MEMORY SYSTEM DESIGN

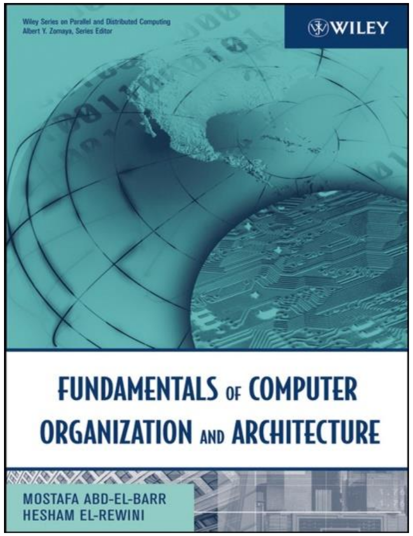
Danang Wahyu Utomo
danang.wu@dsn.dinus.ac.id
+6285 725 158 327

RENCANA KEGIATAN PERKULIAHAN SEMESTER

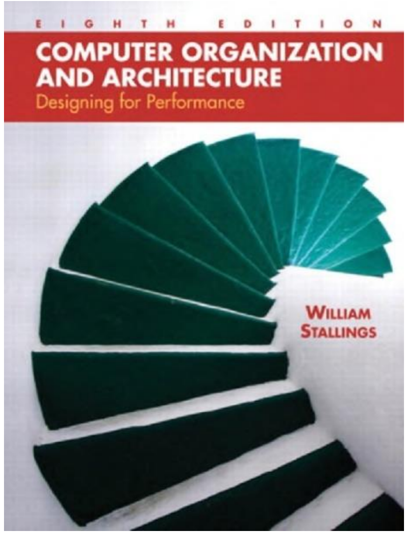
W	Pokok Bahasan
1	Organisasi dan Arsitektur Komputer
2	Sistem Komputer
3	Instruction Set Architecture and Design
4	
5	Computer Arithmetic
6	
7	Review Materi 1-6
8	**Ujian Tengah Semester**
9	Desain Unit Pemrosesan
10	
11	**Desain Sistem Memory**
12	
13	Desain dan Organisasi Input Output
14	
15	Teknik Desain Pipelining
16	**Ujian Akhir Semester**

Reference

- Mustafa Abd-el-Bhar, Hesham El Rewini – Fundamentals of Computer Organization and Architecture 9th Edition (2004)


Wiley Series on Parallel and Distributed Computing
Albert Y. Zomaya, Series Editor
WILEY
FUNDAMENTALS OF COMPUTER ORGANIZATION AND ARCHITECTURE
MOSTAFA ABD-EL-BARR
HESHAM EL-REWINI

