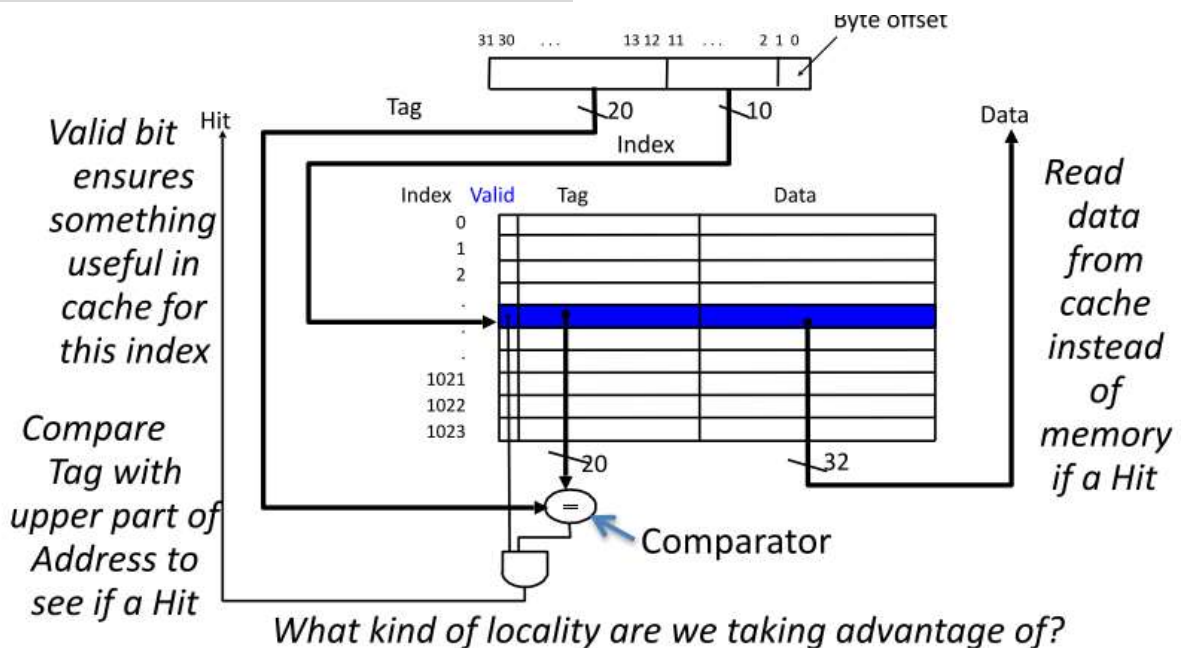


그림출처: 버클리 대학교 인터넷 강의자료



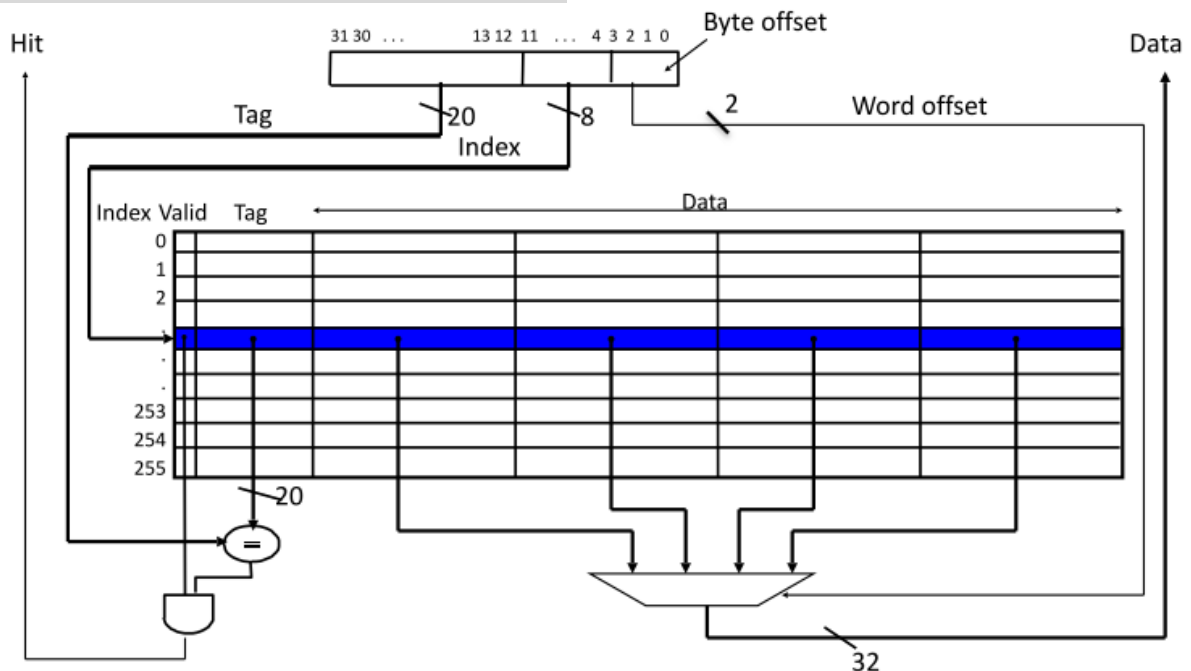
그림출처: 버클리 대학교 인터넷 강의자료

캐시 인덱스 1024 (전체 데이터 크기 4KB)



