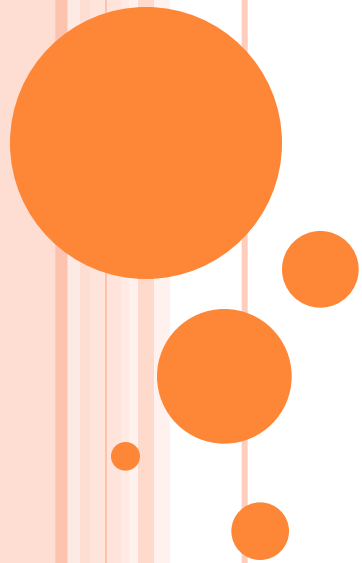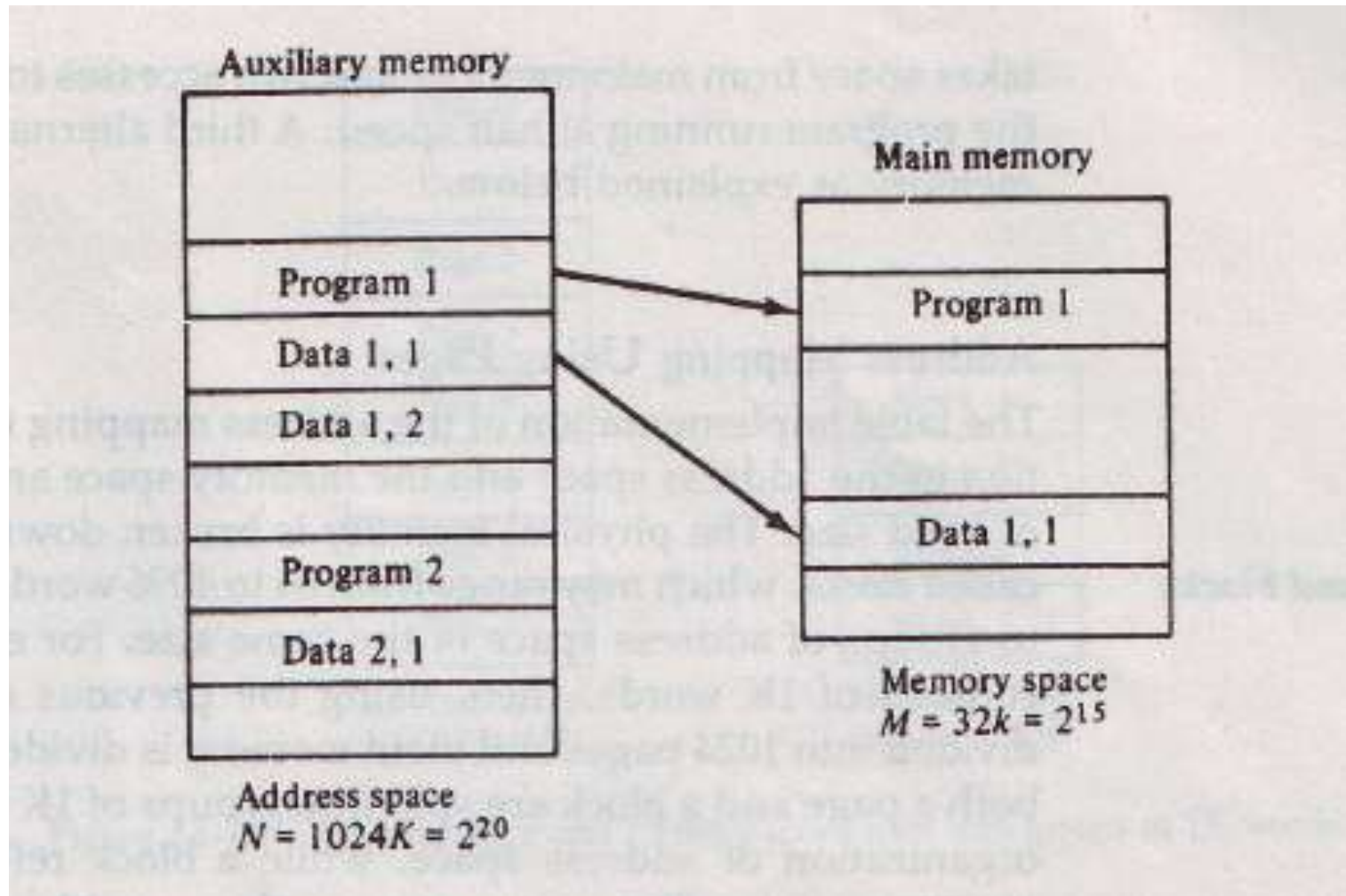# UNIT-4
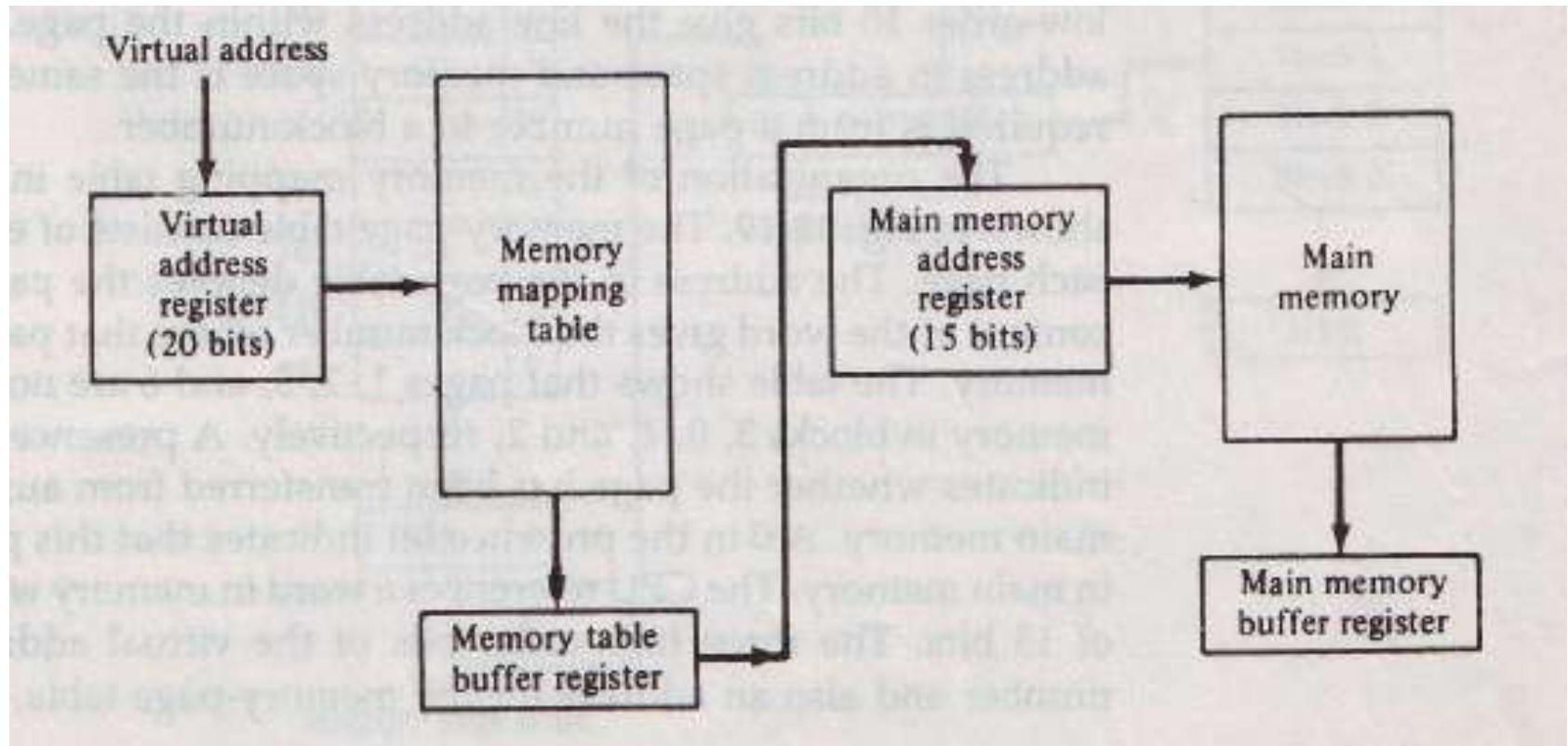
Memory Organization
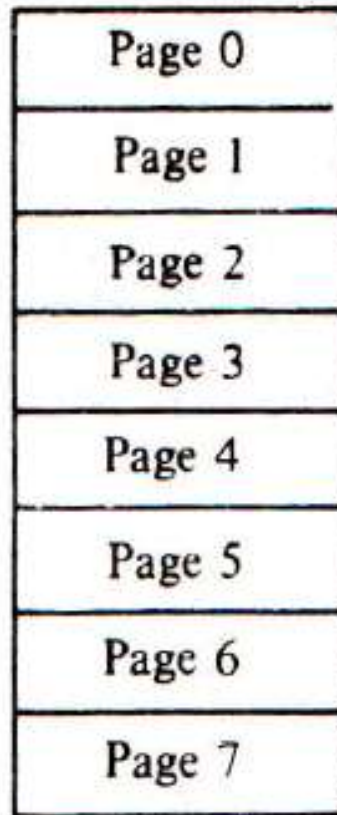
ADDRESS SPACE & MEMORY SPACE