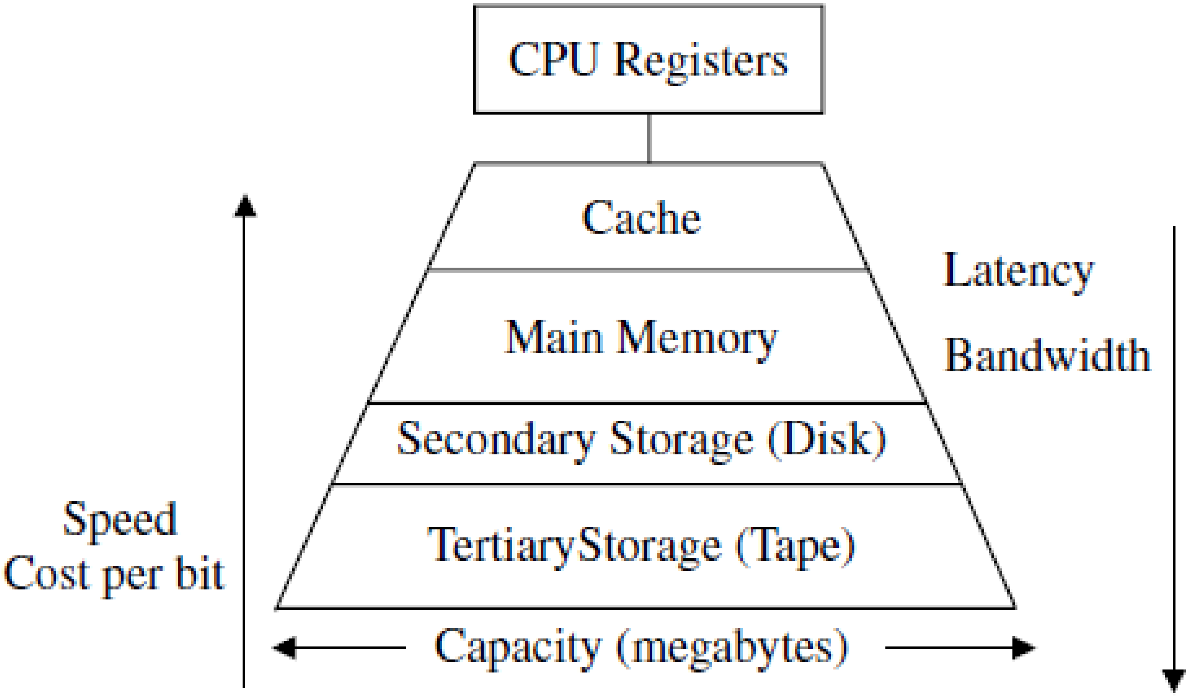
- William Stallings – Computer Organization and Architecture Designing For Performance 8th Edition (2009)


EIGHTH EDITION
COMPUTER ORGANIZATION AND ARCHITECTURE
Designing for Performance
WILLIAM STALLINGS

Content

- Cache Memory
- Main Memory
- Virtual Memory
- Read-Only Memory

Memory Hierarchy



Memory Hierarchy

- Cache : small, expensive, relatively fast unit
- Cache and memory are built using solid-state semiconductor
- Cache and memory are customary to call fast memory level (primary level)
- Larger, less expensive, far slower built using magnetic memories is typically the disk and the tape

Memory Hierarchy

- Memory hierarchy can be characterized by parameter :
  - Access type
  - Capacity
  - Cycle time
  - Latency
  - Bandwidth
  - cost

Hierarchy Memory

- The effectiveness of memory hierarchy depend on the principle of moving information into the fast memory and accessing it many time before replacing with new information, usually called *locality of reference*
- There are two locality : spatial and temporal
- Spatial locality refers to the address has ben referenced

  Example : straightline program
- Temporal locality refers to a particular memory item has been referenced