그림출처: 버클리 대학교 인터넷 강의자료

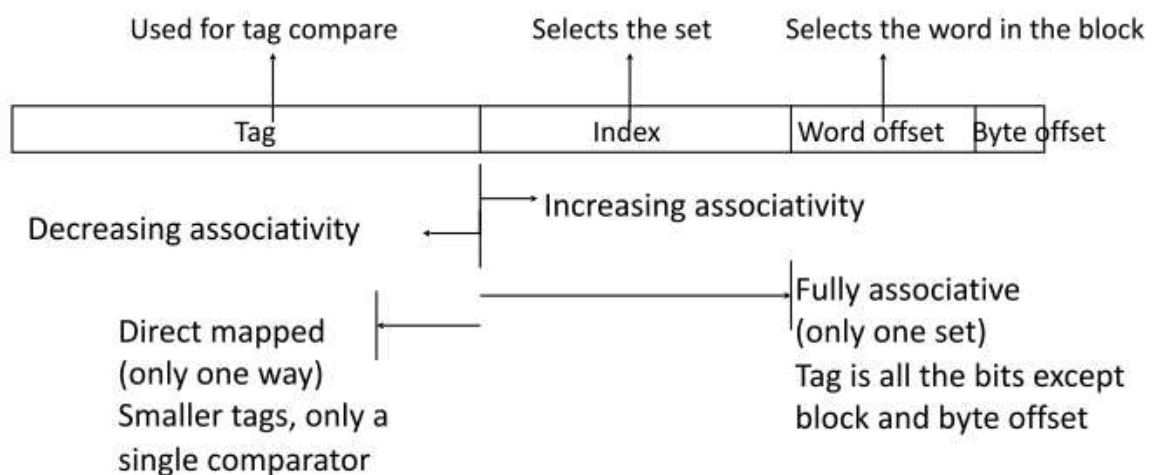
캐시 인덱스 256 (전체 데이터 크기 4KB)



What kind of locality are we taking advantage of?

