Lecture-27

Memory System Basics

§ Introduction:—

Memory system is an important part of the computer. Memory system is where the program is stored and retrieved by the CPU for execution. In memory partial and final results are stored after computation.

§ Memory classifications:—

With respect to the way of data access we can classify memories as:

- Random Access Memory (RAM)
- Sequentially accessible Memory (SAM)
- Direct Access Memory (DAM)
- Contents Addressable Memory (CAM)

Random Access Memory :→ In random access memory the access time to any piece of data is independent to the physical location of data. Access time is constant.

(Access time is the time interval between the instant of data read/write request and the instant at which the delivery of data is completed or its storage is started.)

Random Access memory is of two type

1. Read write memories (RAM)
2. Read only memories (ROM)

Read-write memory (RAM) is further devided in three parts

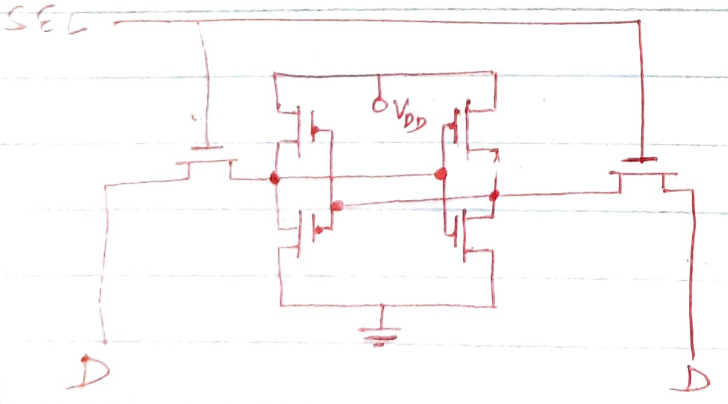
* Static RAM (SRAM)
* Dynamic RAM (DRAM)
* Non-Volatile RAM (NVRAM)

Static Random Access Memory (SRAM) :→

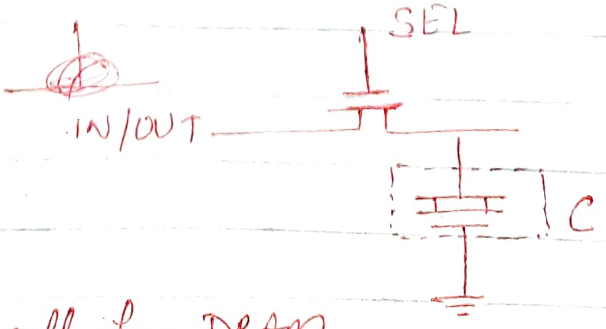
In case of static random access memory (SRAMs), the data is stored esentially in flip-flop circuits.

DRAM :—

Each one-bit memory cell uses a capacitor for data storage. Since capacitors leak there is a need to refresh



Schematics of one-bit cell for static RAM



Memory cell for DRAM

DRAM required to be refreshed every 2 or 4 ms, to keep the data or the charge in the capacitance.

Pros & Cons:—

Static RAM

1. faster and less power hungry than DRAMs.
2. More expensive (6 transistors/cell)

DRAM

- less expensive (only one transistor per cell)
- slower than SRAM

Programmable read only memories (PROM):—

PROM are programmed during manufacturing proces. The contents of each memory cell is locked by a fuse or antifuse (diodes). PROMs are used for permanent data storage.

Erasable read only memories (EPROM):—

Can erase with ultraviolet light (about 20 minutes). Memory cells are built with floating gate transistors. Data can be stored in EPROMs for about 10 years.

Electrically erasable read only memories (EEPROM)

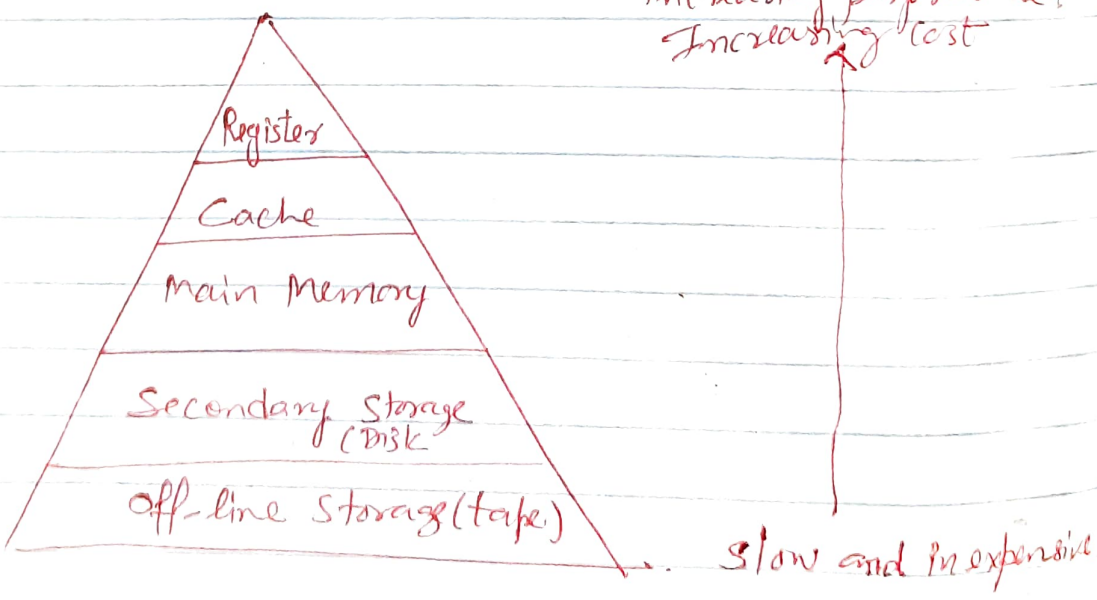
Erasing does not require ultraviolet light but higher voltage and can be applied not to the whole circuit but to each memory cell separately.

Lecture-28

Non-Volatile access memory (NVRAM) :- With this name we refer generally to any memory which does not lose information when power is turned off. Except from ROMs, such as BIOS memory (Basic Input Output System).

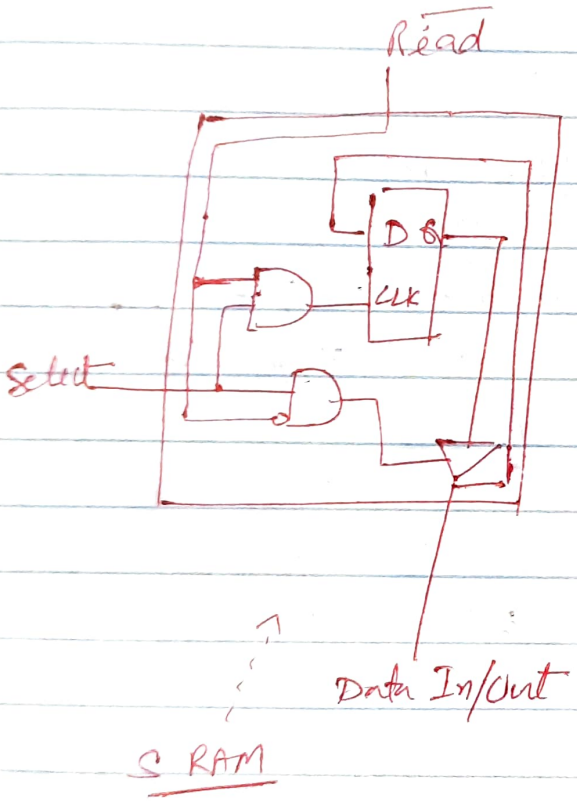
Flash Memory (FLASH) - By this name the cheaper variant of EEPROM is described. In case of FLASH memory not separate bytes but blocks of bytes are being erased at the same time. It makes the construction of such memories cheaper in comparison to regular EEPROMs.