MEMORY TABLE FOR MAPPING A VIRTUAL ADDRESS:

ADDRESS SPACE & MEMORY SPACE SPLIT INTO GROUPS:

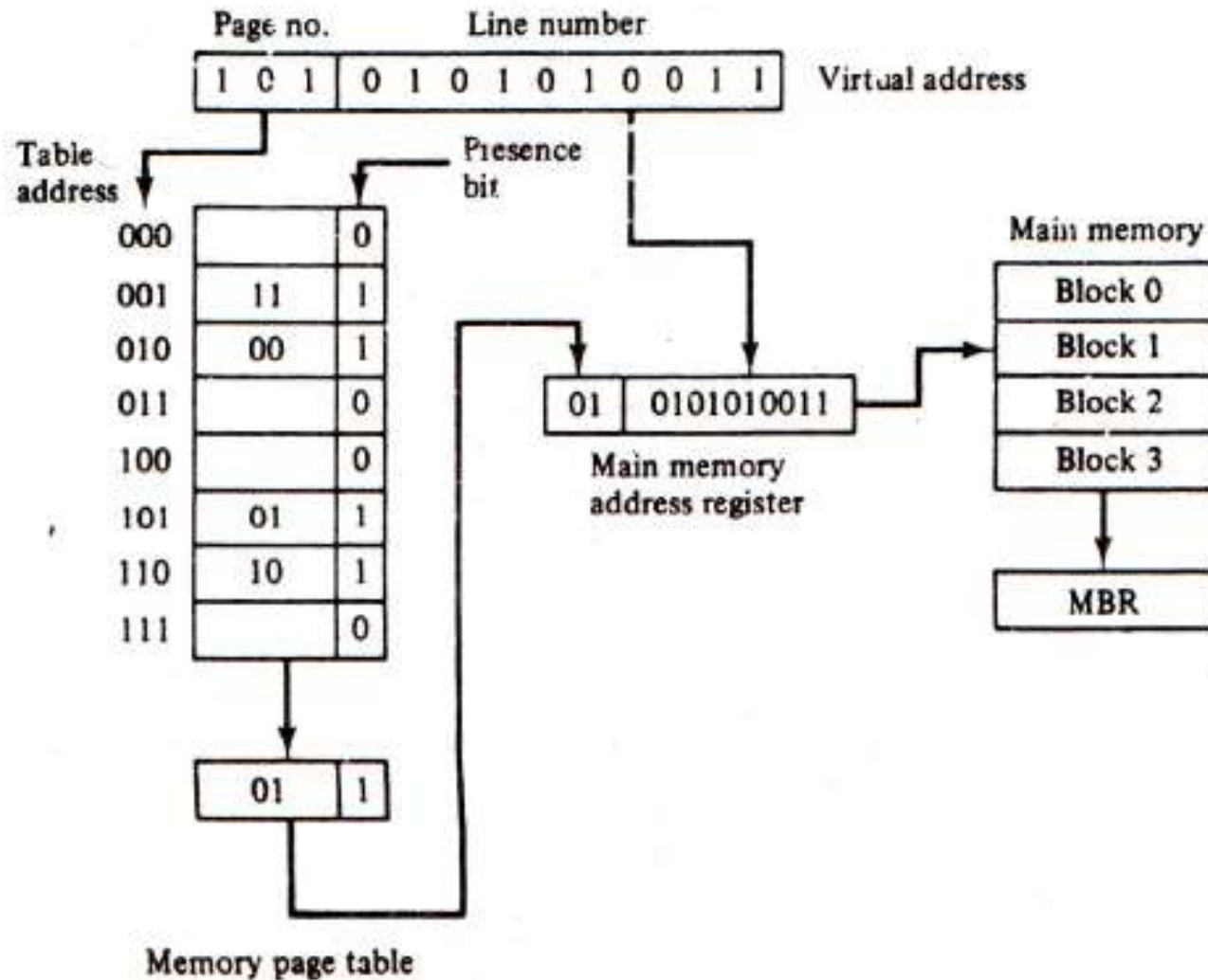| Page 0 |
|--------|
| Page 1 |
| Page 2 |
| Page 3 |
| Page 4 |
| Page 5 |
| Page 6 |
| Page 7 |

Address space
$N = 8K = 2^{13}$

Block 0
Block 1
Block 2
Block 3
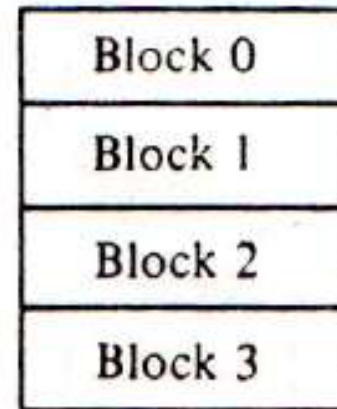
Memory space
$M = 4K = 2^{12}$