그림출처: 버클리 대학교 인터넷 강의자료

2.2 Set-Associative

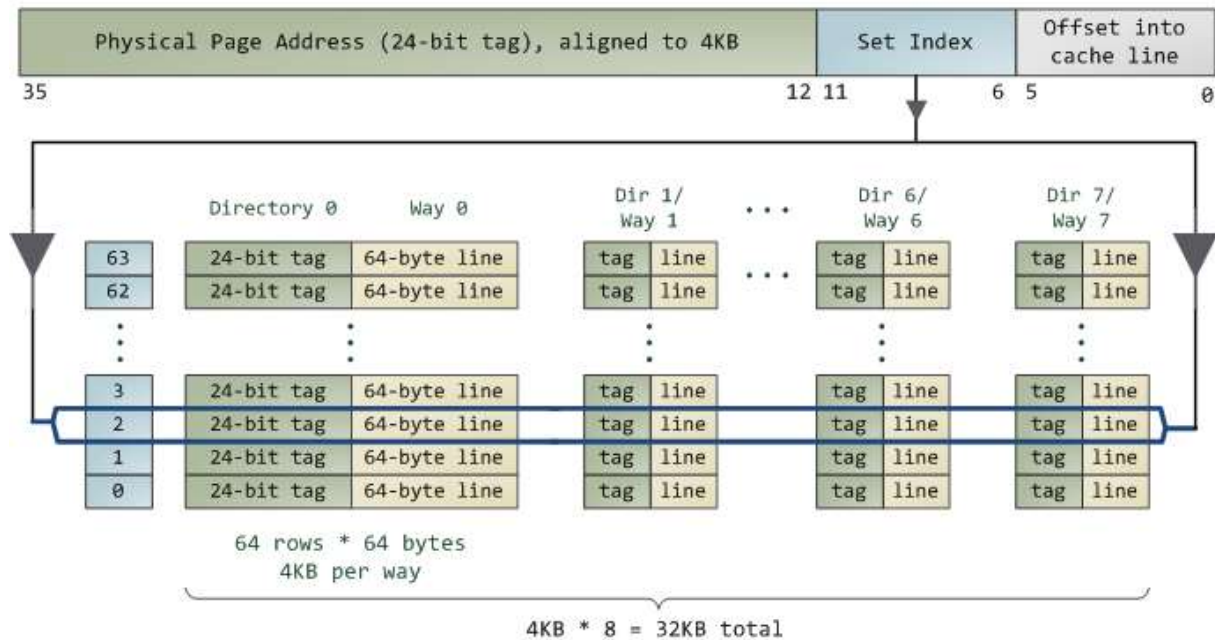


그림출처: 버클리 대학교 인터넷 강의자료

L1 Cache - 32KB, 8-way set associative, 64-byte cache lines

1. Pick cache set (row) by index

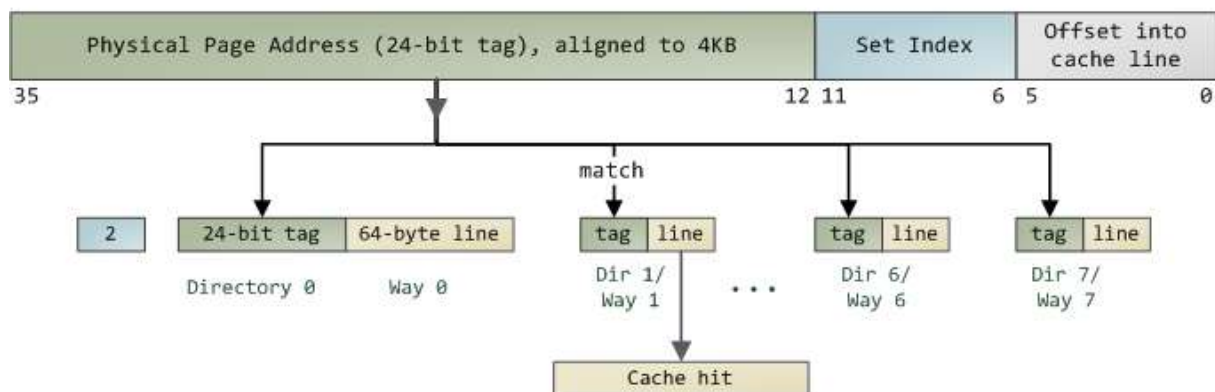
36-bit memory location as interpreted by the L1 cache:



그림출처: <http://www.cnblogs.com/blockcipher/archive/2013/03/27/2985115.html>

2. Search for matching tag in the set

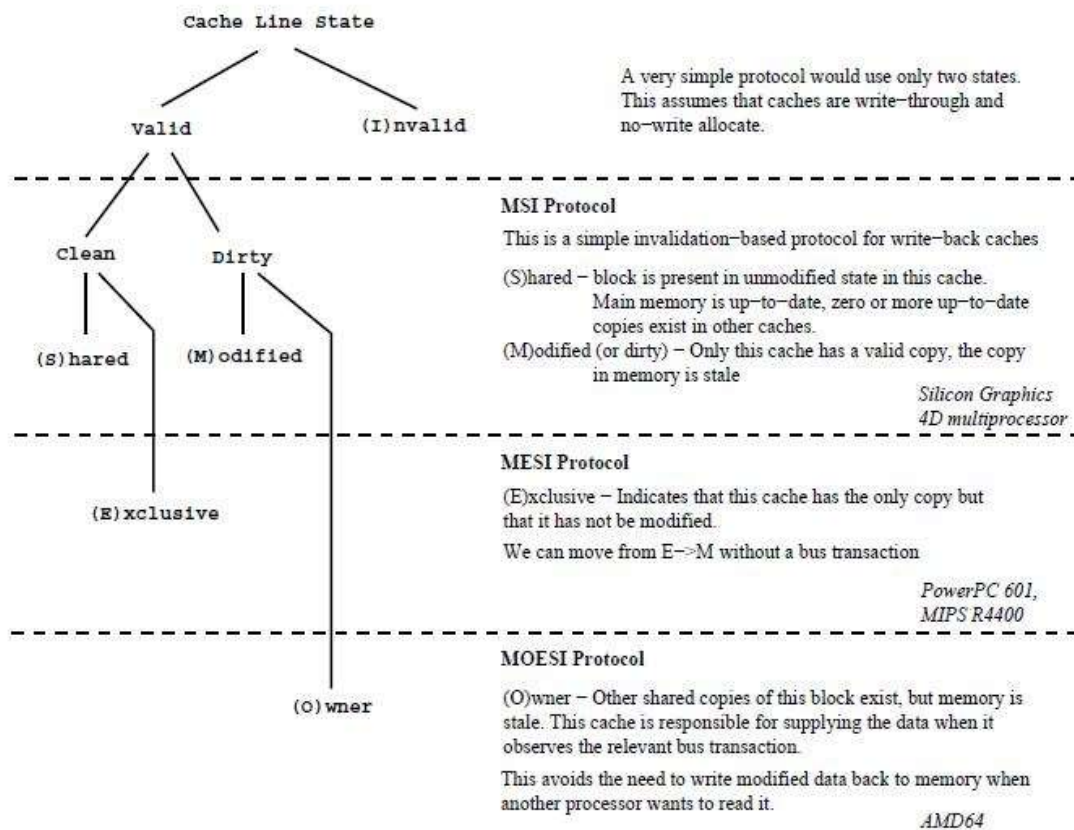
36-bit memory location as interpreted by the L1 cache:



그림출처: <http://www.cnblogs.com/blockcipher/archive/2013/03/27/2985115.html>

2.3 Cache Coherence

캐시 일관성을 위한 캐시 라인 상태



그림출처: http://wiki.expertiza.ncsu.edu/index.php/Chp8_my

Valid: 캐시에 데이터 존재.

Invalid: 캐시에 데이터 없음.

Shared(Clean): 캐시 데이터를 수정하지 않은 상태, 다른 캐시에 동일한 데이터(사본) 존재.

Modified(Dirty): 이 캐시만 유효한(valid) 사본을 가지고 있음.(데이터 수정). 메모리에는 반영하지 않은 상태임.

Exclusive: 이 캐시만 데이터 복사본을 가지고 있지만, 수정은 하지 않음.

Owner: 이 캐시에서 공유한 복사본들이 존재하지만 메모리에는 반영하지 않은 상태.

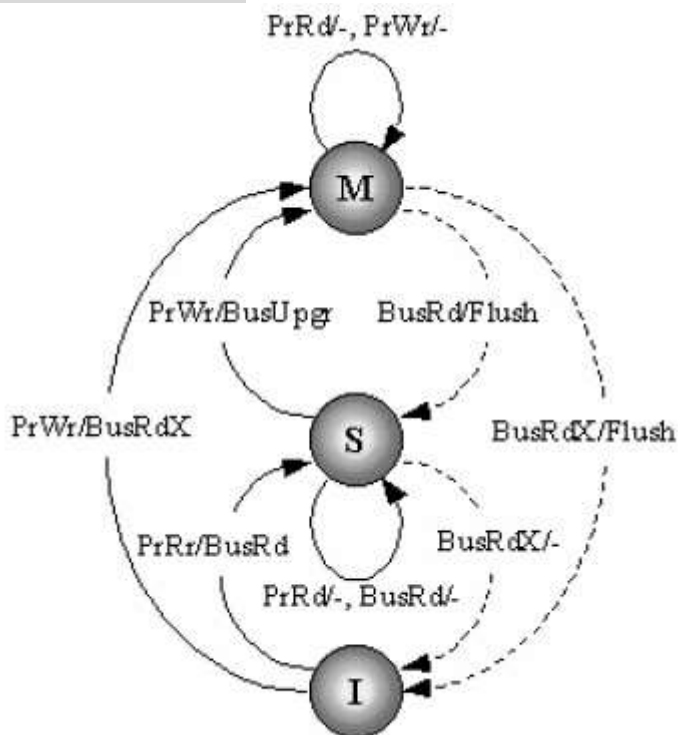
이 캐시는 데이터가 버스에 요청될때 데이터를 제공해야 한다. 그러므로, 또다른 프로세서가 이 캐시의 데이터를 읽고자 할때 수정된 데이터를 메모리에 쓰지 않아도 된다.