Memory Hierarchy :→

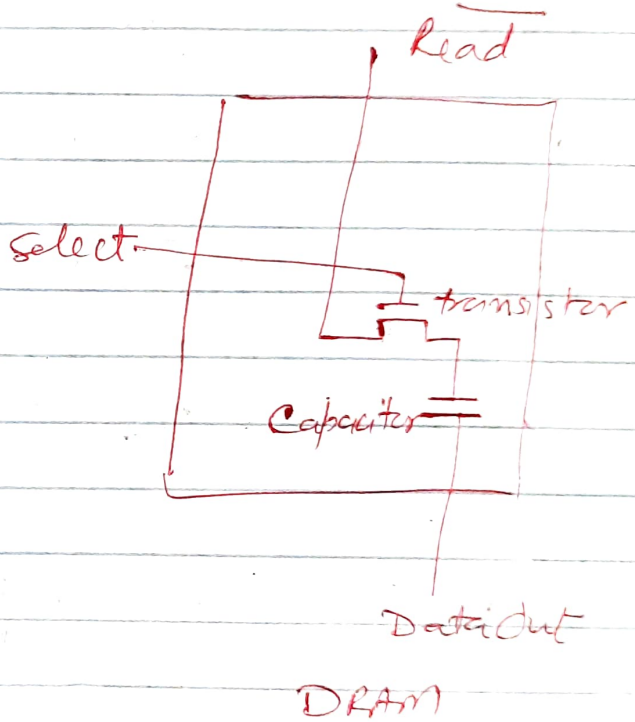


Memory type	Access time	Cost/MB	Typical amount used	Typical cost
Registers	0.5 ns	High	2KB	—
Cache	5-20 ns	$80	2MB	$160
main memory	40-80 ns	$0.40	512MB	$205
Disk memory	5 ms	$0.005	40GB	$200

