CO: Computer Organization

Cache Memory

Indian Institute of Information Technology, Sri City

Jan - May - 2018

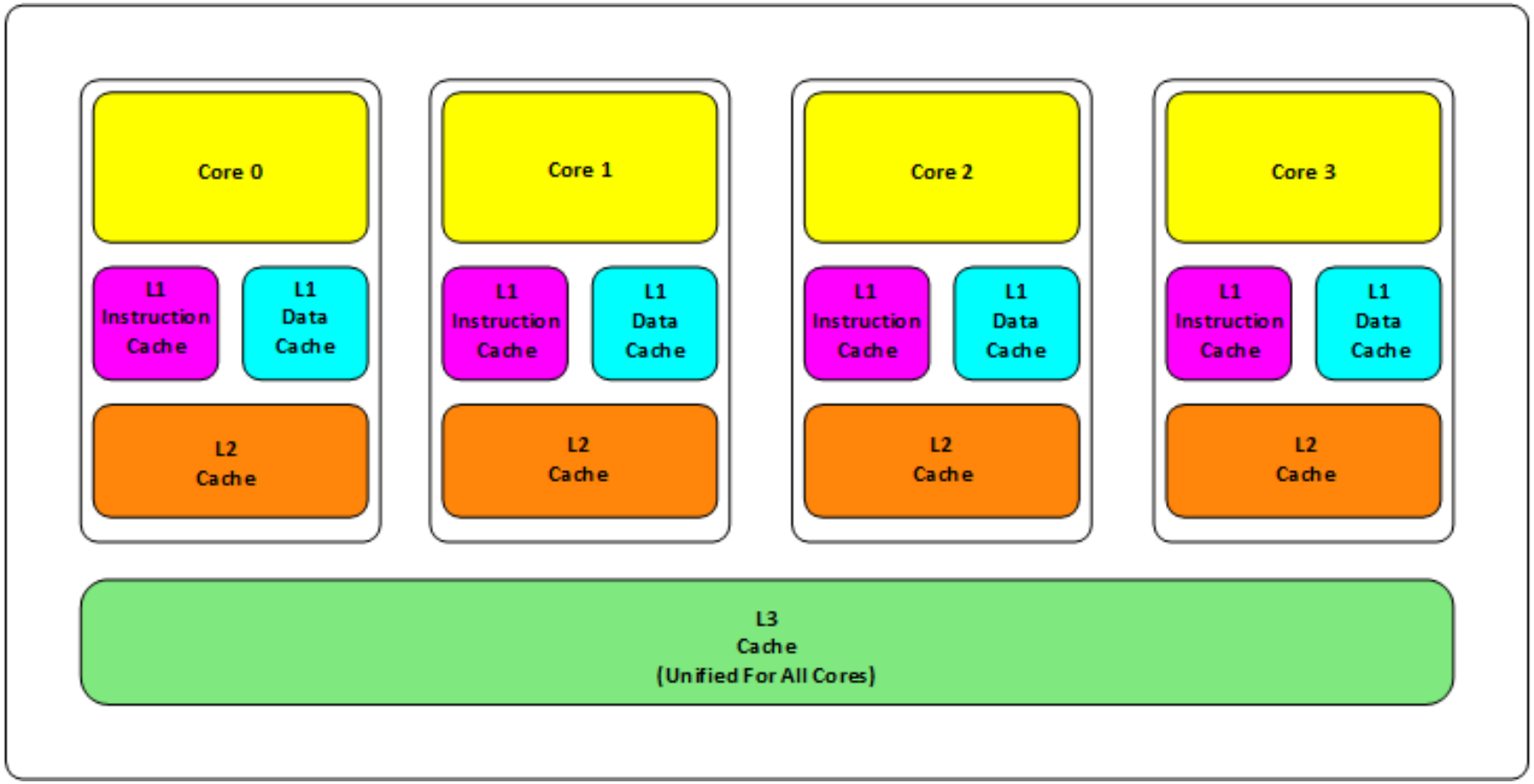
http://co-iiits.blogspot.in/

```
void MultiplyMatrices(int nCount, double **matrixA,
                      double **matrixB, double **matrixC)
{
    int i, j, k ;

    for (i = 0; i < nCount; i++)
    {
        for (j = 0; j < nCount; j++)
        {
            matrixC[i][j]=0;

            for (k = 0; k < nCount; k++)
            {
                matrixC[i][j] +=
                    matrixA[i][k]*matrixB[k][j];
            }
        }
    }
}
```

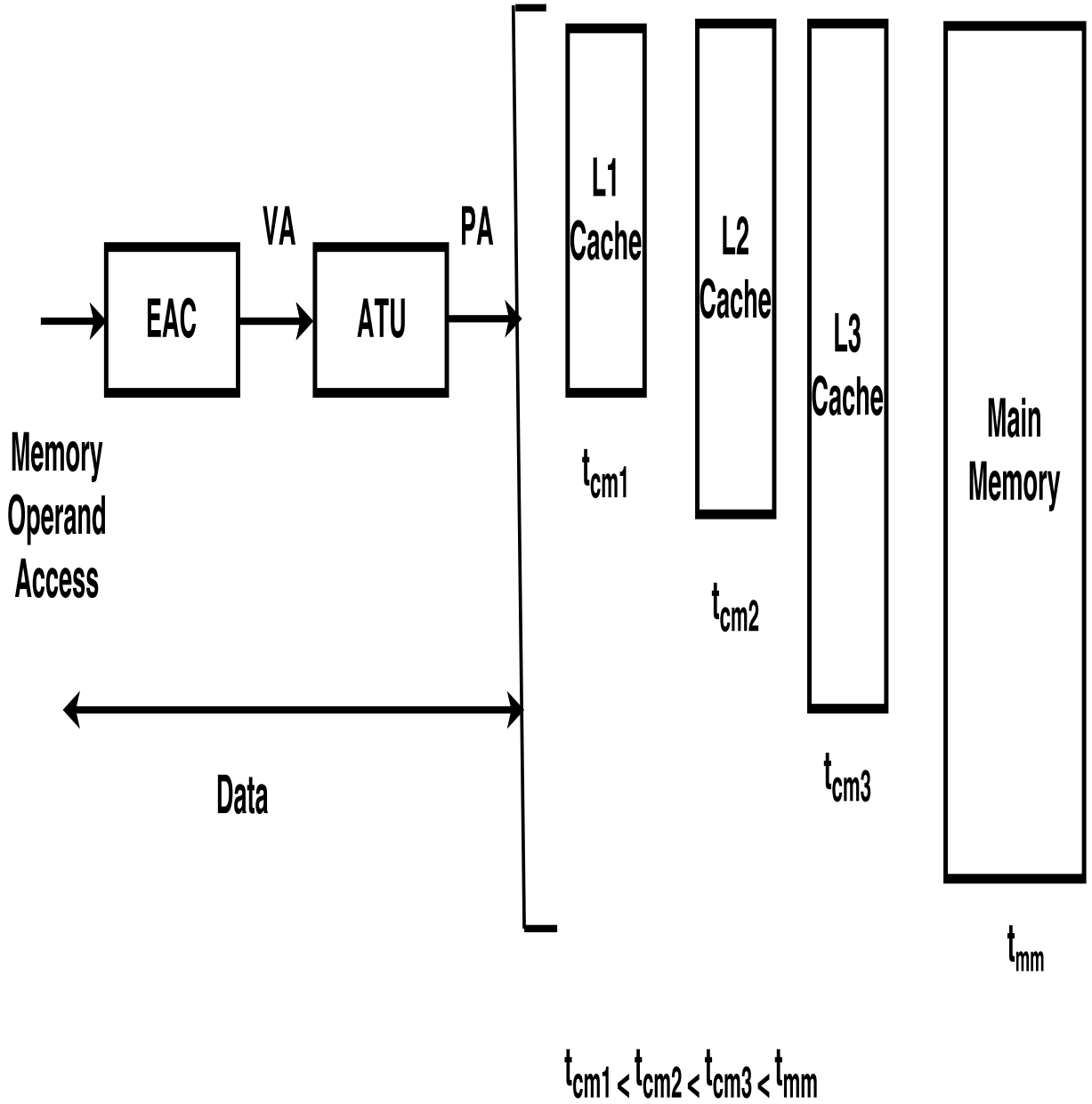
- **Locality of reference:** If a processor accesses some data now, the same data or neighbouring data will be accessed in near feature.
- **Temporal Locality:** Accesses to the same memory location that occur close together in time.
- **Spacial Locality:** Accesses to the memory locations that are close together in space.
- 90% of Execution time is spent on 10% of the code.

Cache Memory

Cache memory is a small-sized volatile memory that provides high-speed access to a processor and stores frequently used instructions and data.