MEMORY TABLE IN A PAGED SYSTEM

Memory page table

AN ASSOCIATIVE MEMORY PAGE TABLE

Page No	Line Number

Argument Register

1 1 1	0 0

Key Register

001	11
010	00
101	01
110	10

Associative Memory

PAGE REPLACEMENT ALGORITHM

- FIFO
- LRU
- OPT

- Page No:
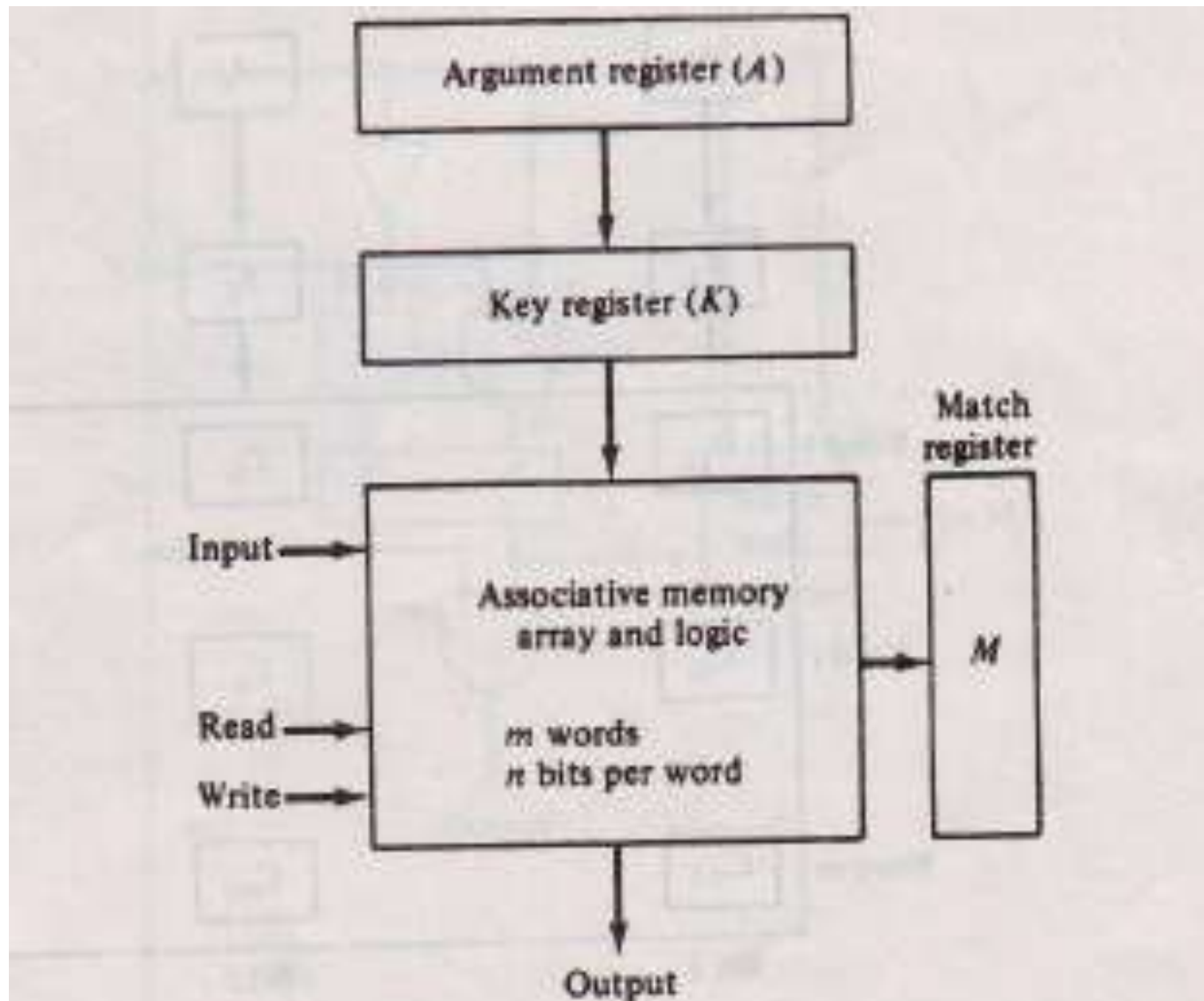
  2   3   2   1   5   2   4   5