year 2007

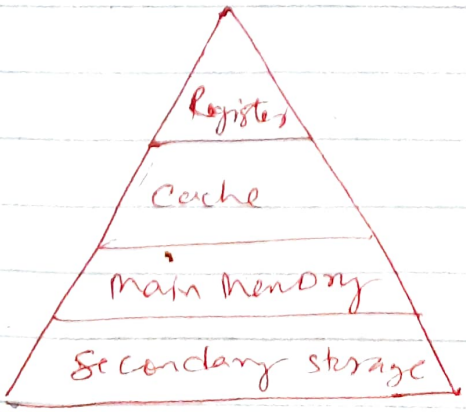


SRAM



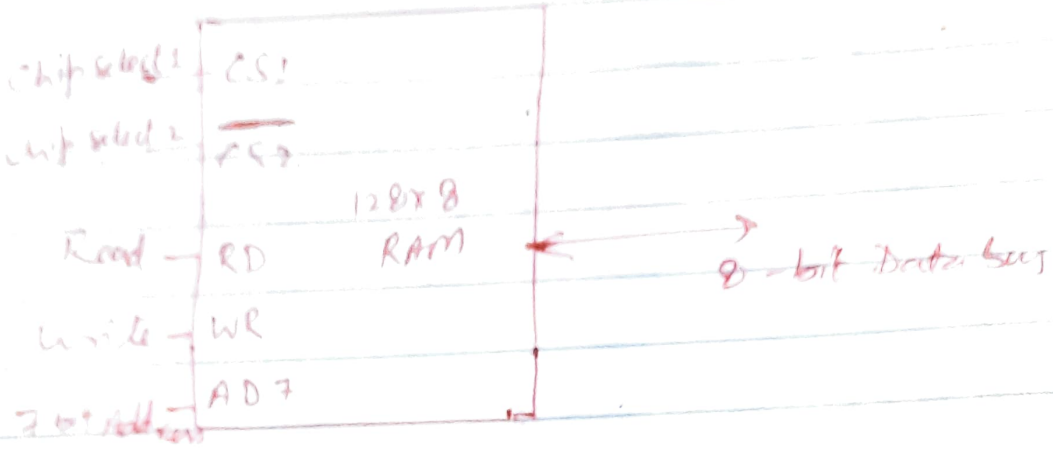
DRAM

Optimization of Memory Hierarchy :→



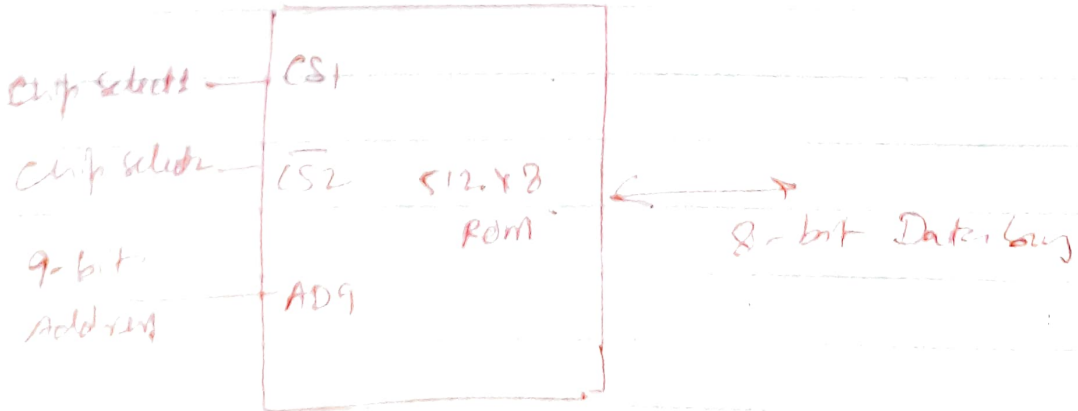
Lecture – 29

RAM Chip



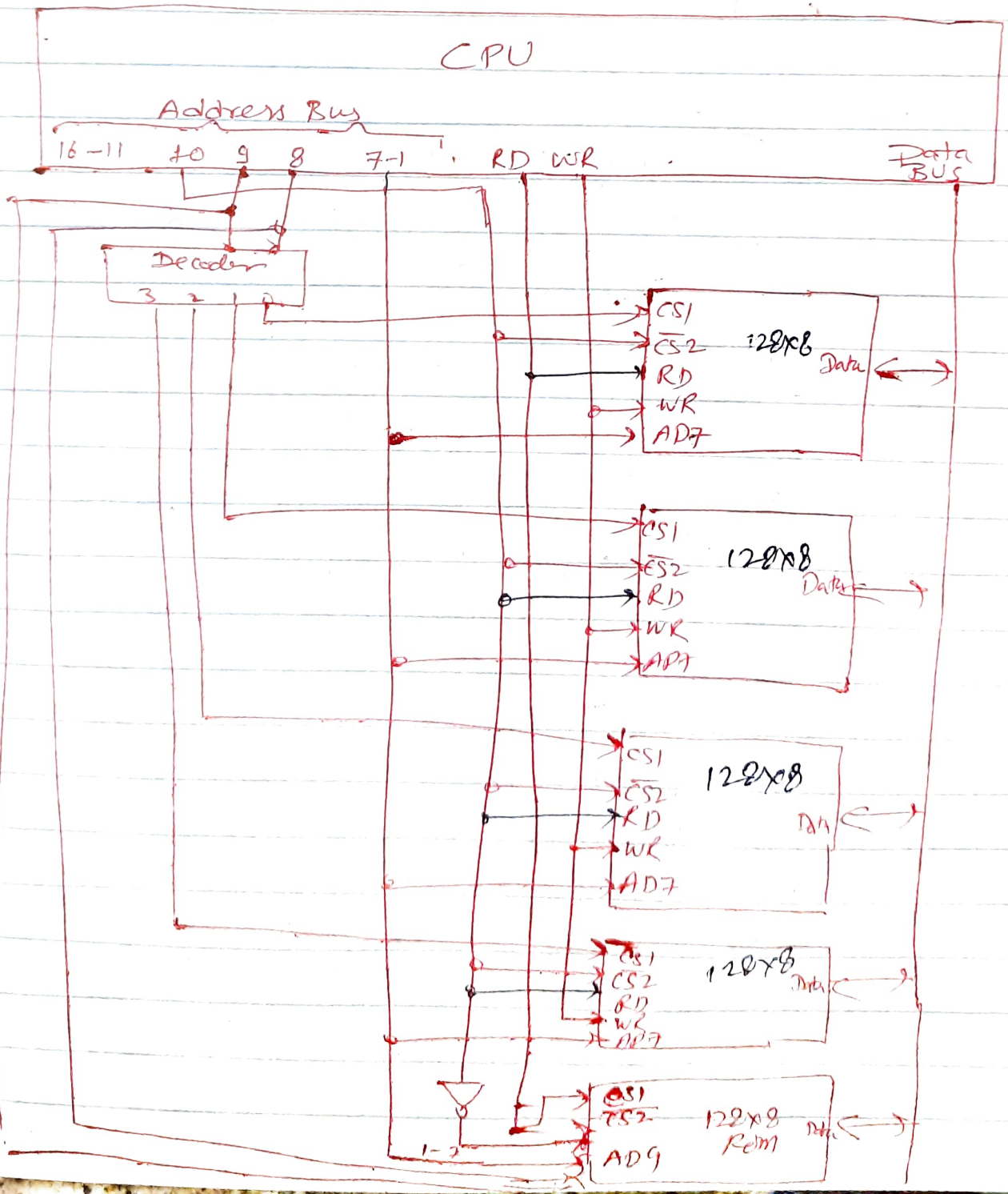
CS1	CS2	RD	WR	Memory function	State of Data bus
0	0	x	x	Inhibit	High-impedance
0	1	x	x	Inhibit	High-impedance
1	0	0	0	"	"
1	0	0	1	write	Input data to RAM
1	0	1	x	Read	Output data from RAM
1	1	x	x	Inhibit	High-impedance

ROM Chip



Memory Address Map for Microprocomputer

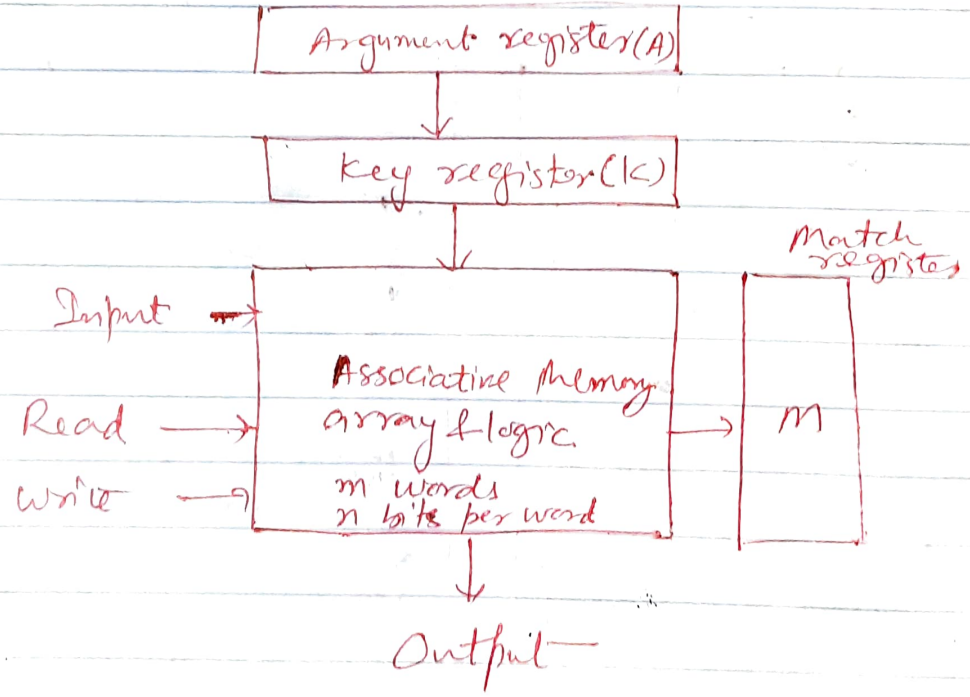
Component	Hexadecimal address	Address Bus 10 9 8 7 6 5 4 3 2 1
RAM 1	0000 - 007f	0 0 0 x x x x x x x
RAM 2	0080 - 00FF	0 0 1 x x x x x x x
RAM 3	0100 - 017F	0 1 0 x x x x x x x
RAM 4	0180 - 01FF	0 1 1 x x x x x x x
ROM	0200 - 03FF	1 x x x x x x x x x



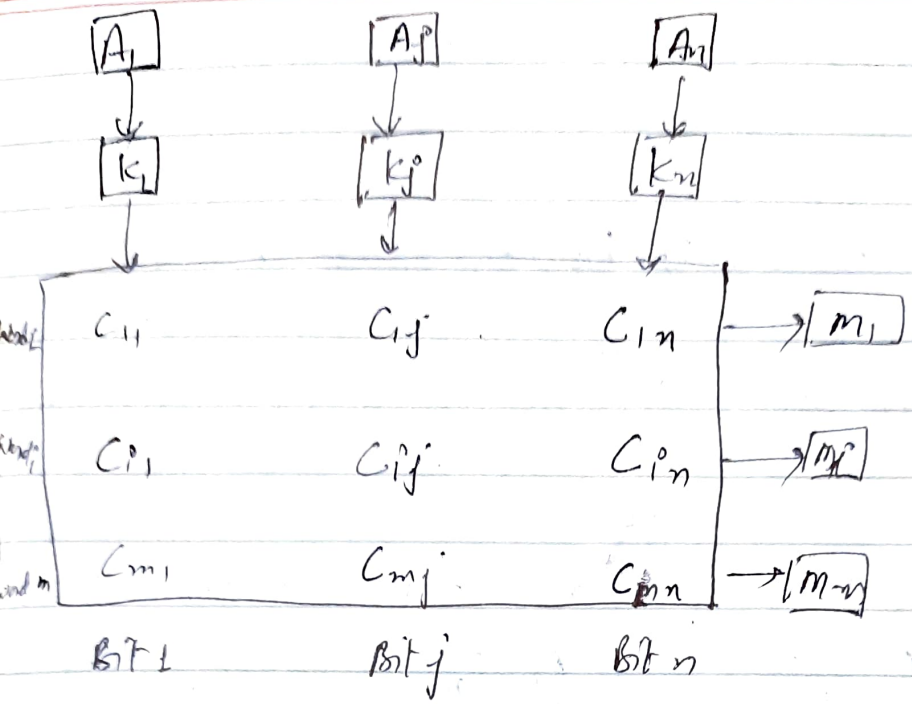
Lecture-30

Associative Memory :→

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit accessed by content is called an associative memory or content addressable memory (CAM)