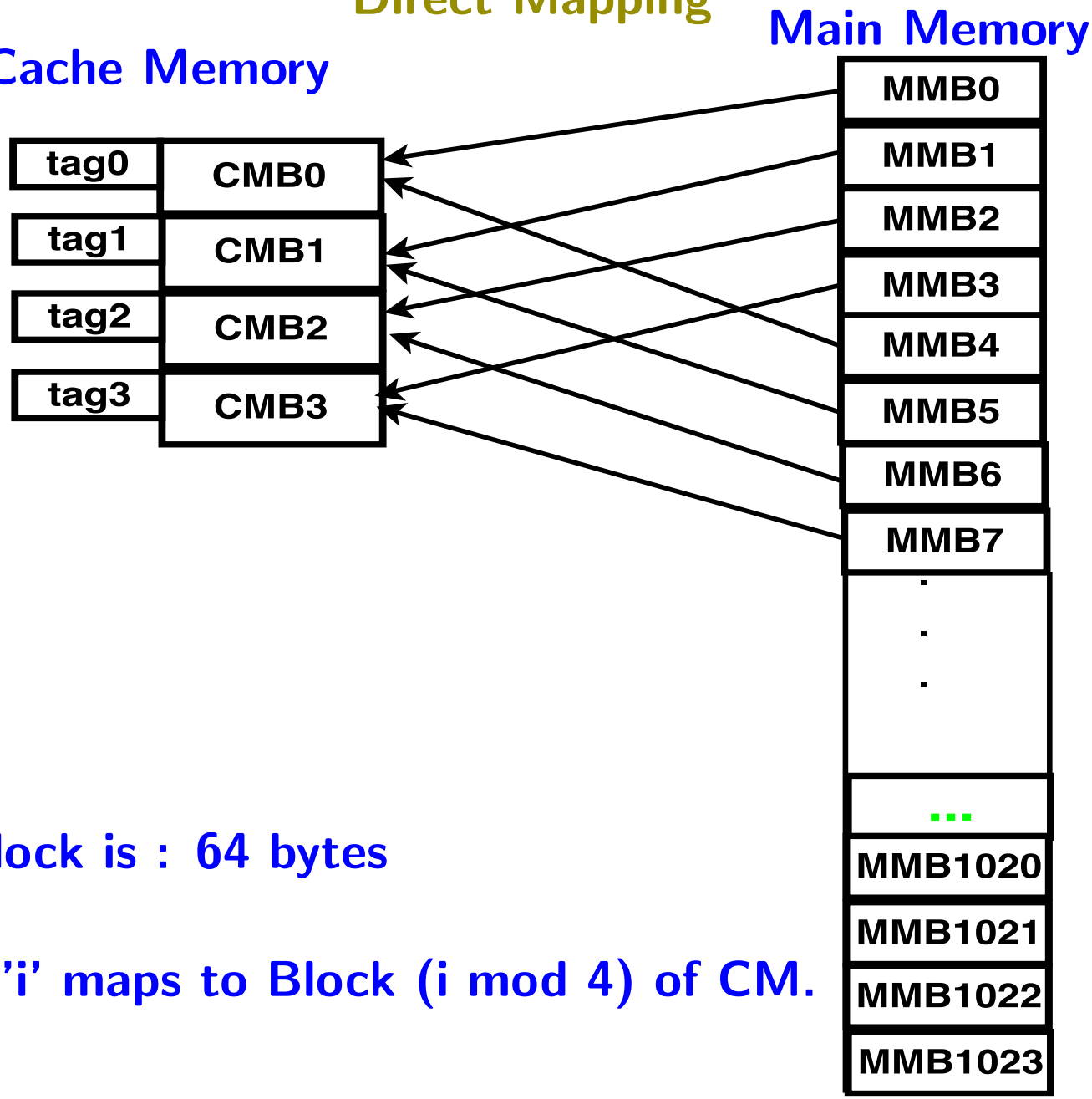


Memory Access



$$t_{cm1} < t_{cm2} < t_{cm3} < t_{mm}$$

Direct Mapping



Size of a Block is : 64 bytes

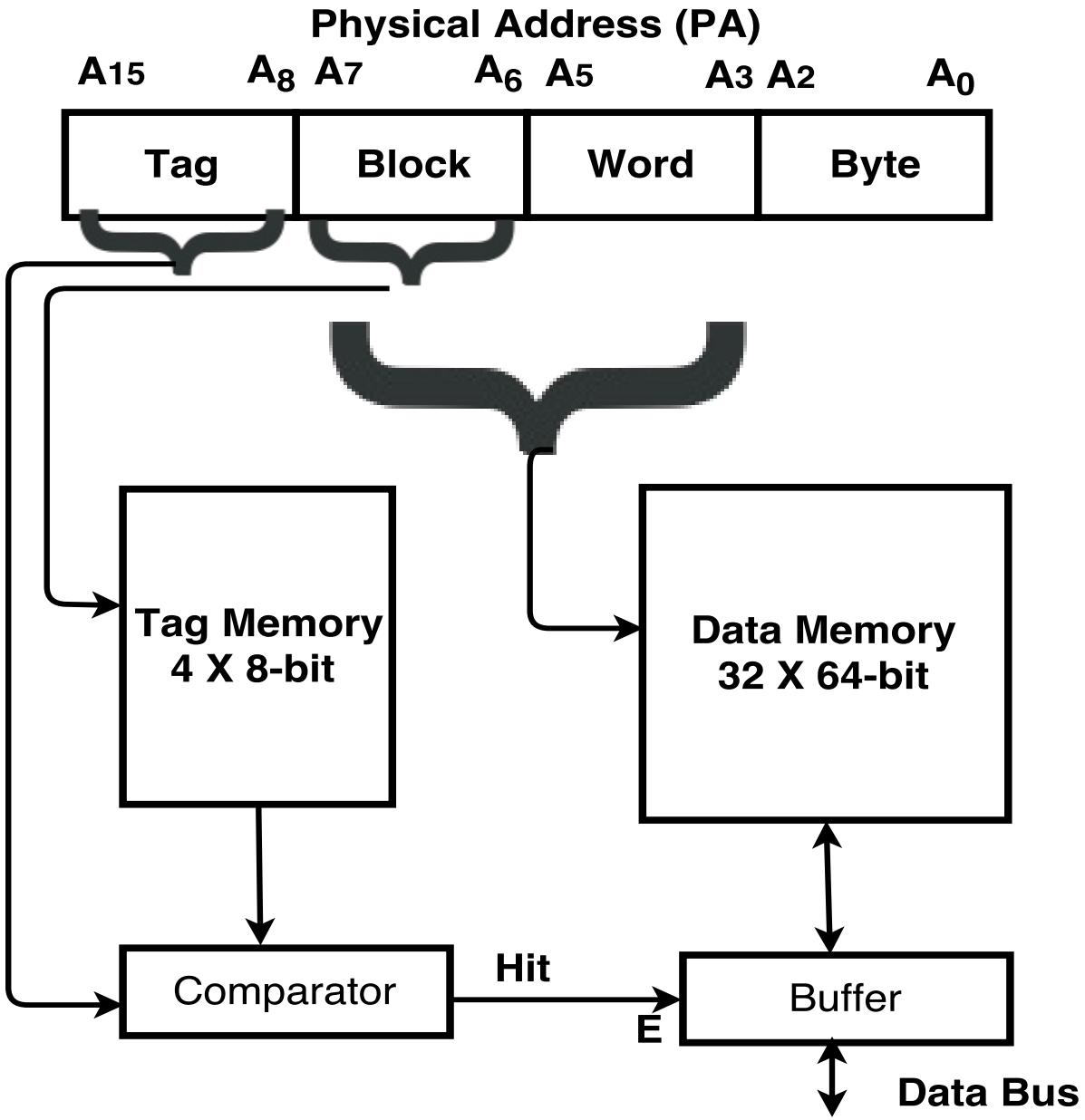
MM Block 'i' maps to Block (i mod 4) of CM.

Physical Addresses

Try to answer the following

- If word length is 64-bits, how many bytes in a word.
- If block size is 64 bytes, how many words in a block.
- If a cache size is 256 bytes, how many blocks in the cache.
- If main memory size is 64KB, how many blocks in the main memory.
- How many MMBs are mapped to single CMB?.

Direct Mapping (1-Way Set Associative Mapping)



Try to answer the following

- If word length is 32-bits, how many bytes in a word.
- If block size is 32 bytes, how many words in a block.
- If a cache size is 256KB, how many blocks in the cache.
- If main memory size is 4GB(Physical Address Space is 4GB) , how many blocks in the main memory.
- How many MMBs are mapped to single CMB?.