  Example : an instruction in program loop

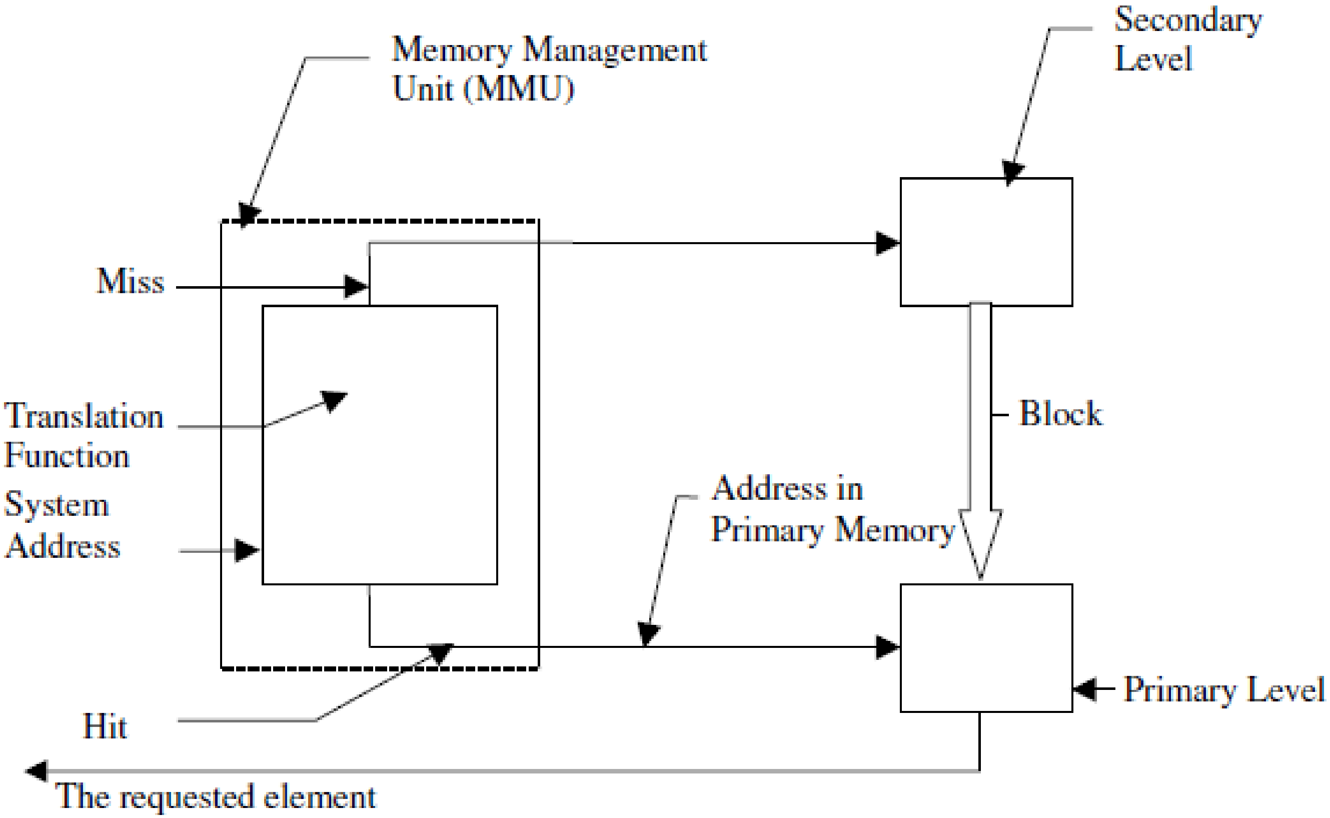
Cache Memory

- Hiding or storing the information
- When processor makes a request for a memory reference, the request is sought in the cache
- If the request corresponds to an element that is currently in the cache, it is called cache **hit**
- If the request corresponds to an element that is not currently in the cache, it is called cache **miss**
- When the requested element is not found in the cache, it has to be brought from a subsequent memory level in the hierarchy memory

Cache Memory – Mapping Function

- Request for accessing a memory element is made by the processor through issuing the address of the requested element
- Address issued by the processor may correspond to an element that exist in the cache; otherwise, an element residing in the memory
- Memory management unit (MMU) used to determine the whereboats of the requested element