A	101 111100	
K	111 000000	
Word 1	100 11110 0	no match
Word 2	101 00000 1	match

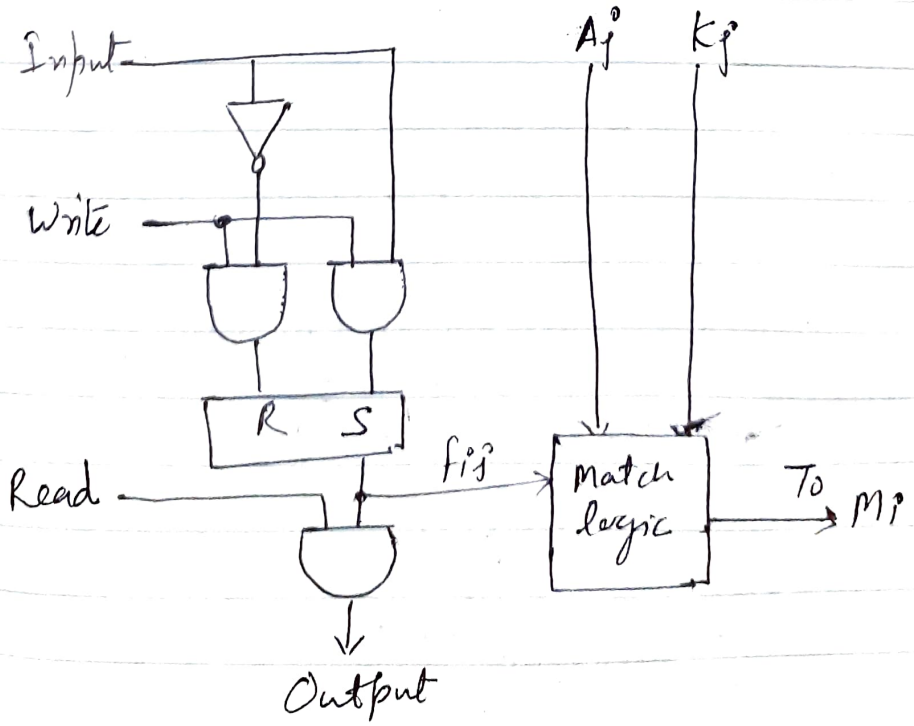


Match Logic

$$x_j = A_j f_{ij} + A_j' f_{ij}'$$

$$M_i = x_1 x_2 x_3 \cdots x_n$$

One cell of associative memory

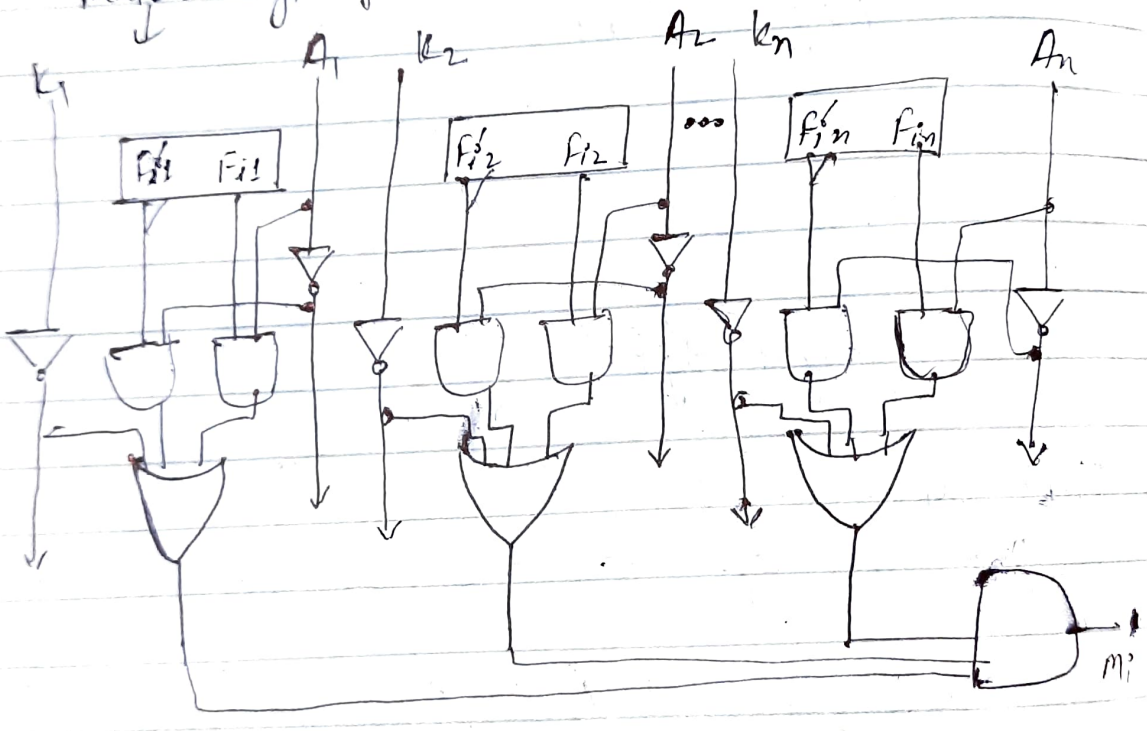


$$x_j + k_j' = \begin{cases} x_j & \text{if } k_j = 1 \\ 1 & \text{if } k_j = 0 \end{cases}$$

$$M_i = (x_1 + k_1')(x_2 + k_2')(x_3 + k_3') \cdots (x_n + k_n')$$

$$m_i = \prod_{j=1}^{n} (A_j F_{ij} + A_j' F_{ij}' + k_j')$$

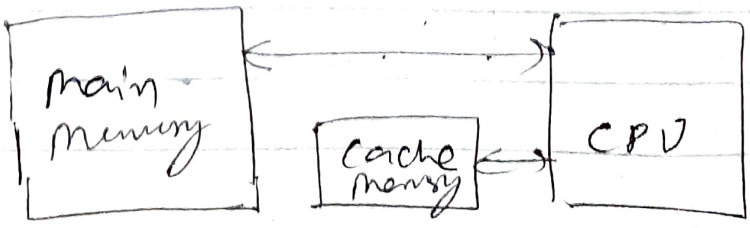
match logic for one word of associative memory



§ Cache Memory :-

Locality of Refrence :- Program access a relatively small portion of the address space at any instant of time.

Hit Ratios and Effective Access Times



Temporal Locality (Locality in Time):- If an item is referenced, it will tend to be referenced again soon.

Spacial Locality (Locality in Space):- If an item is referenced, items whose addresses are close by tend to be referenced soon.

$$\text{Hit ratio} = \frac{\text{No. of times referenced words are in cache}}{\text{Total number of memory accesses}}$$

$$\text{Eff. access time} = \frac{(\#\text{hits})(\text{Time per hit}) + (\#\text{misses})(\text{Time per miss})}{\text{Total number of memory accesses}}$$

Hit Ratios and effective access time for multi-level cache:→

$$H_1 = \frac{\text{No. times accessed word is in on-chip cache}}{\text{Total number of memory accesses}}$$

$$H_2 = \frac{\text{No. times accessed word is in off-chip cache}}{\text{No. times accessed word is not in on-chip cache}}$$

$$T_{EFF} = (\text{No. on-chip cache hits})(\text{One-chip cache hit time}) + (\text{No. off-chip cache hits})(\text{off-chip cache hit time}) + (\text{No. off-chip cache misses})(\text{off-chip cache miss time}) \Big/ \text{Total number of memory accesses}$$

Cache Mapping:→

① Associative mapping:→

In case of associative mapping Address & data of main memory are in the Cache Memory.