Average Memory Access Time (AMAT)

1. Let 'h' be the hit ratio of a cache, 'M' be the time to access information from main memory(MM), and 'C' be the time to access information in the cache.
   AMAT $= T_{avg} = hC + (1 - h)M$
2. Let us assume that there are two computers A and B.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has a cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.
3. Q1: Then How much time A and B take, to execute 100 instructions (assume that no instruction requires a read or a write operation).
4. Q1: Then How much time A and B take, to execute 100 instructions (assume that 30 instructions requires a read or a write operation).

Average Memory Access Time (AMAT)

1. Let 'h' be the hit ratio of a cache, 'M' be the time to access information from main memory(MM), and 'C' be the time to access information in the cache.
   AMAT $= T_{avg} = hC + (1 - h)M$

2. Let us assume that there are two computers A and B.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has a cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.

3. Q1: Then How much time A and B take, to execute 100 instructions (assume that no instruction requires a read or a write operation).

4. Q1: Then How much time A and B take, to execute 100 instructions (assume that 30 instructions requires a read or a write operation).

Average Memory Access Time (AMAT)

1. Let 'h' be the hit ratio of a cache, 'M' be the time to access information from main memory(MM), and 'C' be the time to access information in the cache.
   AMAT $= T_{avg} = hC + (1 - h)M$

2. Let us assume that there are two computers A and B.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has a cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.

3. Q1: Then How much time A and B take, to execute 100 instructions (assume that no instruction requires a read or a write operation).

4. Q1: Then How much time A and B take, to execute 100 instructions (assume that 30 instructions requires a read or a write operation).

Average Memory Access Time (AMAT)

1. Let 'h' be the hit ratio of a cache, 'M' be the time to access information from main memory(MM), and 'C' be the time to access information in the cache.
   AMAT $= T_{avg} = hC + (1 - h)M$

2. Let us assume that there are two computers A and B.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has a cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.

3. Q1: Then How much time A and B take, to execute 100 instructions (assume that no instruction requires a read or a write operation).

4. Q1: Then How much time A and B take, to execute 100 instructions (assume that 30 instructions requires a read or a write operation).

Average Memory Access Time (AMAT)

1. Let 'h1' be the hit ratio of L1 cache, 'h2' be the hit ratio of L2 cache, 'M' be a MM access time, 'C1' be L1-CM access time, and 'C2' be L2-CM access time.
   AMAT $= T_{avg} = h1.C1 + (1 - h1).h2.C2 + (1 - h1).(1 - h2).M$

2. Let us assume that there are three computers A,B, and C.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has L1 cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.
   C has L1 and L2 caches. L1 cache with hit rate of 90% and its access time is 10ns. L2 cache with hit rate of 99% and its access time is 30ns Each memory access from MM takes 130ns.

3. Q1: Then How much time A,B and C take, to execute 100 instructions (assume that no instruction requires a read or a write operation).

Average Memory Access Time (AMAT)

1. Let 'h1' be the hit ratio of L1 cache, 'h2' be the hit ratio of L2 cache, 'M' be a MM access time, 'C1' be L1-CM access time, and 'C2' be L2-CM access time.
   AMAT $= T_{avg} = h1.C1 + (1 - h1).h2.C2 + (1 - h1).(1 - h2).M$

2. Let us assume that there are three computers A,B, and C.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has L1 cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.
   C has L1 and L2 caches. L1 cache with hit rate of 90% and its access time is 10ns. L2 cache with hit rate of 99% and its access time is 30ns Each memory access from MM takes 130ns.

3. Q1: Then How much time A,B and C take, to execute 100 instructions (assume that no instruction requires a read or a write operation).

Average Memory Access Time (AMAT)

1. Let 'h1' be the hit ratio of L1 cache, 'h2' be the hit ratio of L2 cache, 'M' be a MM access time, 'C1' be L1-CM access time, and 'C2' be L2-CM access time.
   AMAT $= T_{avg} = h1.C1 + (1 - h1).h2.C2 + (1 - h1).(1 - h2).M$

2. Let us assume that there are three computers A,B, and C.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has L1 cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.
   C has L1 and L2 caches. L1 cache with hit rate of 90% and its access time is 10ns. L2 cache with hit rate of 99% and its access time is 30ns Each memory access from MM takes 130ns.

3. Q1: Then How much time A,B and C take, to execute 100 instructions (assume that no instruction requires a read or a write operation).

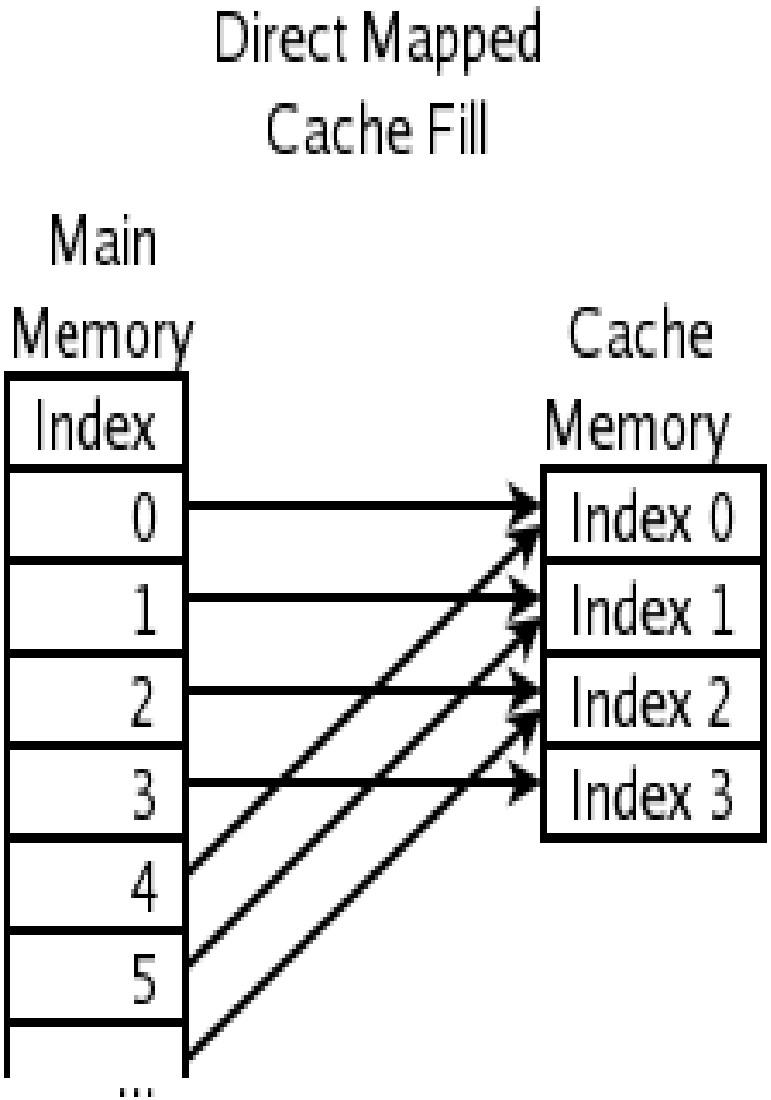
Average Memory Access Time (AMAT)