2.3.1 MSI 프로토콜

Shared(Clean): 캐시 데이터를 수정하지 않은 상태, 다른 캐시에 동일한 데이터(사본) 존재.

Modified(Dirty): 이 캐시만 유효한(valid) 사본을 가지고 있음.(데이터 수정). 메모리에는 반영하지 않은 상태임.

MSI 캐시 상태 변화



그림출처: http://wiki.expertiza.ncsu.edu/index.php/Chp8_my

MSI 프로토콜은 가장 간단하지만 실제로 사용되지는 않는다. 하지만 캐시 상태 변화를 이해하는 기본이고 다음 프로토콜로 발전된다. MSI 프로토콜에는 2개의 프로세서 요청과 4개의 버스 요청이 있다.

PrRd : 캐시 블록으로 읽는 프로세서 측 요청.

PrWr : 캐시 블록에 기록하기위한 프로세서 측 요청.

BusRd : 다른 프로세서가 만든 캐시 블록을 읽고자 하는 버스 측 요청.

BusRdX : 다른 프로세서가 해당 블록의 유효한 복사본을 가지고 있지 않은 경우 다른 프로세서가 만든 캐시 블록에 쓰기 (읽기 제외) 요청이 있음을 나타내는 버스 측 요청.

BusUpgr : 다른 프로세서가 이미 블록의 유효한 복사본을 가지고 있으면 다른 프로세서가 만든 캐시 블록에 쓰기 (읽기 전용) 요청이 있음을 나타내는 버스 측 요청.

Flush : 전체 캐시 블록이 다른 프로세서에 의해 주 메모리에 다시 쓰여지는 것을 나타내는 버스 측 요청.

Modified (M) : 캐시 블록은 하나의 캐시에서만 유효하며 독점적으로 소유. Modified 상태는 캐시가 주 메모리의 값과 다르다는 것을 의미하며 한 위치에서만 캐시됨.

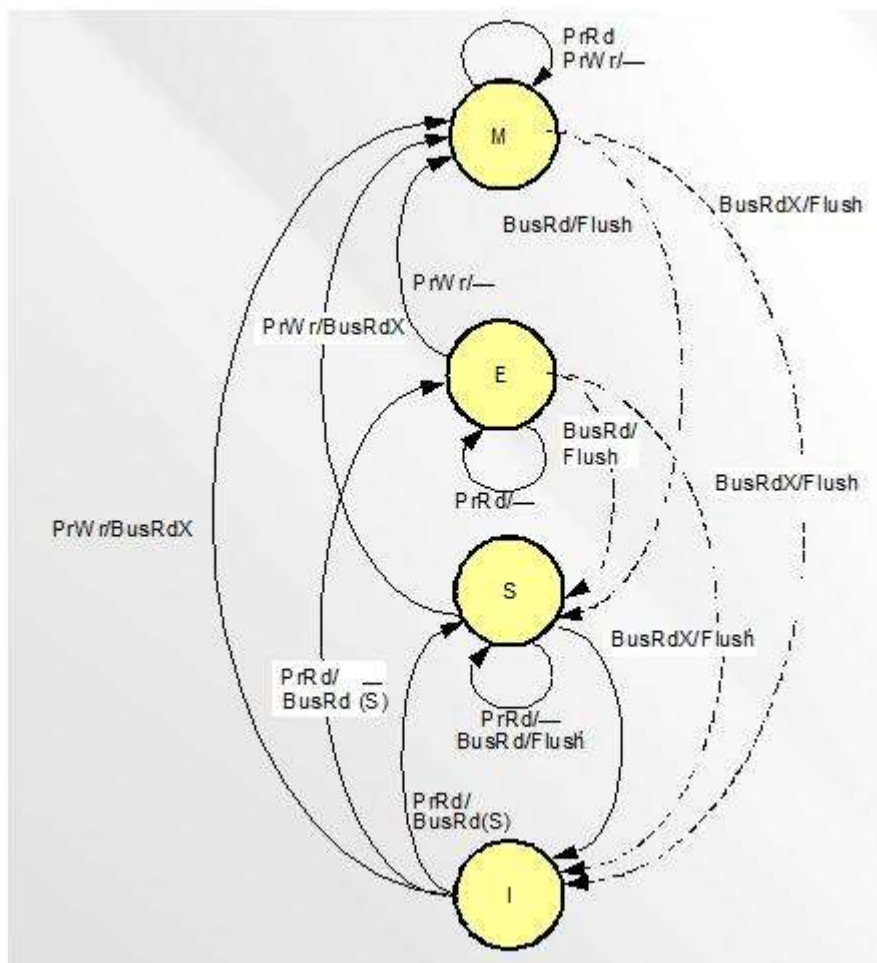
Shared (S) : 캐시 블록이 유효하며 여러 프로세서에서 공유 할 수 있음. Shared는 또한 값이 주 메모리의 값과 같음을 의미.