Address Mapping Operation



Address Mapping Function

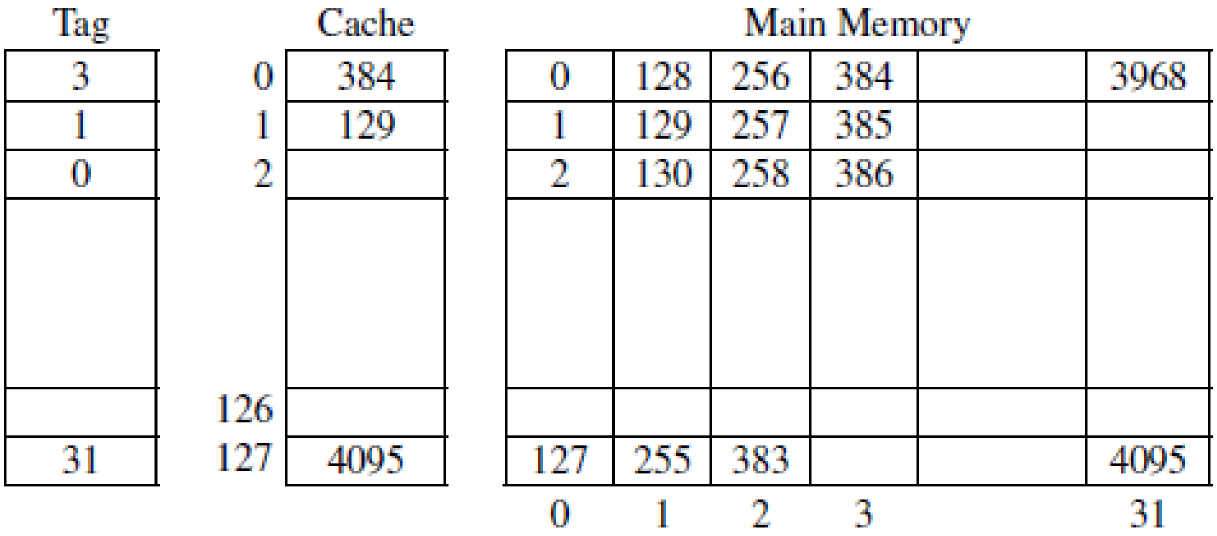
- If address translation reveals that the issued address corresponds to an element residing in the cache, the element will be made available to the processor

- If the element is not currently residing in the cache, it will be brought from the memory and place in the cache, the element will be made available to the processor

Direct Mapping

- Places an incoming main memory block into a specific fixed cache block location
- Its simplicity in determining where to place an incoming main memory block in the cache
- Inefficient use of the cache ; a number of main memory blocks may compete for a given cache block even if there exist other empty cache blocks.