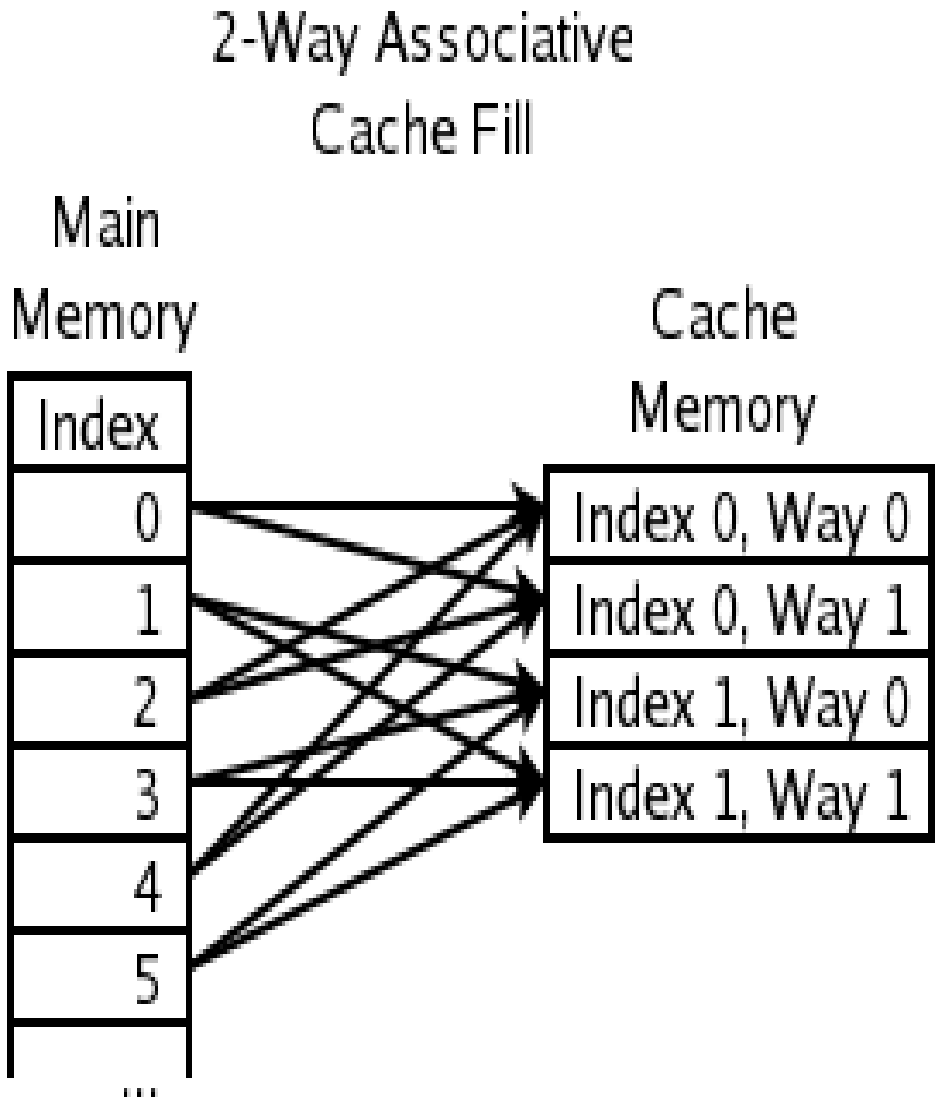
1. Let 'h1' be the hit ratio of L1 cache, 'h2' be the hit ratio of L2 cache, 'M' be a MM access time, 'C1' be L1-CM access time, and 'C2' be L2-CM access time.
   AMAT $= T_{avg} = h1.C1 + (1 - h1).h2.C2 + (1 - h1).(1 - h2).M$

2. Let us assume that there are three computers A,B, and C.
   A has no cache, the processor takes 100ns (nano seconds) for each memory access.
   B has L1 cache with hit rate of 90% and its access time is 10ns. Each memory access from MM takes 110ns.
   C has L1 and L2 caches. L1 cache with hit rate of 90% and its access time is 10ns. L2 cache with hit rate of 99% and its access time is 30ns Each memory access from MM takes 130ns.

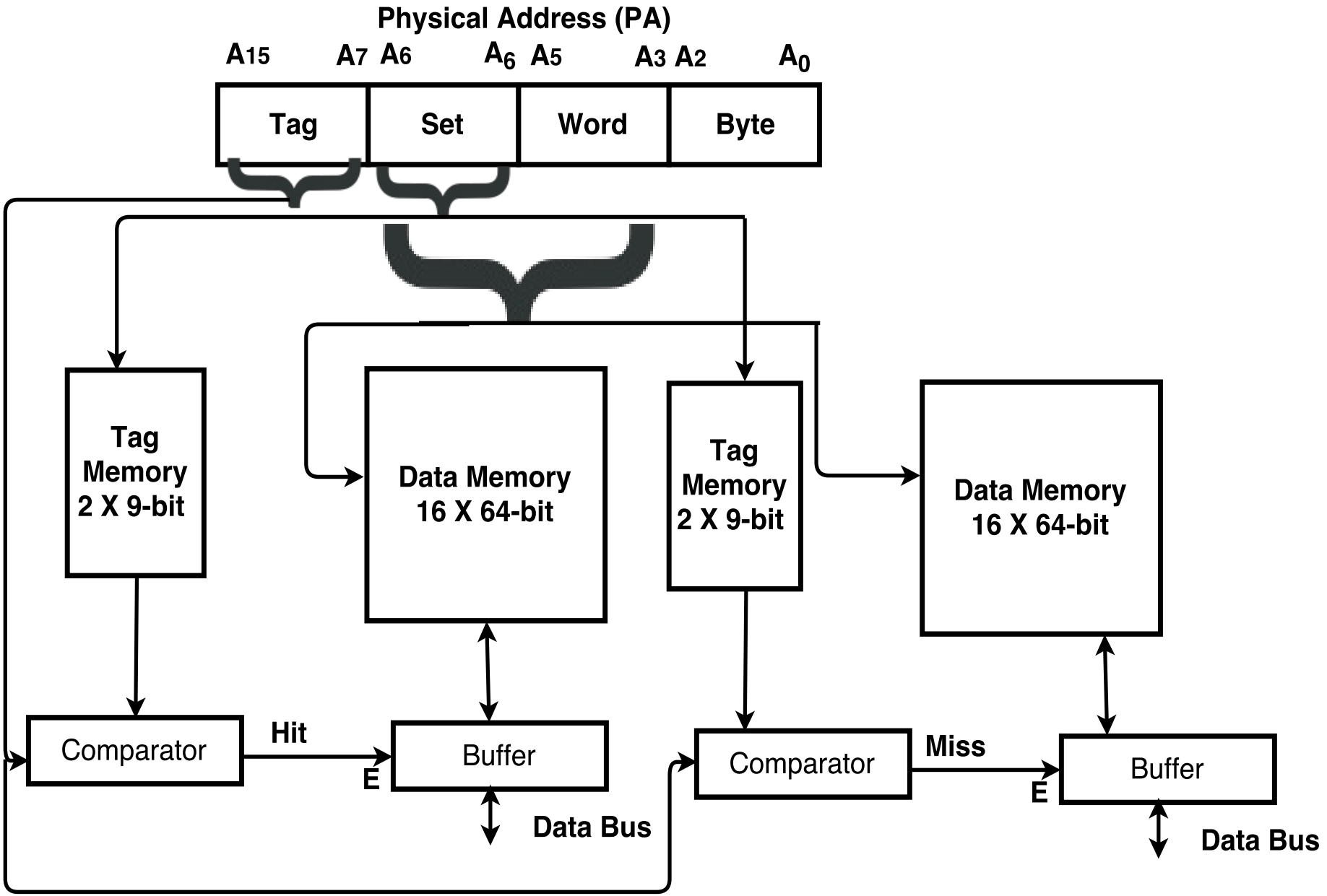
3. Q1: Then How much time A,B and C take, to execute 100 instructions (assume that no instruction requires a read or a write operation).