2.3.2 MESI 프로토콜

Shared(Clean): 캐시 데이터를 수정하지 않은 상태, 다른 캐시에 동일한 데이터(사본) 존재.

Modified(Dirty): 이 캐시만 유효한(valid) 사본을 가지고 있음.(데이터 수정). 메모리에는 반영하지 않은 상태임.

Exclusive: 이 캐시만 데이터 복사본을 가지고 있지만, 수정은 하지 않음.



그림출처: http://wiki.expertiza.ncsu.edu/index.php/Chp8_my

MESI 프로토콜에는 4개의 캐시 블록 상태가 있다.

1. **Modified (M)** : 캐시 블록은 하나의 캐시에서만 유효하며 값은 주 메모리와 다름(수정됨).
2. **Exclusive (E)** : 캐시 블록이 유효하고 깨끗하지만 하나의 캐시에만 상주(독점, 배타적).
3. **Shared (S)** : 캐시 블록이 유효하고 깨끗하지만 여러 캐시에 존재할 수 있음(공유).
4. **Invalid (I)** : 캐시 블록이 유효하지 않습니다.

PrRd : 프로세서가 캐시 블록 읽기를 요청.

PrWr : 프로세서가 캐시 블록 쓰기를 요청.

BusRd : 다른 프로세서가 만든 캐시 블록에 대한 읽기 버스 요청.

BusRdX : 아직 차단(block)되지 않은 다른 프로세서가 만든 캐시 블록에 read exclusive (쓰기) 버스 요청 (write cache-to-memory)

BusUpgr : 다른 프로세서가 이미 이 캐시를 가지고 있는 블록에 대해서 쓰기 버스 요청.

Flush: 전체 캐시 블록이 다른 프로세서에 의해 주 메모리에 다시 쓰여짐(write to memory).

FlushOpt : 전체 블록 캐시 블록을 다른 프로세서에 공급하기 위해 버스에 게시했음을 나타냄. (cache-to-cache).

2.3.3 MOESI 프로토콜

Valid: 캐시에 데이터 존재(유효)

Invalid: 캐시에 데이터 없음(무효)

Shared(Clean): 공유함, 수정하지 않음.