Mapping Main Memory to Cache Blocks



Tag

3
1
0
31

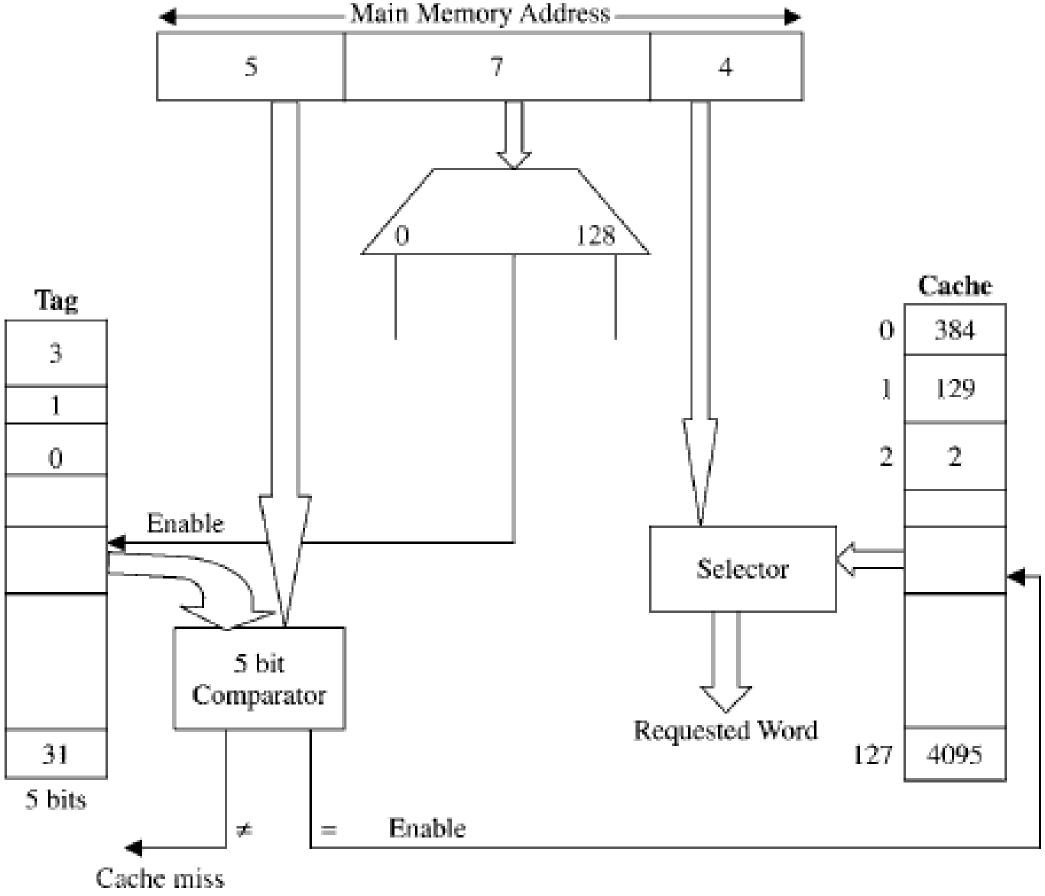
Cache

0	384
1	129
2	
126	
127	4095

Main Memory

0	128	256	384		3968
1	129	257	385		
2	130	258	386		
127	255	383			4095
0	1	2	3		31