Difference between Direct and 2-Way Set Associative Mappings





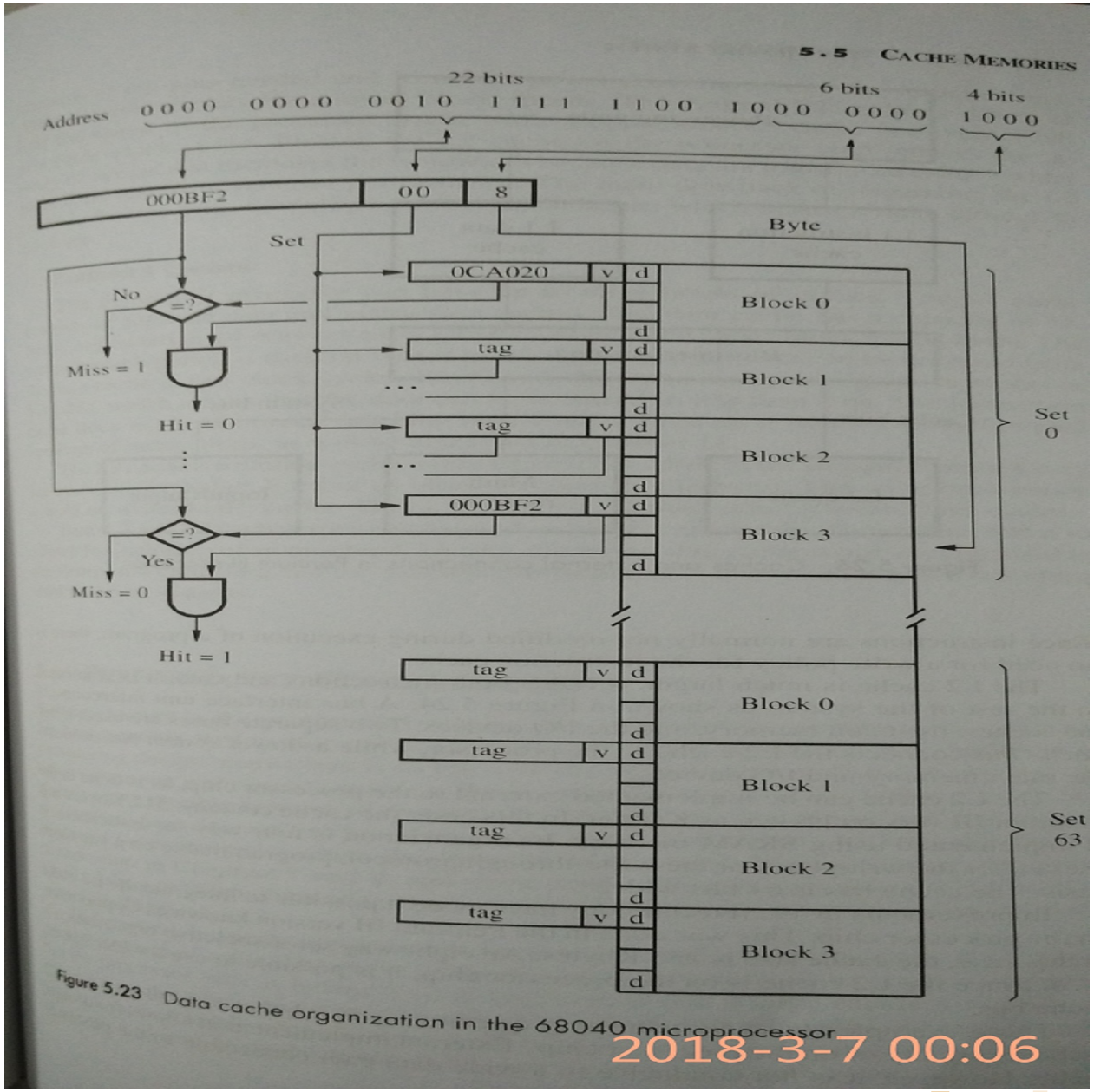
2-Way Set Associative Mapping



Try to answer the following

- If word length is 32-bits, how many bytes in a word.
- If block size is 32 bytes, how many words in a block.
- If main memory size is 4GB(Physical Address Space is 4GB) , how many blocks in the main memory.
- If a cache size is 256KB, how many blocks in the cache.
- If a cache supports 2-way set associative, how many MMBs are mapped to a single CMB?.
- If a cache supports 4-way set associative, how many MMBs are mapped to a single CMB?.

4-way Mapping

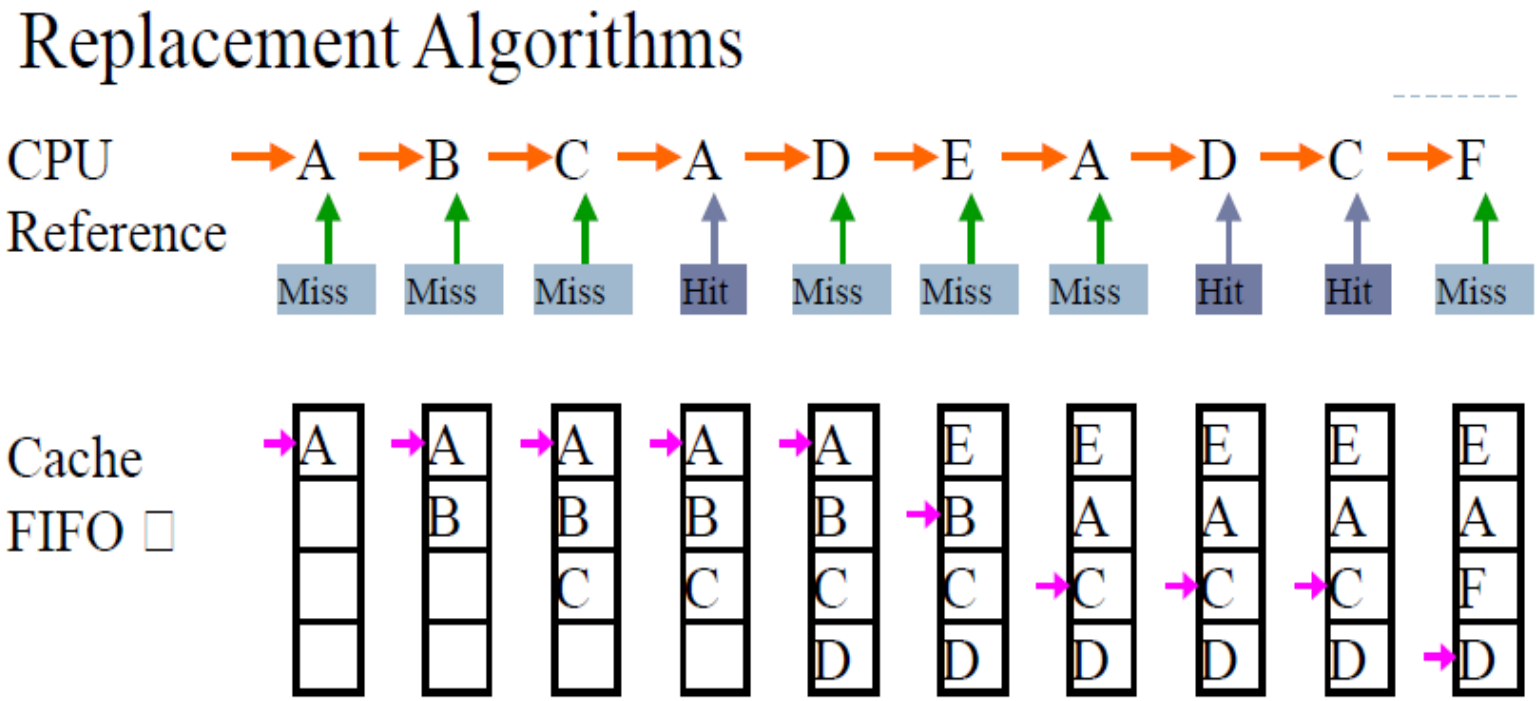


Cache miss is a state where the data requested for processing is not found in the cache memory.

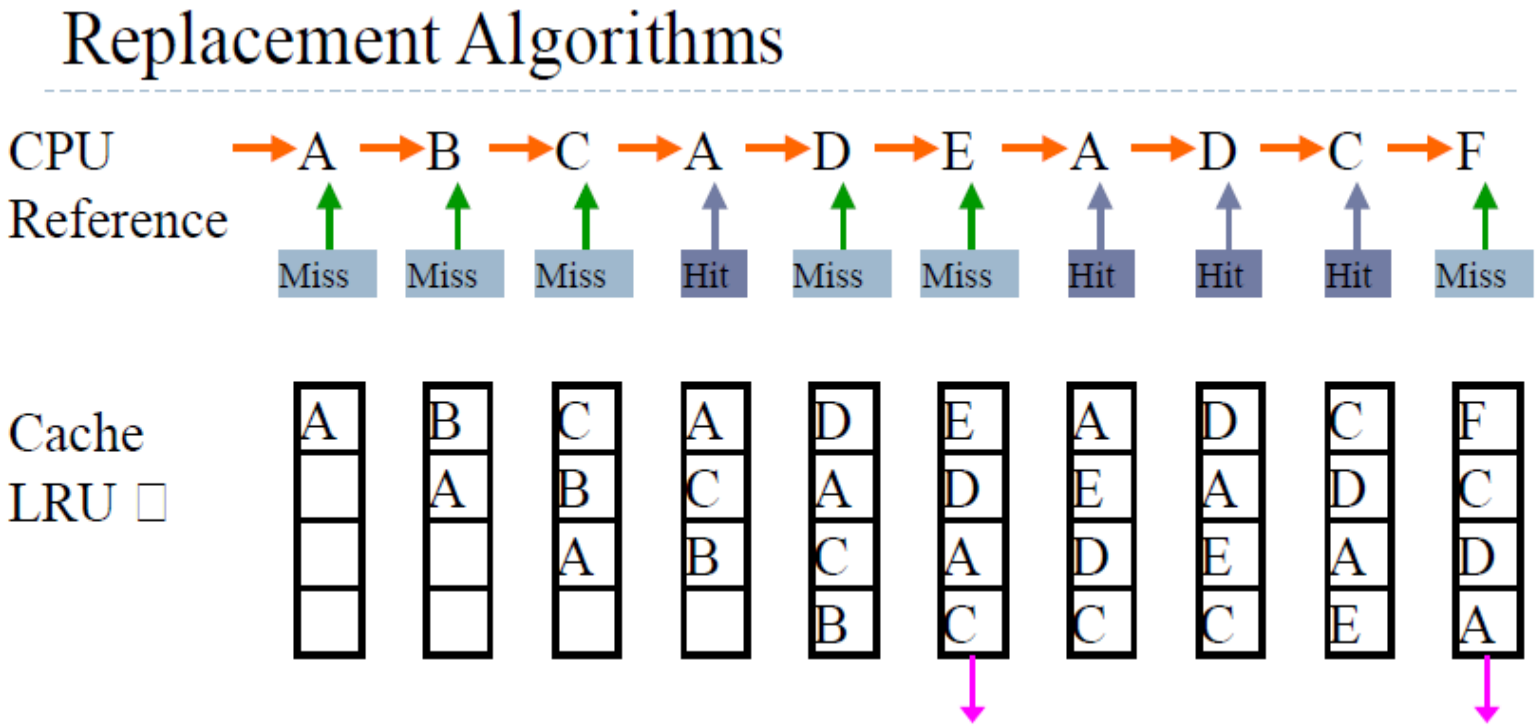
Types of Cache Misses

1. **Compulsory or Cold Misses**: The first reference to a block of memory, starting with an empty cache.
2. **Capacity Misses**: The cache is not big enough to hold every block you want to use.
3. **Conflict Misses**: Two blocks are mapped to the same location and there is not enough room to hold both.