CPU Address (15 bits)

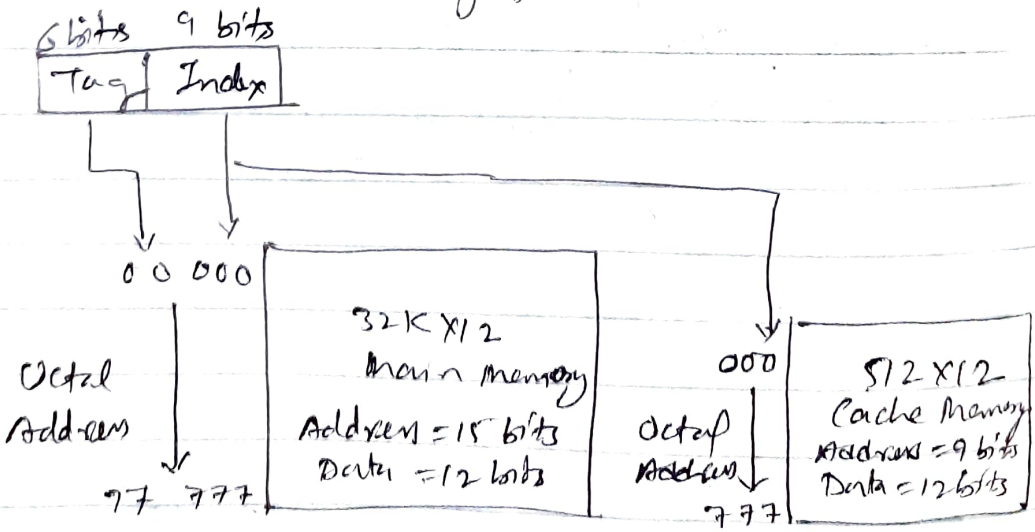
↓

Argument register

← Address →	← Data →
01000	3450
02777	6710
22345	1234

Direct Mapping

In the general case, there are 2^k words in cache memory and 2^n words in main memory. The n-bit memory address is divided into two fields; k bits for the index field and n−k bits for the tag field.



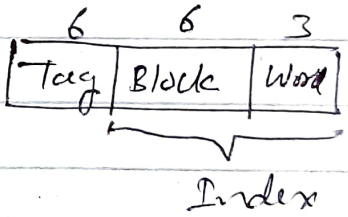
Memory address	Memory data
00000	1220
00777	2340
01000	3450
01777	4580
02000	5670
02777	6710

(a) Main Memory

Index address	Tag	Data
000	00	1220
777	02	6710

(b) Cache memory

	Index	Tag	Data
Block 0	000	01	3450
	007	01	6578
Block 1	010		
	017		
Block 63	770	02	
	777	02	6710



Direct mapping cache with block size of 8 words

Lecture-31

Set-Associative Mapping

In general, a set associative cache of set size k will accommodate k words of main memory in each word of cache.

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

Two-way set associative mapping cache.

Write into Cache

(1) Write through:- parallelly update cache & main memory

(2) Write back → Only the cache location is updated during write back operation. The location is then marked by a flag so that later when the word is removed from the cache it is copied into main memory.

Cache Initialization :-

Initially cache has some invalid data. We have a valid bit field for it. When the value of valid

bit is 1 then its data is valid. Initially we make all the valid bits zero.

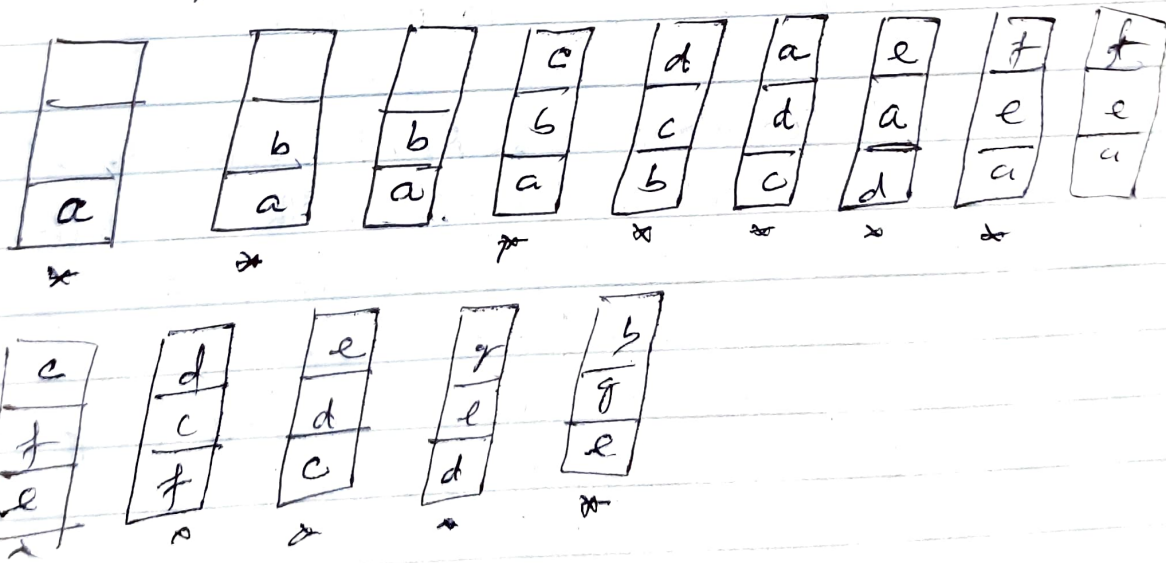
Block Replacement Policies :-
Random Replacement
FIFO (First in first out)
LRU (Leaft Recently Used)

Random Replacement :-
Randomly any block can changed.

FIFO

a, b, a, c, d, a, e, f, a, c, d, e, g, b

block size for 3.

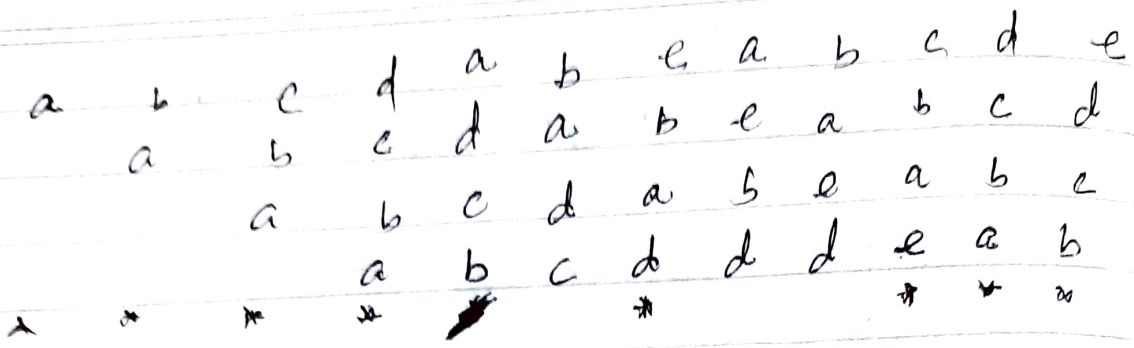


LRU :-

a b c d a b e a b c d e

block size for 4.