  3   2   5   2

- Total No of Frames:-3
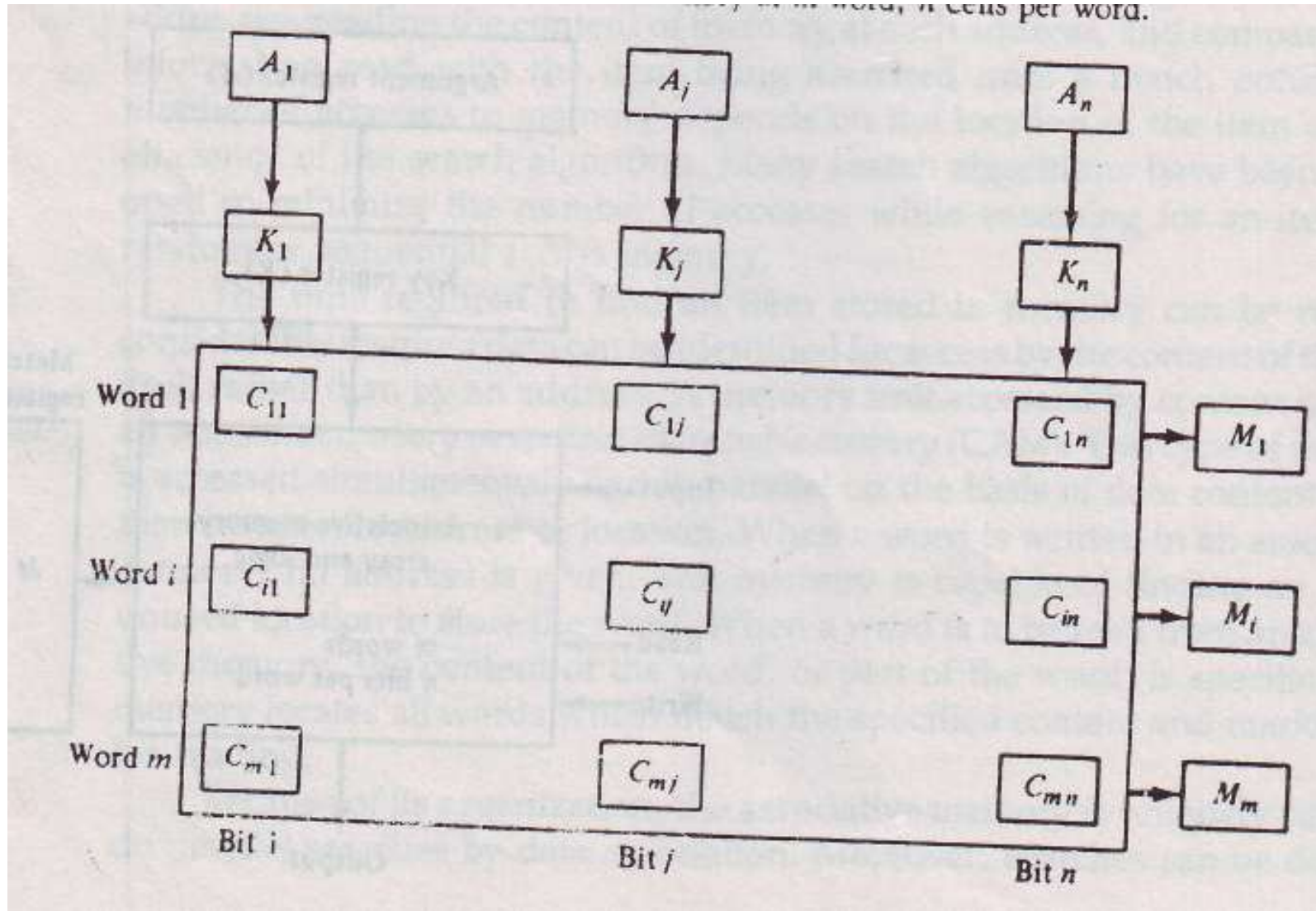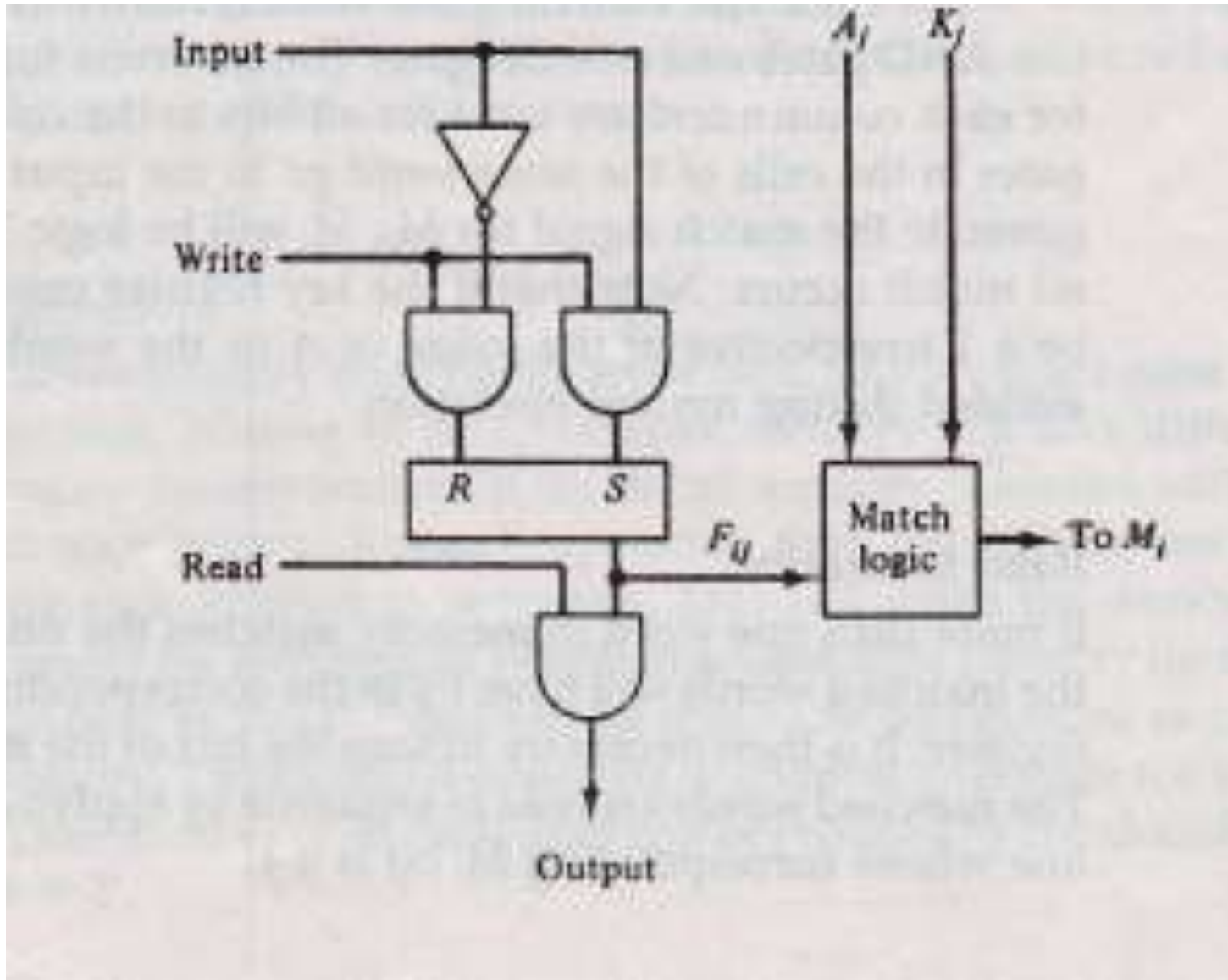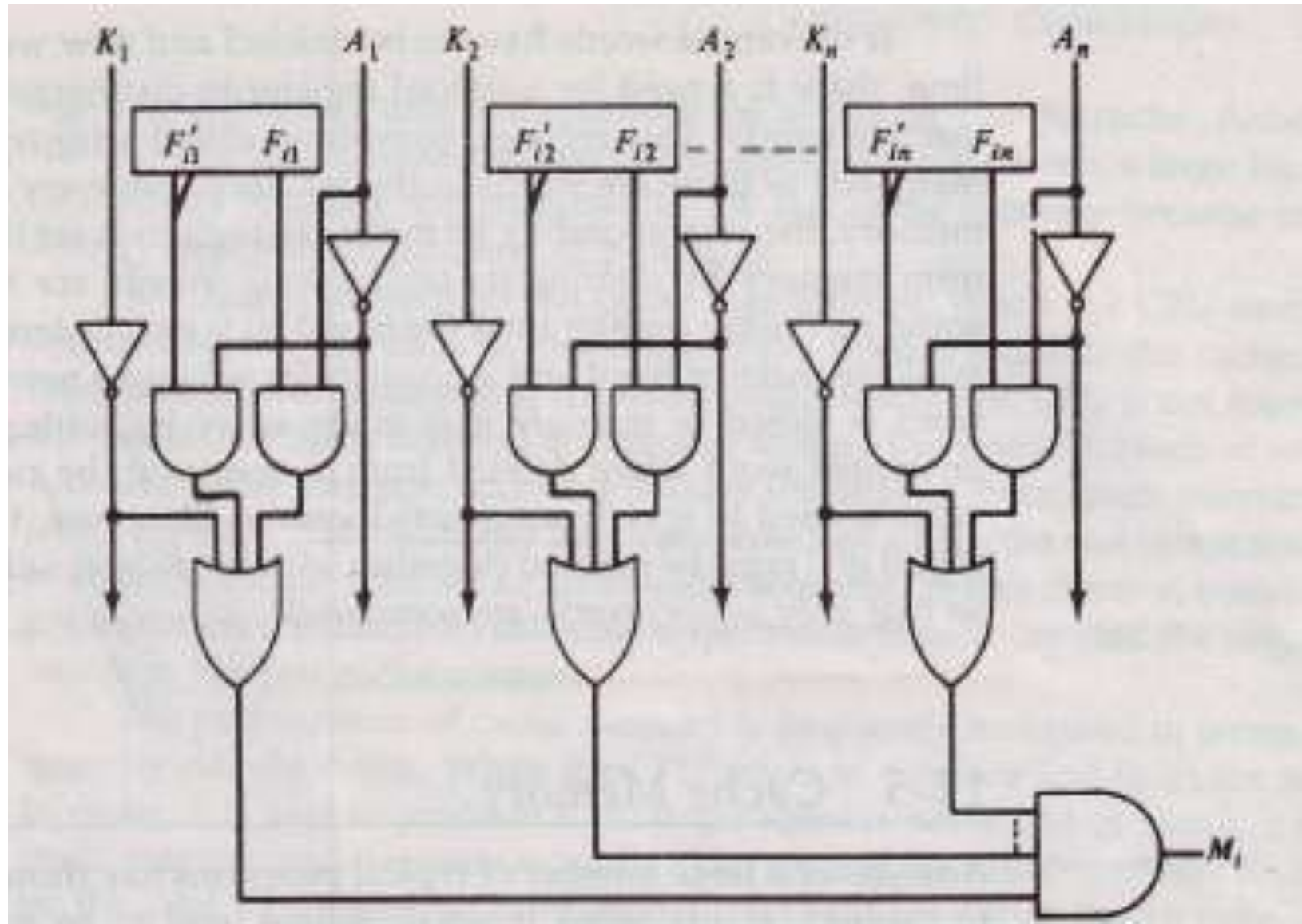
ASSOCIATIVE MEMORY