FIFO Replacement Algorithm



Least Recently Used (LRU) Replacement Algorithm



Important Points

- **Valid bit** says whether the cache block has a valid data or not.
- **Dirty bit (modify bit)** says whether the contents of the cache line/block are different to what are there in main memory.
- Inclusive Cache: $L1 \subset L2 \subset L3$
- Exclusive Cache: $L1 \cap L2 \cap L3 = \varnothing$
- Non-inclusive Cache : $(L1 \cap L2 = \varnothing)$ and $((L1 \cup L2) \cap L3 = L1 \cup L2)$

All Mappings

One-way set associative (direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

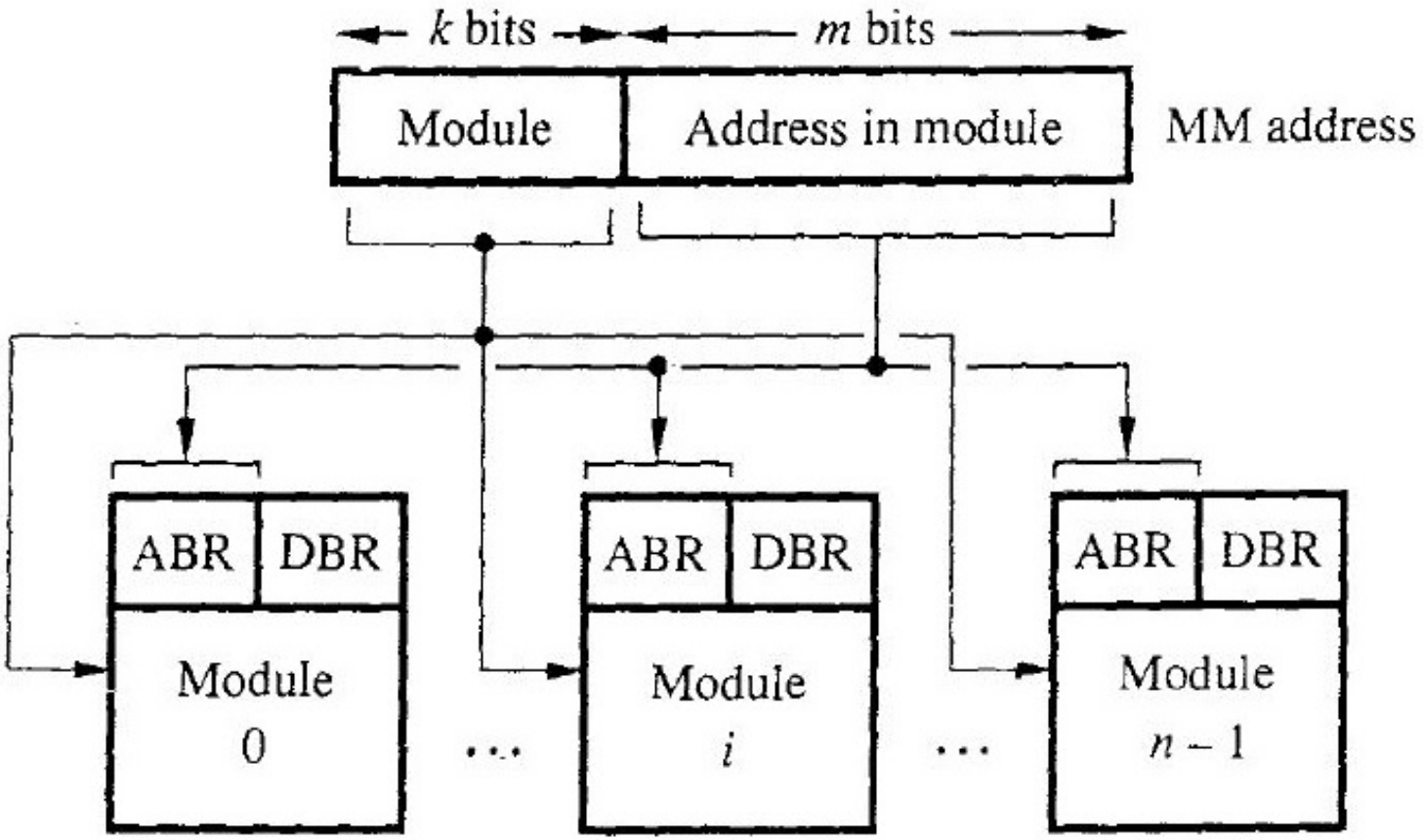
Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

Interleaving

Interleaving is a technique for improving the speed of access.

Our objective - transfer data blocks from MM to CM.



(a) Consecutive words in a module

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
  - One clock cycle to send a address to MM.
  - First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
  - Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Calculate the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.

Properties of H/W:

- Cache with 8-word blocks.
- On a read miss, the block contains the desired word must be copied from the MM into the CM.
- One clock cycle to send a address to MM.
- First word is accessed in 8 clock cycles and subsequent words are accessed in 4 clock cycles.
- Once clock cycle to send one word to the CM.

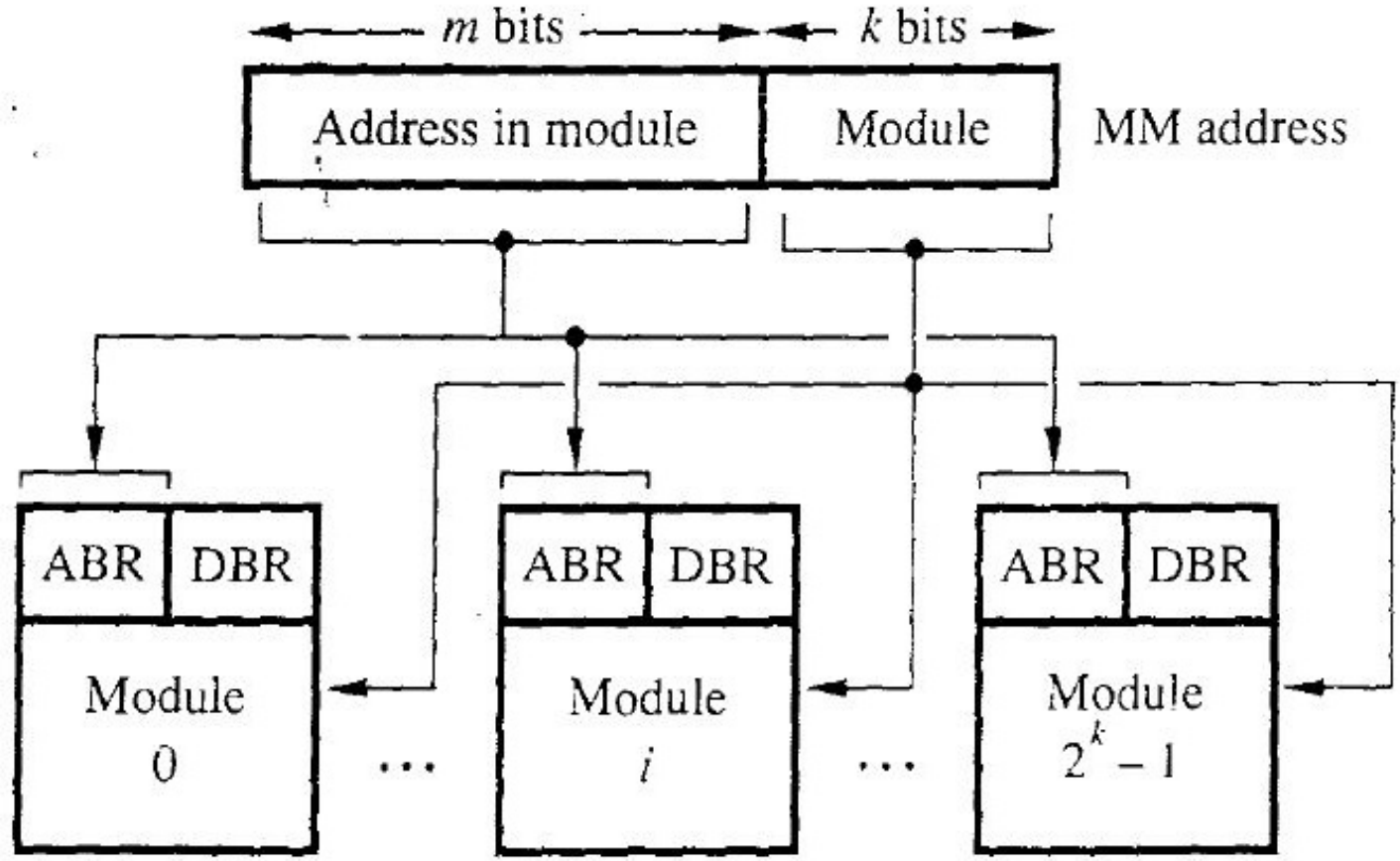
If consecutive words in a module, then how much time it takes.