Lecture-32



Q1. What do you mean by write-back method? (2012-13) 2 Marks

Q2. Explain the various types of mapping procedures used by cache memory (2012-13) – 10 Marks

Q3. Explain the need of memory hierarchy. What is the main reason for not having a large main memory for storing the totality of information in a computer system (2012-13) 10 Marks.

Q4. A computer employs RAM chips of 256×8 and ROM chips of 1024×8. The computer system needs 2K bytes of RAM, 4K bytes of ROM and 4 interface unit each with 4 registers. A memory mapped configuration is used. The 2-highest order bits of address bus are assigned '00' for RAM '01' for ROM and '10' for interface registers.

(i) How many RAM & ROM chips are needed.
(ii) Draw a memory address map for the system.
(iii) Give the address range in hexadecimal for

teff = effective access time

$t_{eff} = t_{cache} + (1-h)\, t_{main}$

e.g. for a hit ratio of 90% (0.9), a cache access time of 10 ns, and a main memory access time of 60 ns, we have

$t_{eff} = 10 + (1-0.9)\,60 = 16$ ns.

RAM, ROM and Interface.

Write through :→ Parallel updated cache & main memory.

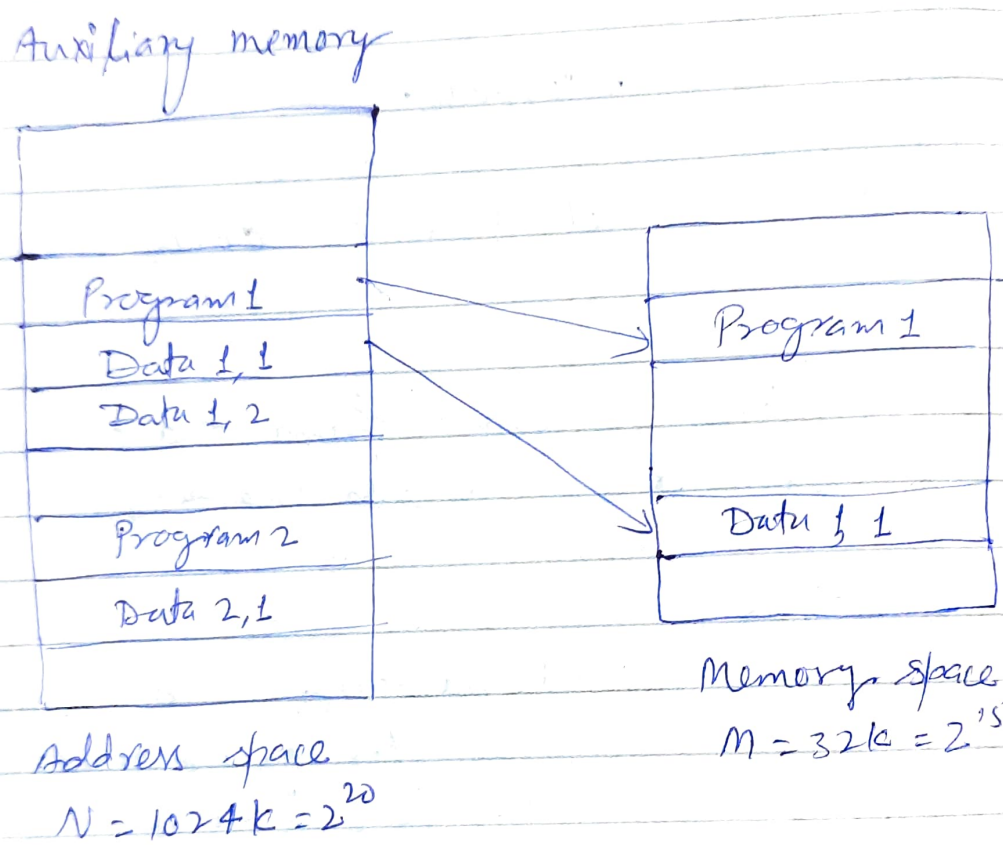
Write - back :— After removing from the cache it may be updated in main memory.

Virtual Memory :— Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large memory space were available, equal to the totality of auxiliary memory.

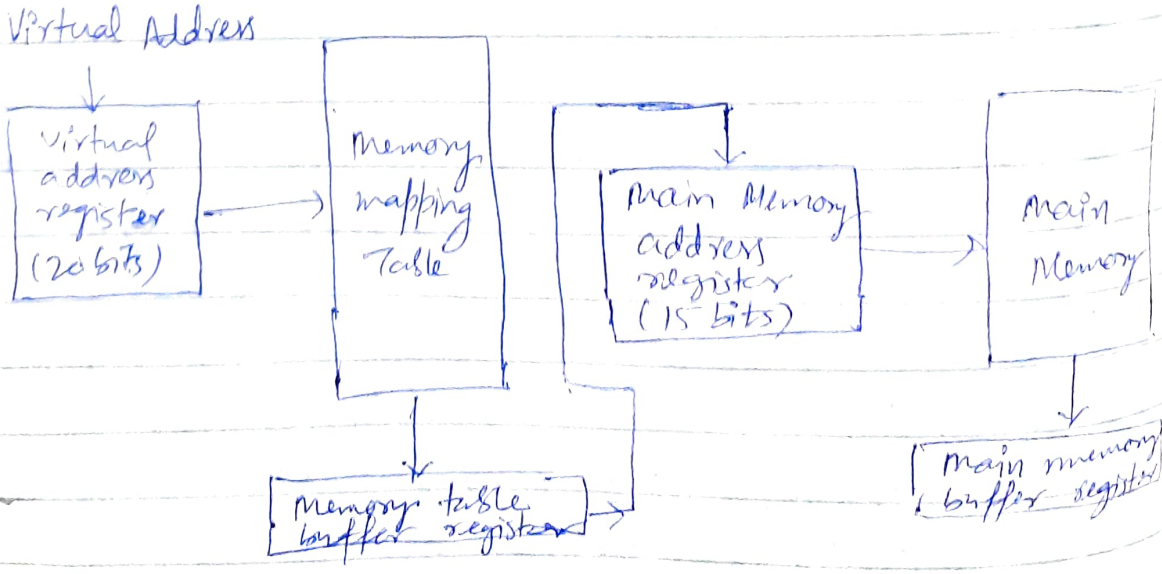
CPU generate virtual address (Address space) & main memory has physical address (memory space)

eg. → main memory capacity of 32K words (1K = 1024). Fifteen bits are needed to specify a physical address in memory since $32K = 2^{15}$. Suppose that the computer has available auxiliary memory for storing $2^{20} = 1024$ words. Thus auxiliary memory has a capacity for storing information equivalent to the capacity of 32 main memories. Denoting

the address space by N and the memory space by M, we then have for this example N = 1024K and M = 32K.



Memory table for mapping a virtual address



For mapping

In first case, an additional memory unit is required as well as one extra memory access time.

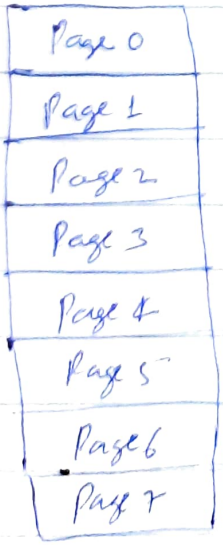
In second case, the table takes space from main memory and two accesses to memory are required with the programs running at half speed.