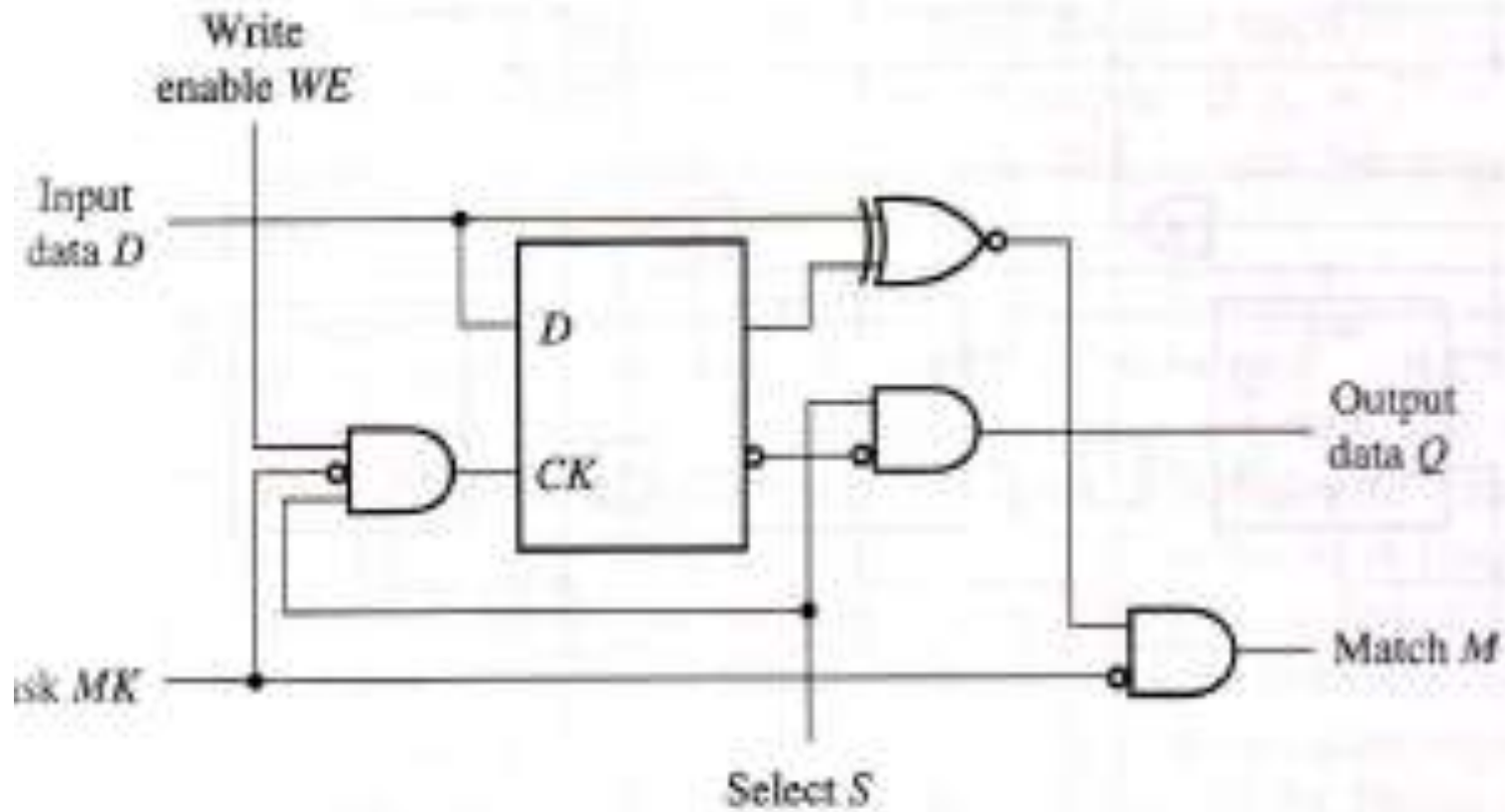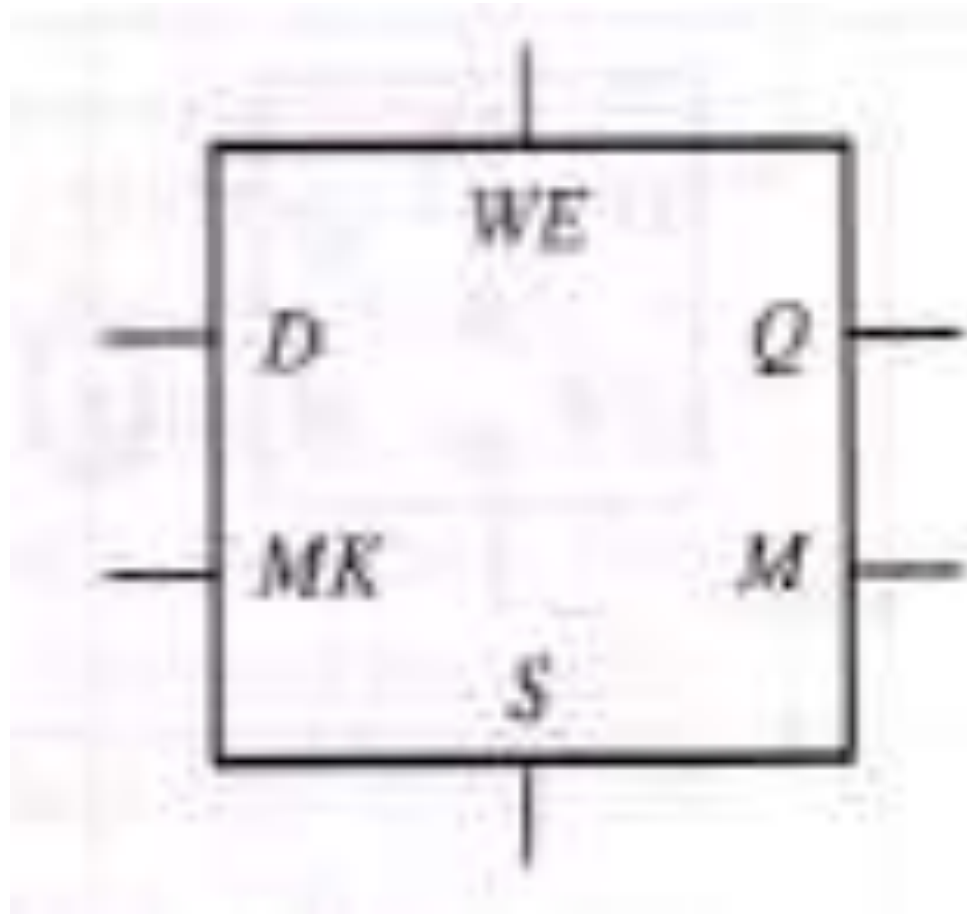# ASSOCIATIVE MEMORY WITH M WORDS AND N BITS/WORD:

EACH CELL CONSIST OF

MATCH LOGIC:

CELL MADE BY MAKING USE OF D FLIP-FLOP:

THE ABOVE DIAG CAN BE SYMBOLISED AS:

4 * 4 BIT ASSOCIATIVE MEMORY ARRAY
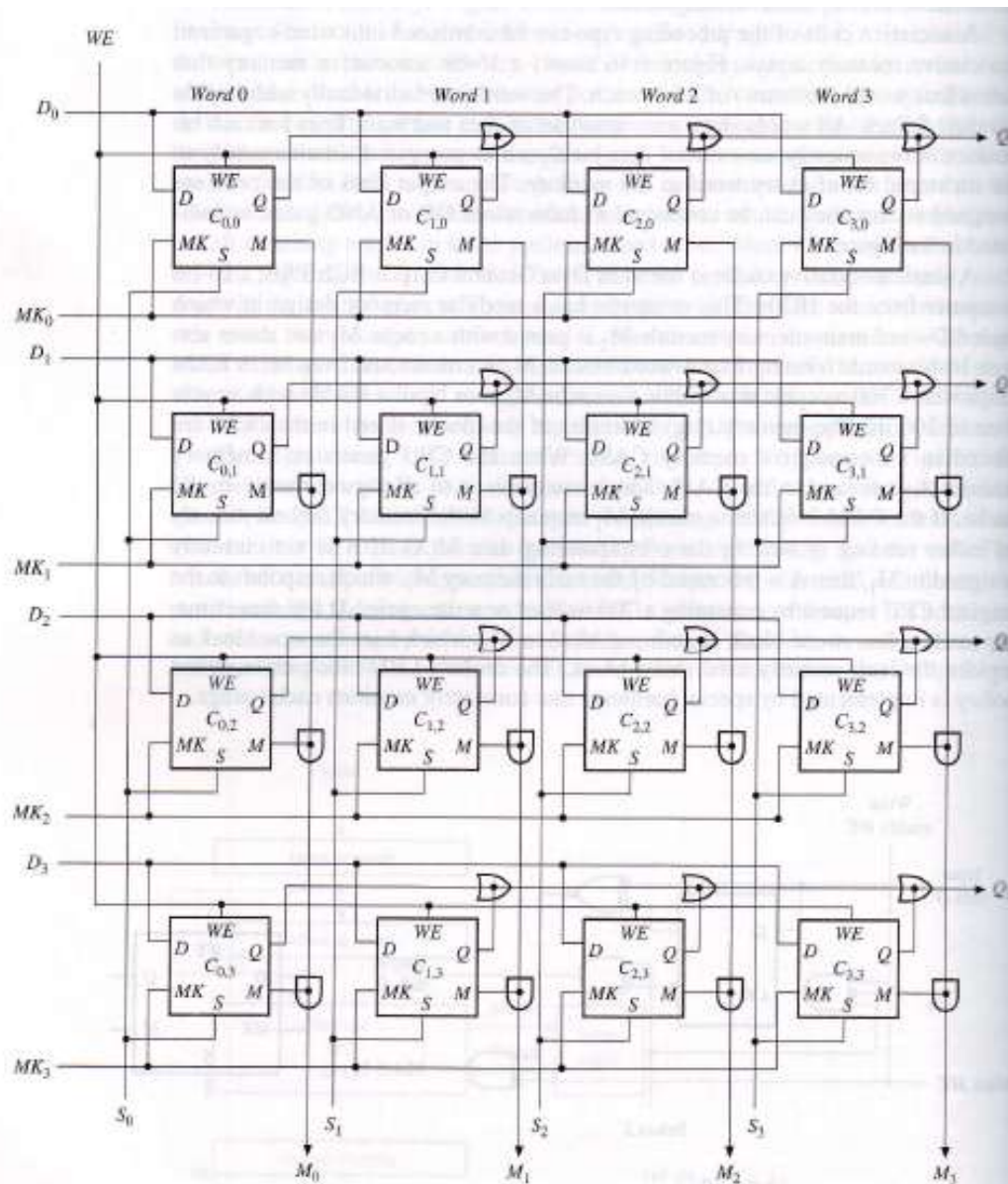
CACHE MEMORY

- The active portions of the program and data are placed in a fast small memory then the average memory access time is reduced, such a fast small memory is known as cache memory.

- When the CPU refers to the memory and finds the word in cache it means there is a hit and if it is not found in cache then it is a miss. The hit ratio can be calculated as:
  - No of Hits/Total No of References(hits+miss)

LOOK-ASIDE CACHE
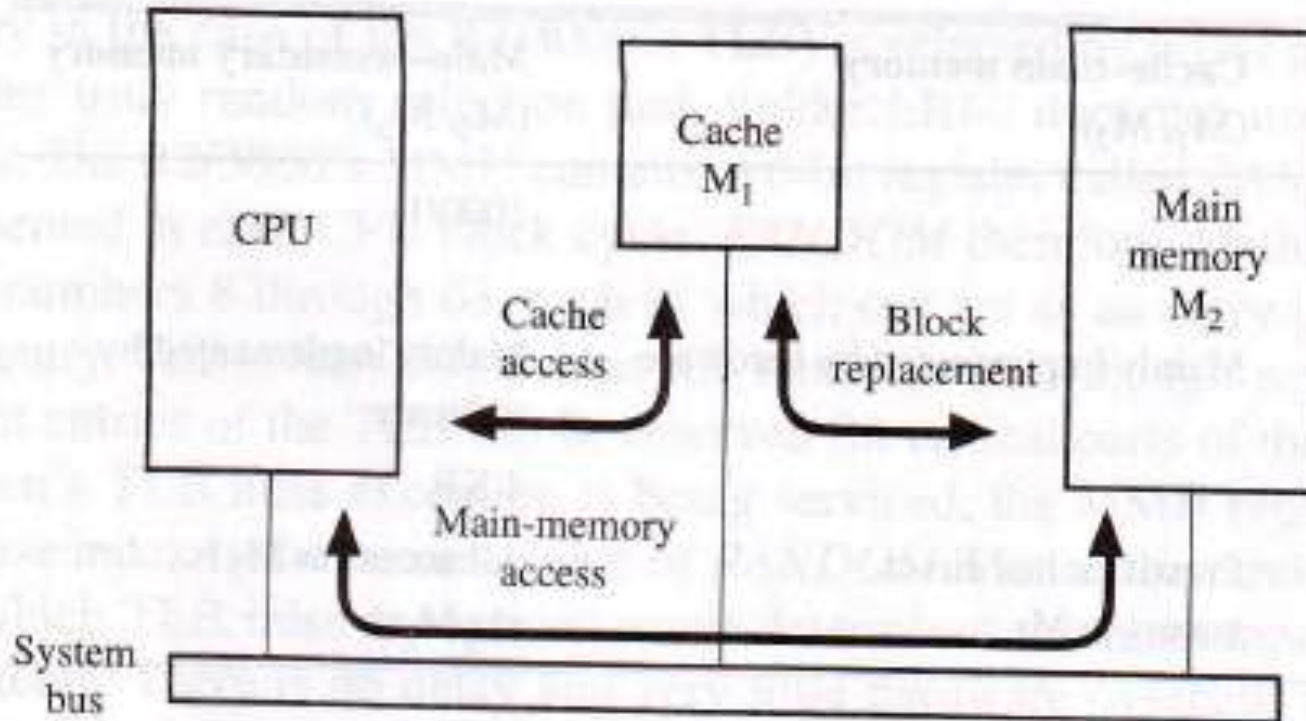
LOOK-THROUGH CACHE

TYPES OF MAPPING

- There are 3 different types of mapping
  - Associative Mapping
  - Direct Mapping
  - Set Associative Mapping