If consecutive words in consecutive modules, then how much time it takes.

Interleaving

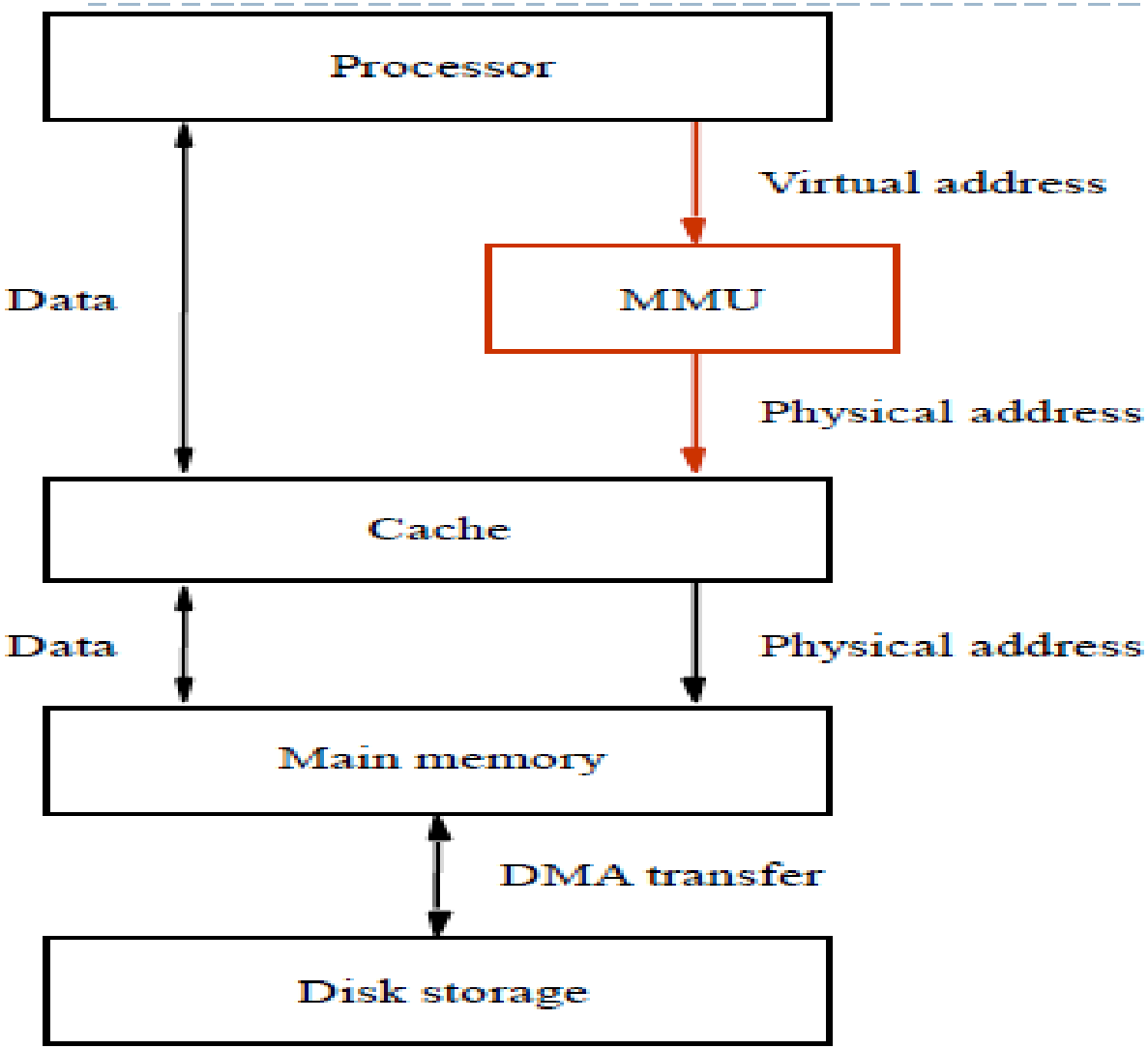
Interleaving is a technique for improving the speed of access.

Our objective - transfer data blocks from MM to CM.



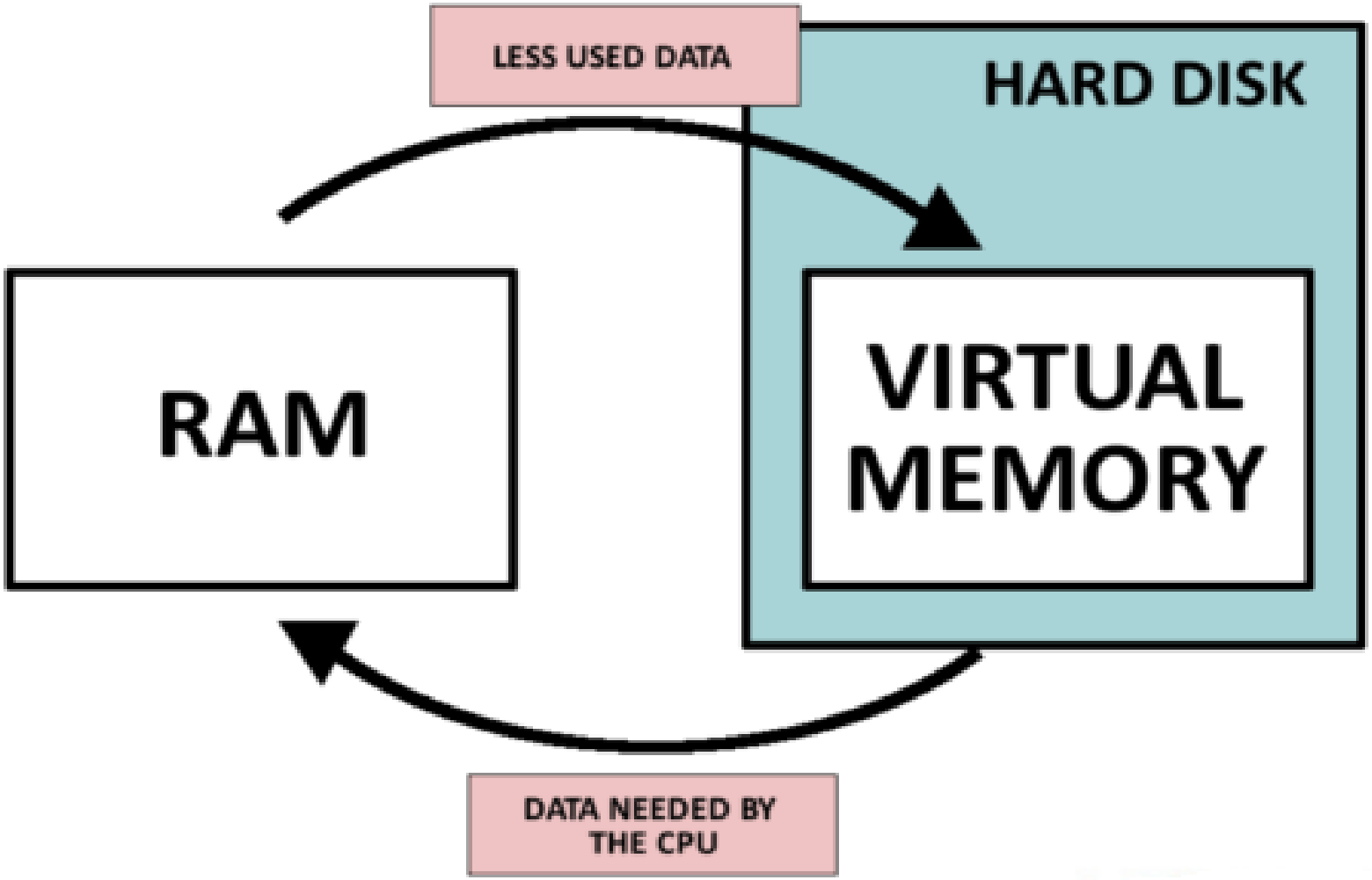
(b) Consecutive words in consecutive modules