A third alternative is to use an associative memory. (We will explain)

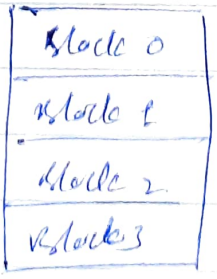
Address Mapping Using Pages :→

Physical memory is broken down into groups of equal size called blocks which may range from 64 to 4096 words each. (page, frame) & virtual memory & (programs) are divided into equal size pages.

eg. Consider a computer with an address space of 8K and a memory space of 4K. If we split each into groups of 1K words we obtain eight pages and four blocks.

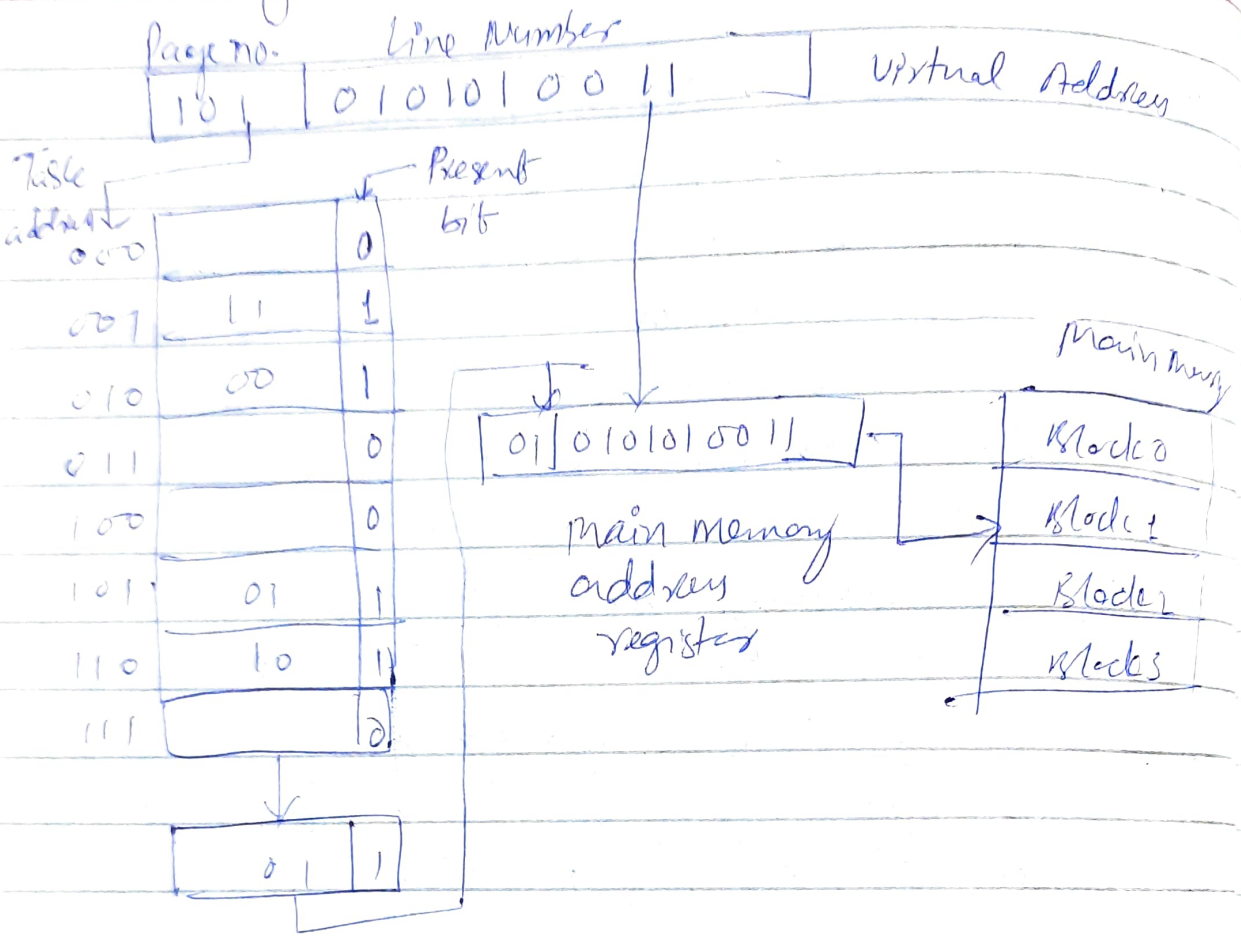


Address space
$N = 8K = 2^{13}$

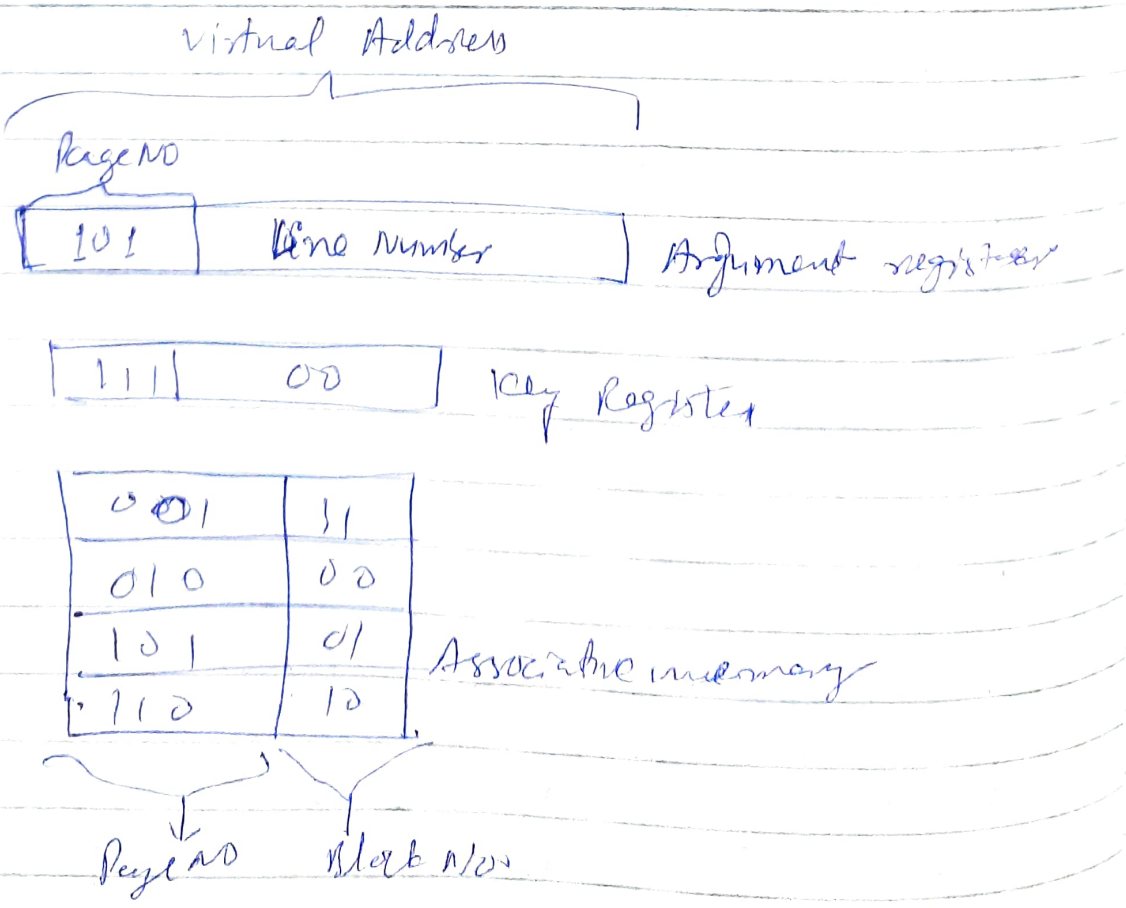


memory space
$M = 4K = 2^{12}$

Memory Mapping in page systems



Associative Memory Page Table



Lecture-33

Page Replacement
FIFO, LRU, Optimal

Memory Management Hardware:-

MMU is combination of software & Hardware.