ASSOCIATIVE MAPPING

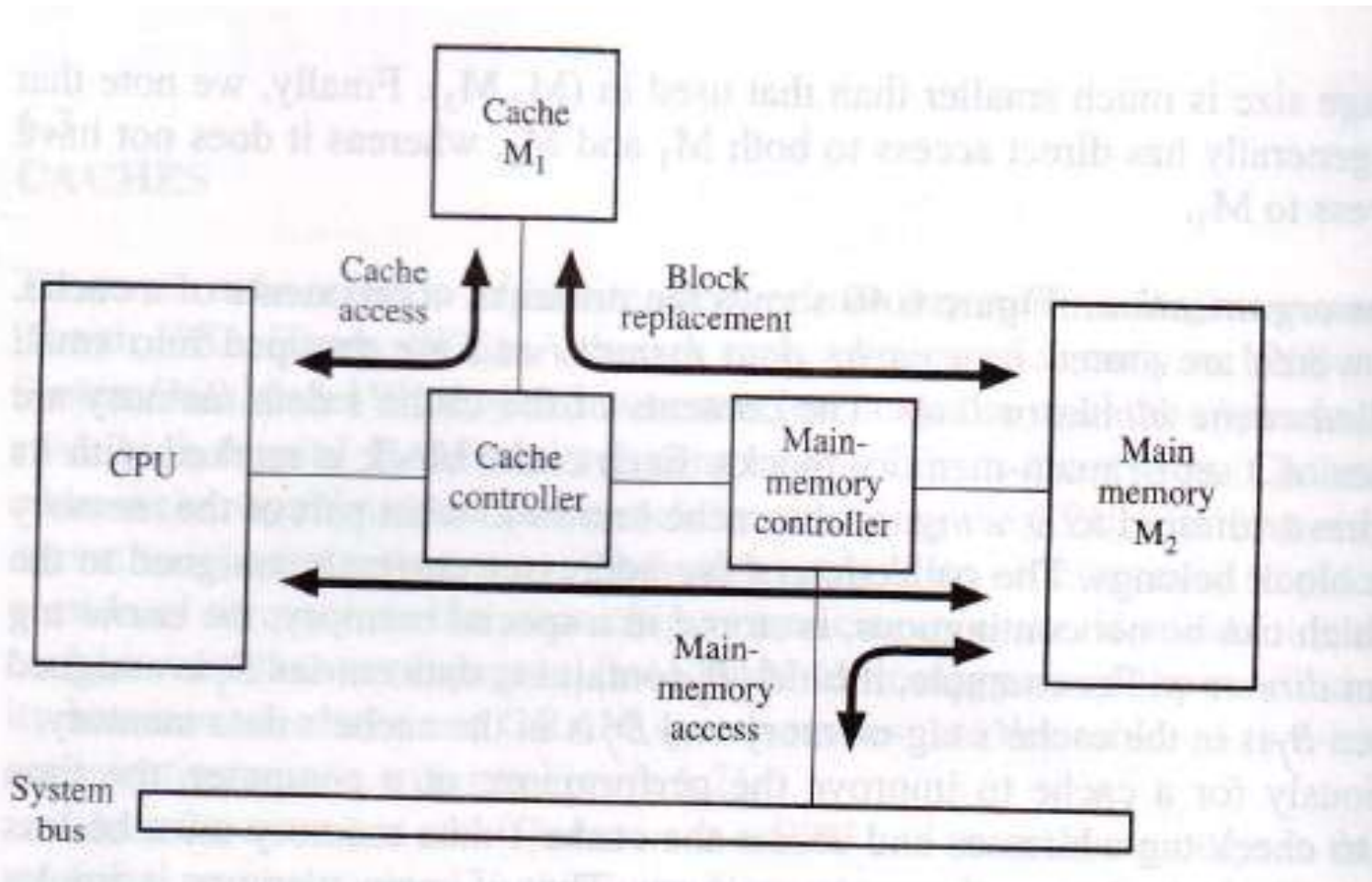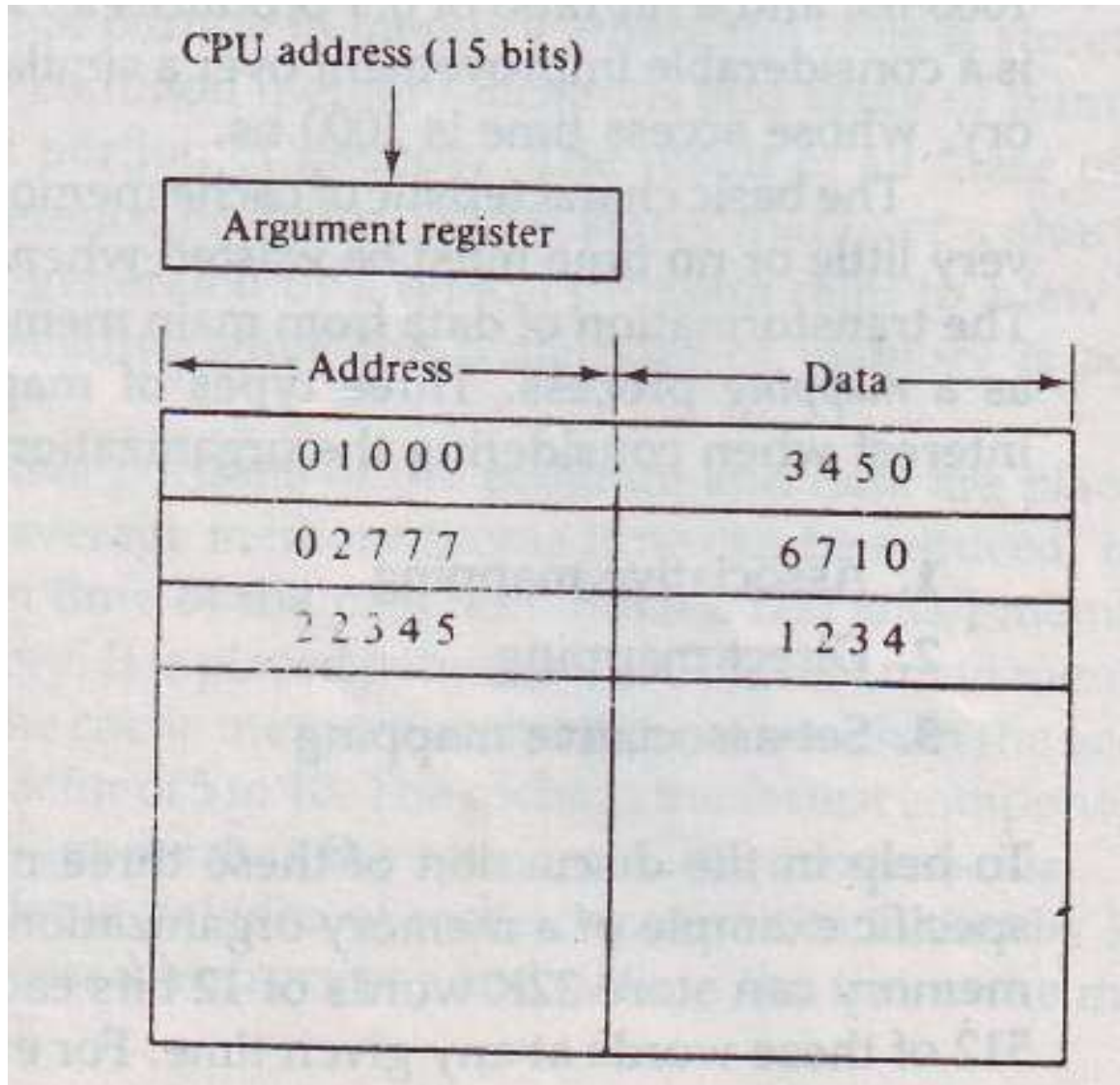CPU address (15 bits)

Argument register

Address	Data
0 1 0 0 0	3 4 5 0
0 2 7 7 7	6 7 1 0
2 2 3 4 5	1 2 3 4

DIRECT MAPPING (1)

DIRECT MAPPING (2)

Memory address	Memory data
00000	1 2 2 0
00777	2 3 4 0
01000	3 4 5 0
01777	4 5 6 0
02000	5 6 7 0
02777	6 7 1 0