Addressing Processor References

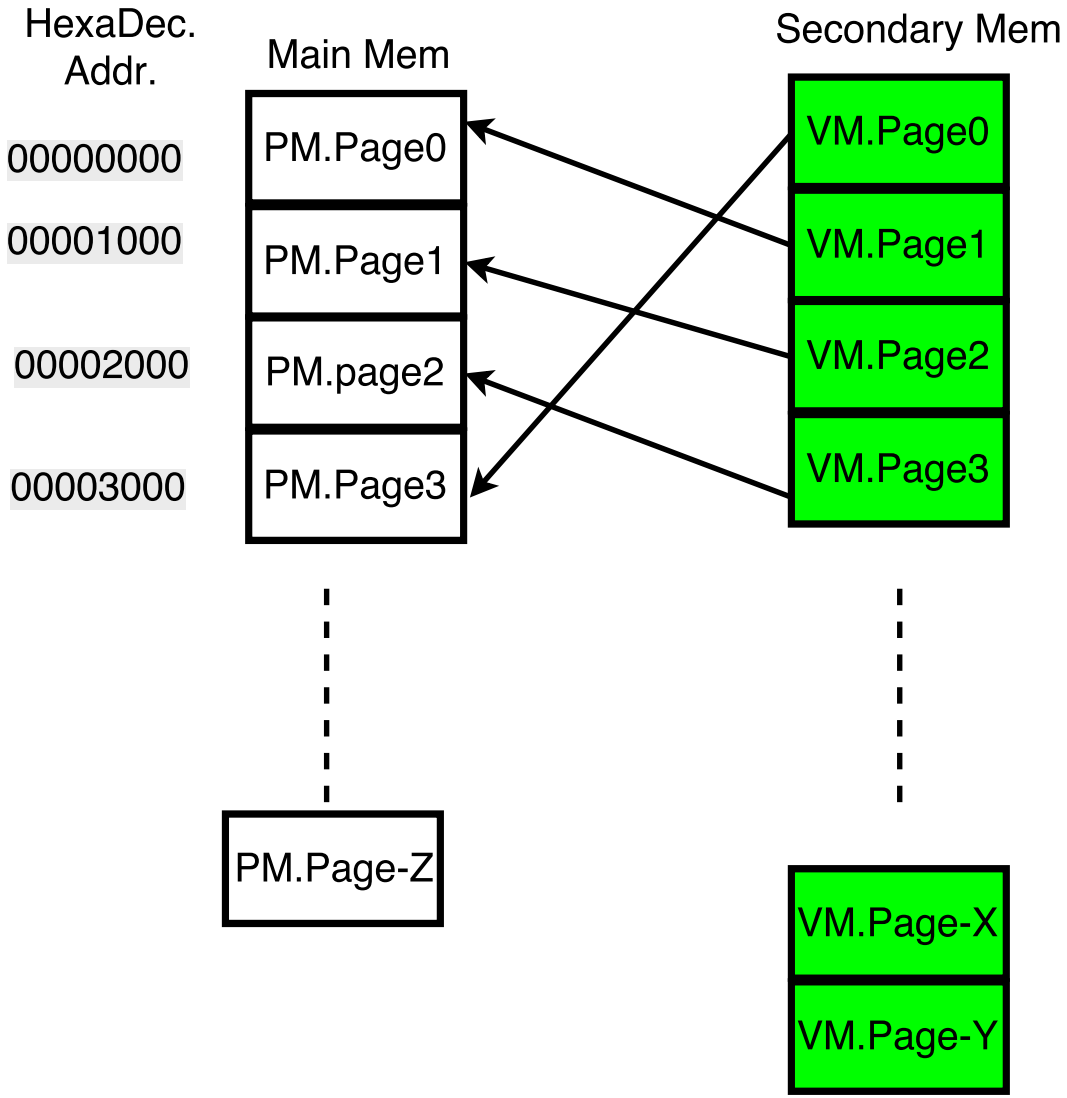


Virtual Memory

A technique for moving data between MM and Secondary storage device.



Mapping from a VM-Page to a PM-Page or MM-Page (PM-Page is also called a Page Frame)



Try to answer the following

- If size of a page is 4KB, how many bits are required to identify a byte in the page.
- If size of a program is 2GB, how many pages are required to store the program.
- If the size of MM is 512MB, how many pages it can accommodate.
- If the size of virtual address space is 4GB, how the VA is converted to a physical address.

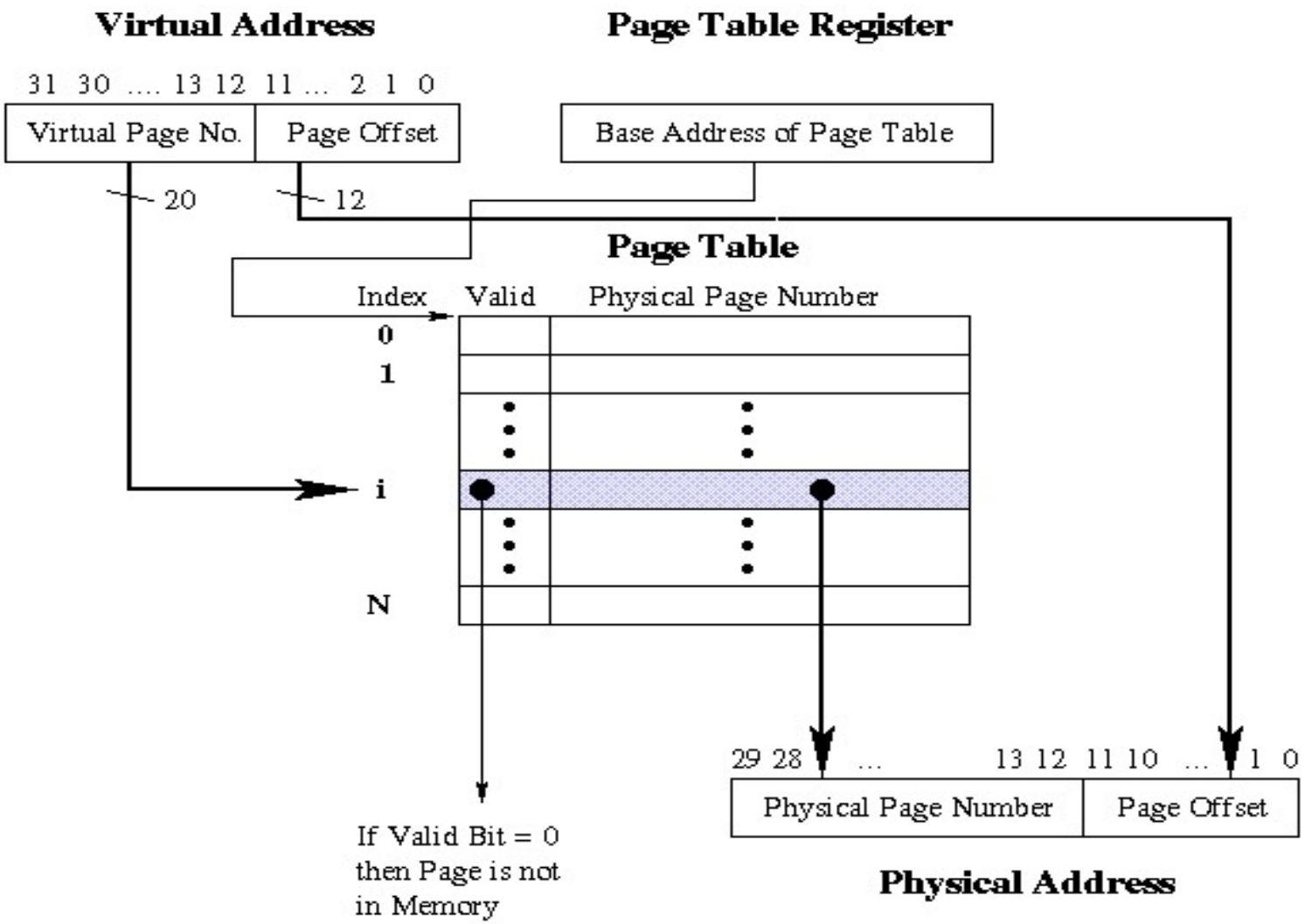
Mapping Table

index	Valid Bit	Physical Page Number
0	1	3
1	1	0
2	1	1
3	1	2

Table 1: Page Table.

Assume that base address of page table is available in PTBR (Page Table Base Register).

Translation of VA to PA



Translation Lookaside Buffer(TLB)

Small Cache for a Page Table (available in MMU).

It has information about most recently accessed pages.