Direct-Mapped Address



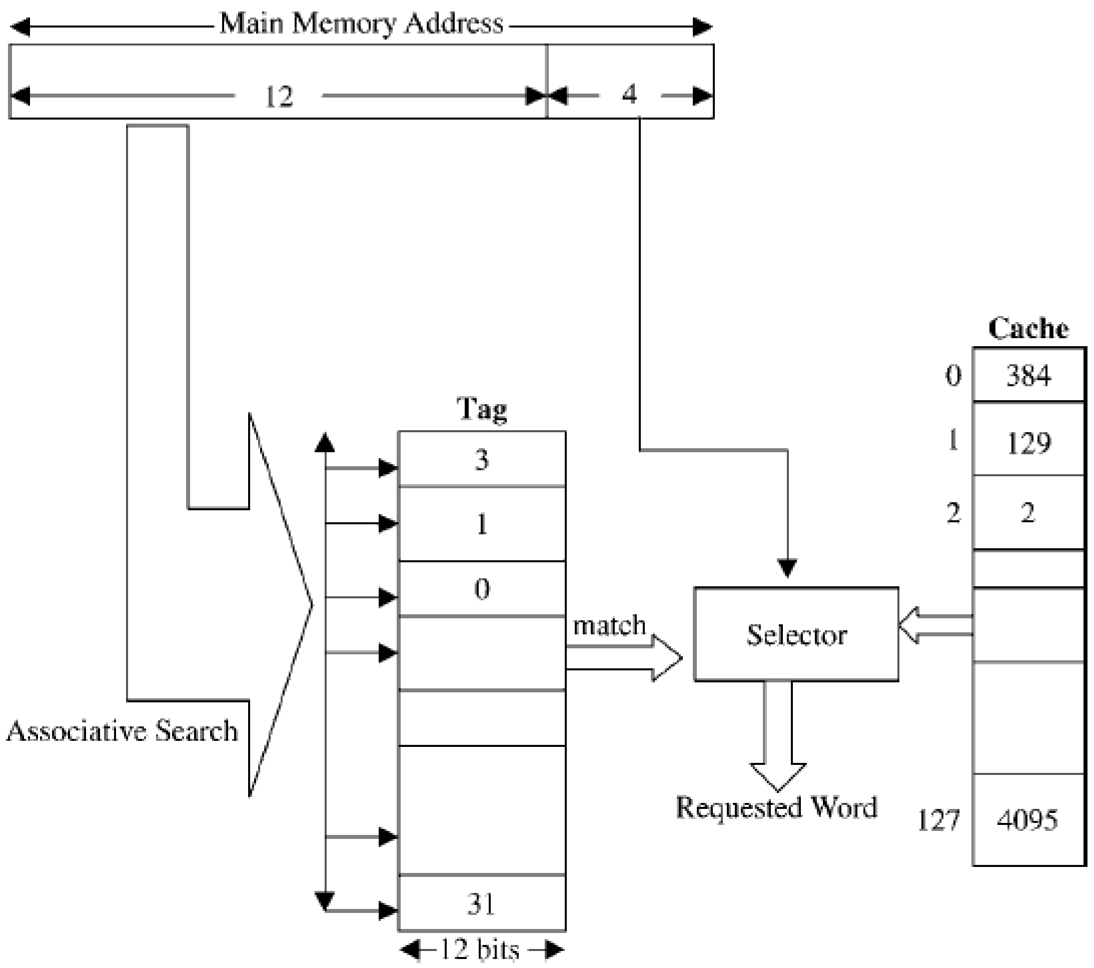
Associative Mapping

- An incoming main memory block can be placed In any available cache block
- The MMU interprets the address issued by the processor by dividing it into two field : **Tag Field** and **Word Field**
- Tag field

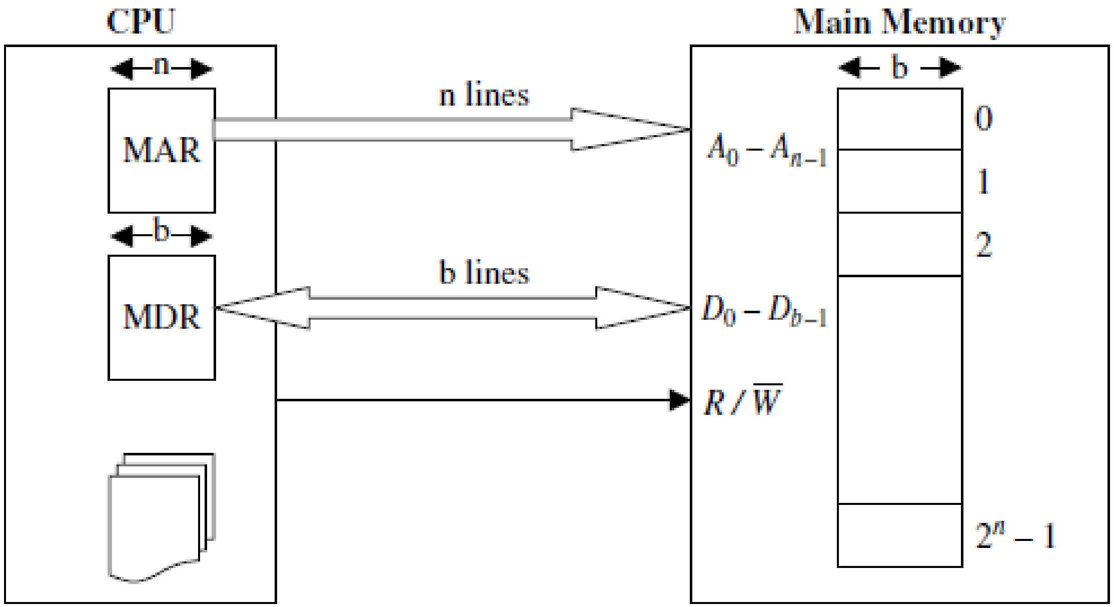
  identify the block while residing in the **cache**
- Word field