(a) Main memory

Index address	Tag	Data
000	0 0	1 2 2 0
777	0 2	6 7 1 0

(b) Cache memory
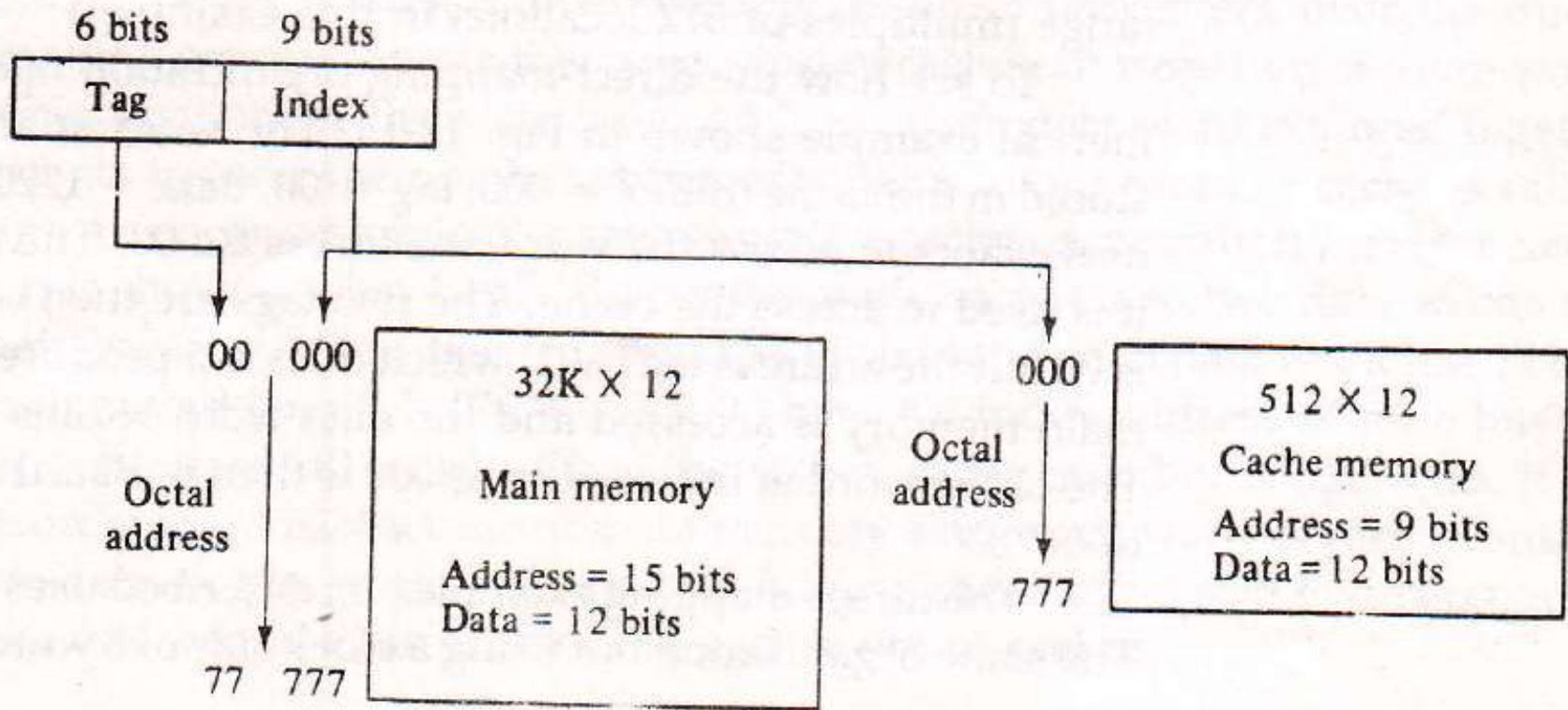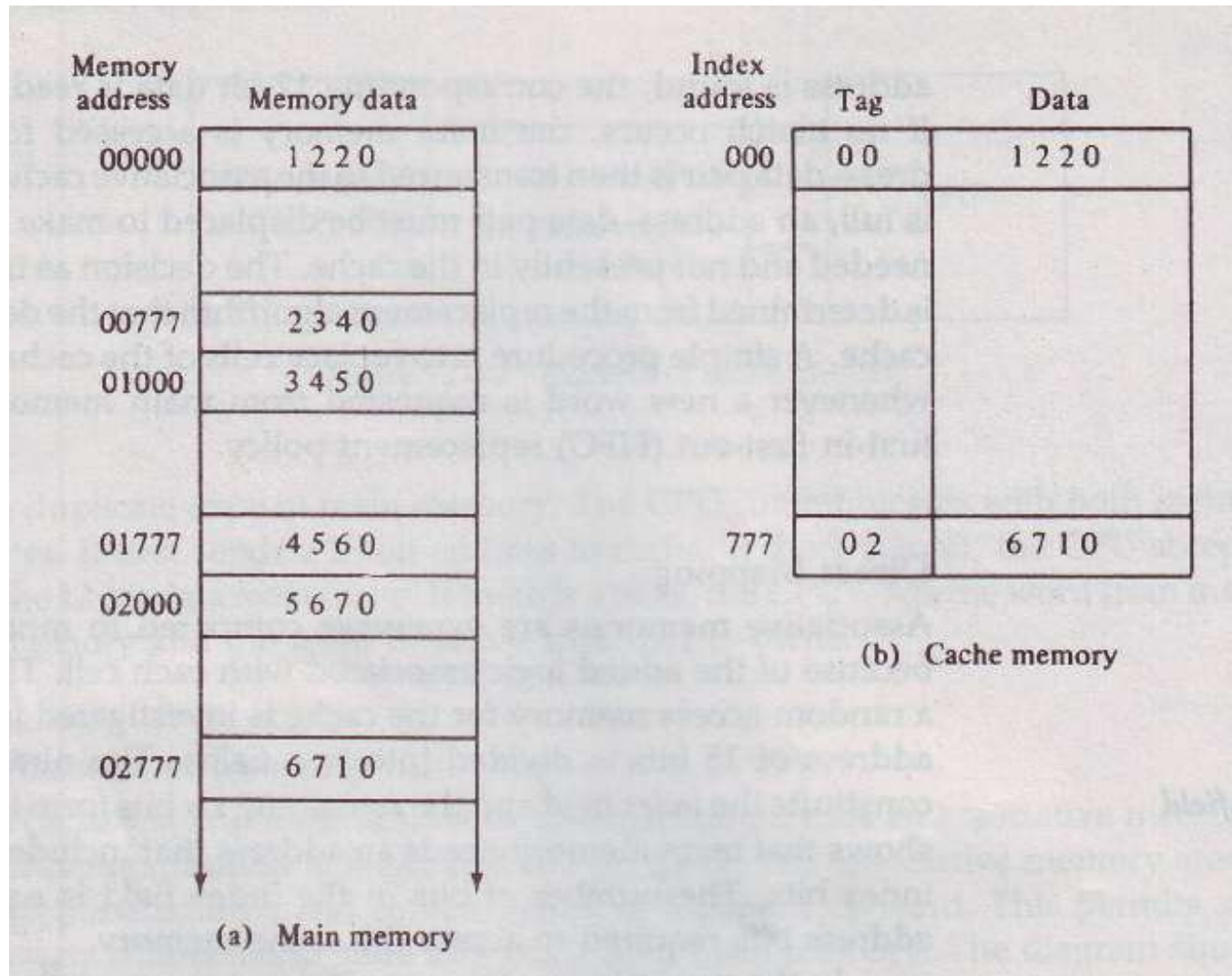
Direct Mapping By Using Blocks

If the direct mapping uses a block size of more than 1k words then each block consist of more than an individual data with different index number but with the same tag

DIRECT MAPPING BY USING BLOCKS (CONTD…)

000	01	3450
007	01	6578
770	02	8891
777	02	6710

Block 0

Block1

Block63

Set Associative Mapping

Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

WRITING INTO CACHE

- Write Through
- Write Back

- Valid Bit-→ Initially it is 0