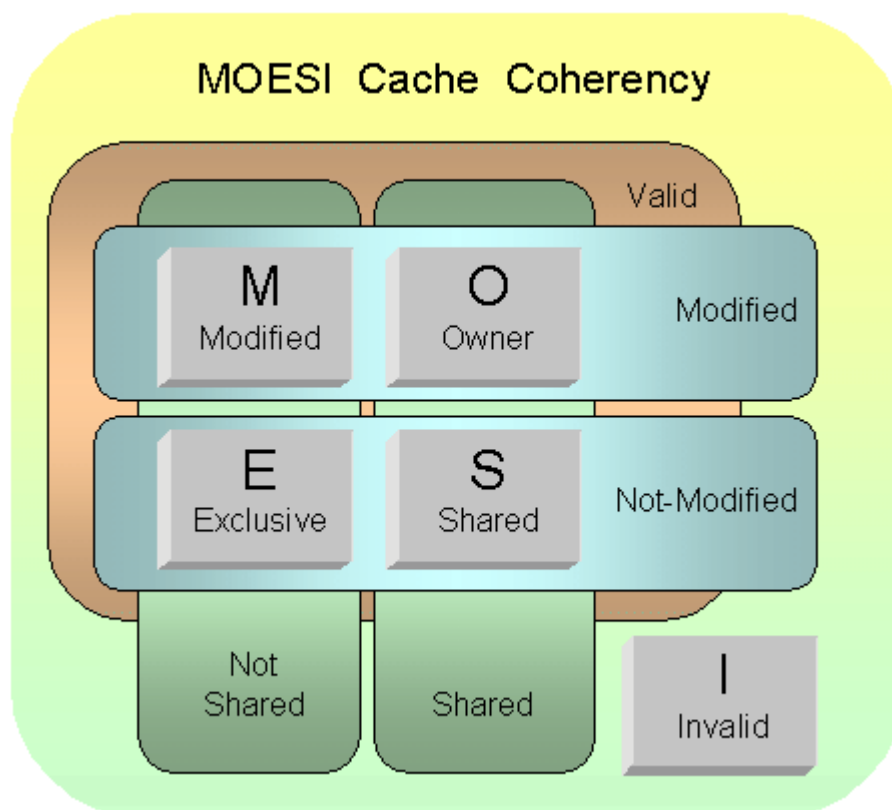
Modified(Dirty): 공유하지 않음, 수정함.

Exclusive: 공유하지 않음. 수정하지 않음.(독점)

Owner: 공유함. 수정함.

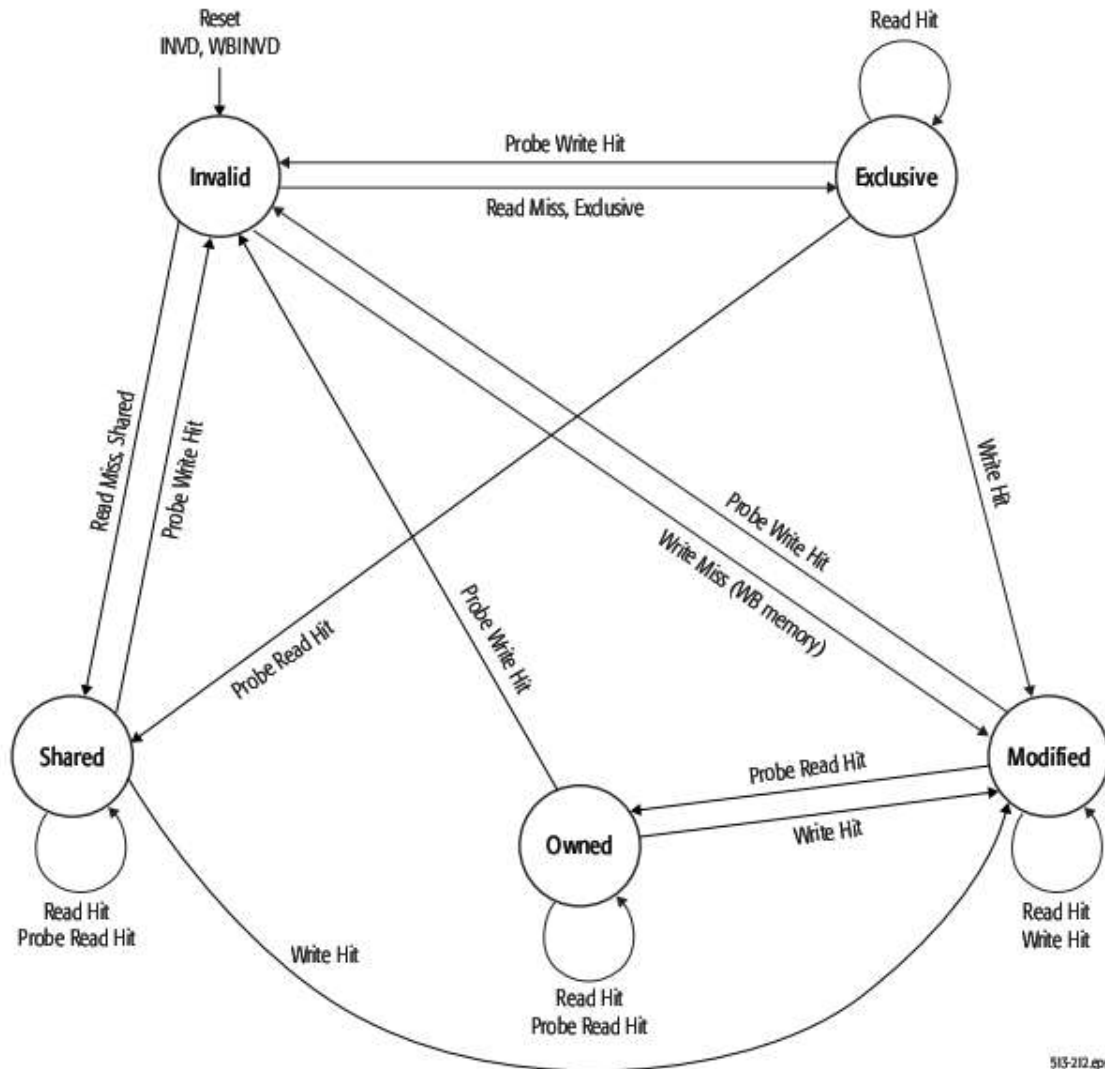
이 캐시에서 공유한 복사본들이 존재하지만 메모리에는 반영하지 않은 상태. 이 캐시는 데이터가 버스에 요청될때 데이터를 제공해야 한다. 그러므로, 또다른 프로세서가 이 캐시의 데이터를 읽고자 할때 수정된 데이터를 메모리에 쓰지 않아도 된다.