  identify the element **within the block** that is requested by the processor

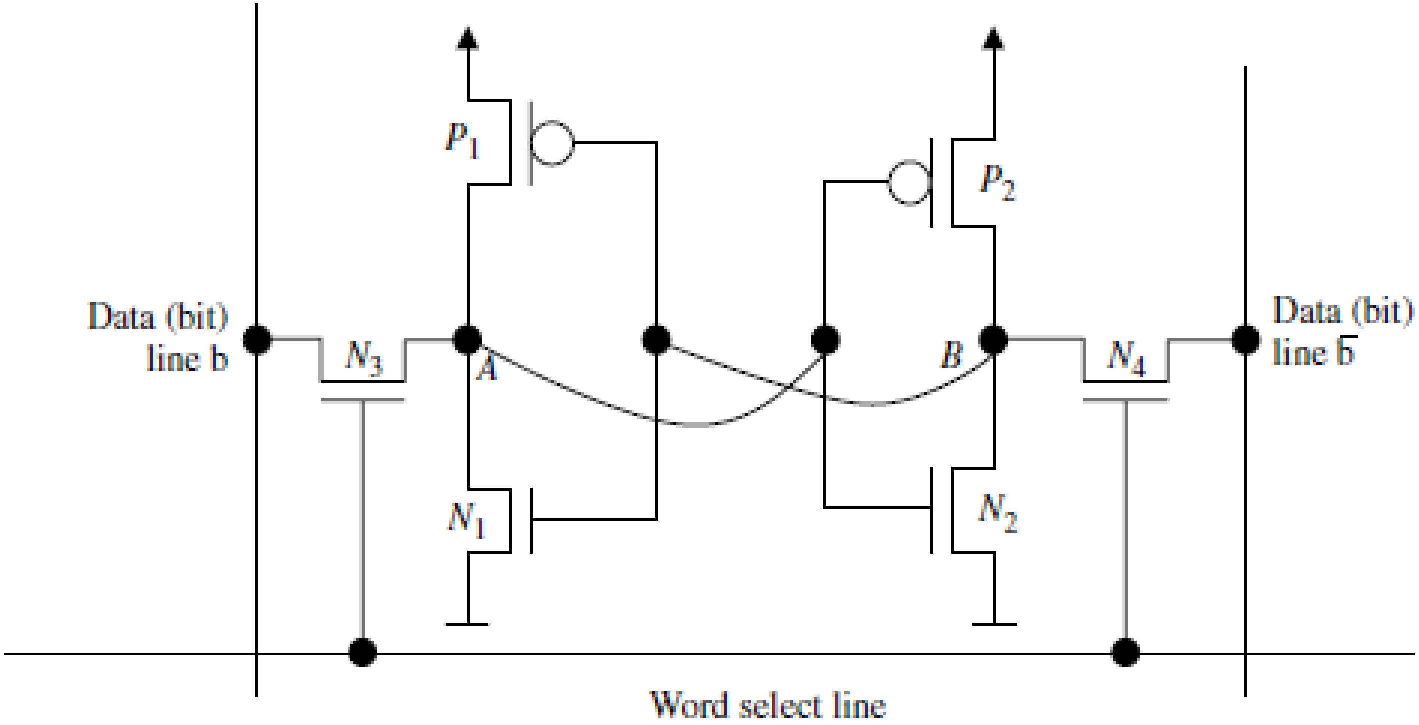
Associative-Mapped Address



Main Memory


CPU
n
MAR
b
MDR
n lines
b lines
Main Memory
b
$A_0 - A_{n-1}$
$D_0 - D_{b-1}$
$R / \overline{W}$
0
1
2
$2^n - 1$

Main Memory



Data (bit) line b
P_1
P_2
N_3
N_4
A
B
Data (bit) line $\bar{b}$
N_1
N_2
Word select line

Read Operation

- Both line *b* and *ƀ* are precharged high
- the word select line is activated, thus turning on both transistor N_3 and N_4
- Depending on the internal value stored in the cell, pointA(B) will lead to the discharge of line b(*ƀ*)

Write Operation

- The bit lines are precharged such that b(ƀ) = 1(0)
- The word select line is activated, thus turning on both transistor N_3 and N_4
- The bit line precharged with 0 will have to force the point A(B), which has 1 to 0

Virtual Memory

- The concept of virtual memory is similar to that of the cache memory
- Vritual memory attempts to optimize the use of the main memory with the hard disk
- Direct Mapping ???
- Associative Mapping ???

Read-Only Memory

- ???

TERIMA KASIH