MOESI를 사용하면 수정 된 데이터를 공유하기 전에 주 메모리에 다시 기록 할 필요가 없기 때문에 대역폭을 절약하고 캐시에 훨씬 빠르게 액세스 할 수 있음.



그림출처: http://wiki.expertiza.ncsu.edu/index.php/Chp8_my

MOESI 캐시 상태 전이도



그림출처: http://wiki.expertiza.ncsu.edu/index.php/Chp8_my

MOESI 프로토콜의 상태는 다음과 같다.

Invalid (무효):

캐시 라인에 유효한 데이터 복사본이 없음. 유효한 데이터 복사본은 주 메모리 또는 다른 프로세서 캐시에 있을 수 있음.

Exclusive (배타적): 공유하지 않음. 수정하지 않음.(독점)

배타적 상태의 캐시 라인은 가장 최근의 올바른 데이터 복사본을 보유함. 주 메모리에있는 사본은 또한 가장 최근의 올바른 데이터 사본임. 다른 프로세서는 데이터 사본을 보유하지 않음.

Shared (공유): 공유함, 수정하지 않음.

캐시 라인은 가장 최근의 올바른 데이터 사본을 보유함. 시스템의 다른 프로세서는 공유 상태의 데이터 사본도 보유 할 수 있음. 다른 프로세서가 소유 상태에서 그것을 보유하지 않으면, 메인 메모리의 복사본 또한 가장 최근의 것임.

Modified (수정) : 공유하지 않음, 수정함.

수정된 캐시 라인은 가장 최근의 정확한 데이터 사본을 보유함. 주 메모리의 사본이 부실함(올바르지 않음). 다른 프로세서에는 사본이 없음.

Owner (소유): 공유함. 수정함.

소유된 캐시 라인은 가장 최근의 올바른 데이터 사본을 보유함. 소유 상태는 다른 프로세서가 가장 최근의 정확한 데이터의 사본을 보유 할 수 있다는 점에서 공유 상태와 유사함. 그러나 공유 상태와 달리 주 메모리의 복사본은 부적절 할 수 있음(수정됨). 하나의 프로세서 만 소유 상태의 데이터를 보유 할 수 있으며 다른 모든 프로세서는 공유 상태의 데이터를 보유해야 함.