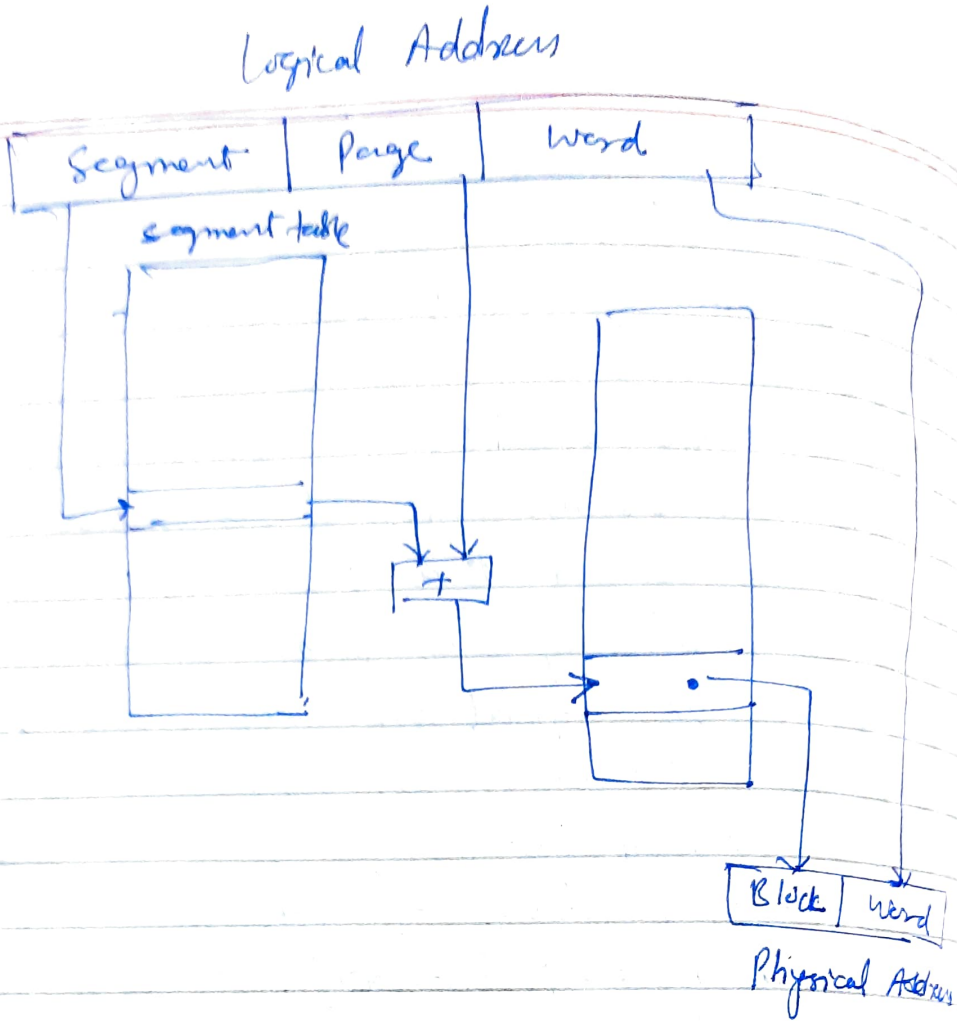
→ The basic components of a memory management unit are:

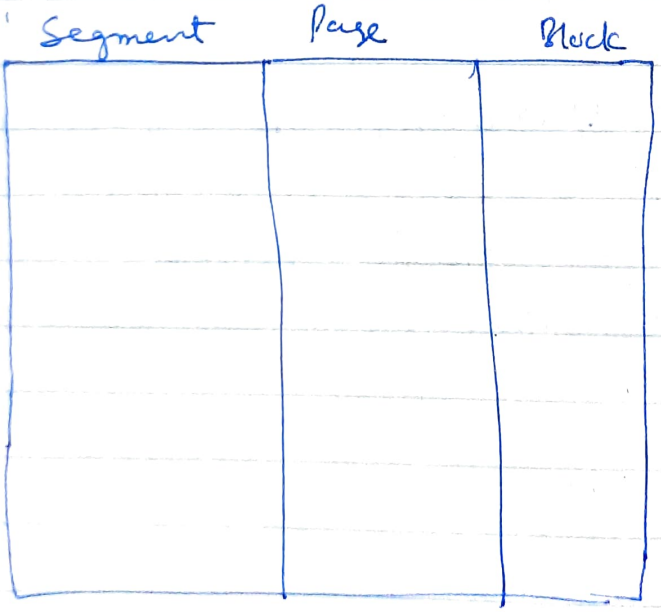
1. A facility for dynamic storage relocation that maps logical memory references into physical memory address.
2. A provision for sharing common programs stored in memory by different users.
3. Protection of information against unauthorized access between users and preventing users from changing operating system functions.

Segment :→
To divide programs and data into logical parts called segments. A segment is a set of logically related instructions or data elements associated with a given name.

Logical Address

Segmented page Mapping :→

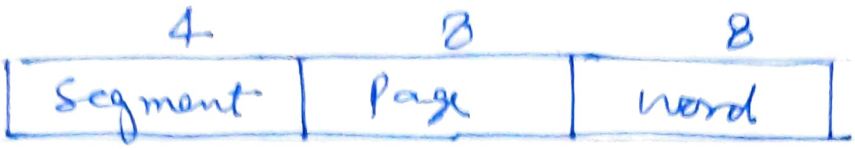




(6) Associative memory translation look-aside buffer (TLB)

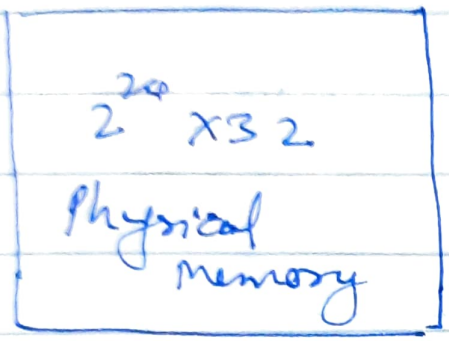
Lecture-34

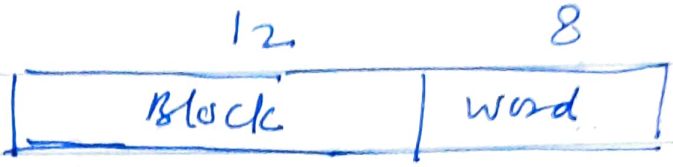
Numerical Example :→



⇒ 20 bits

(a) Logical address format : 16 segments of 256 pages each, each page has 256 words





(b) Physical address format: 4096 blocks ~~of~~ of ~~physical~~ 256 words ~~each~~, each word has 32 bits.

(c) Logical address Assignment

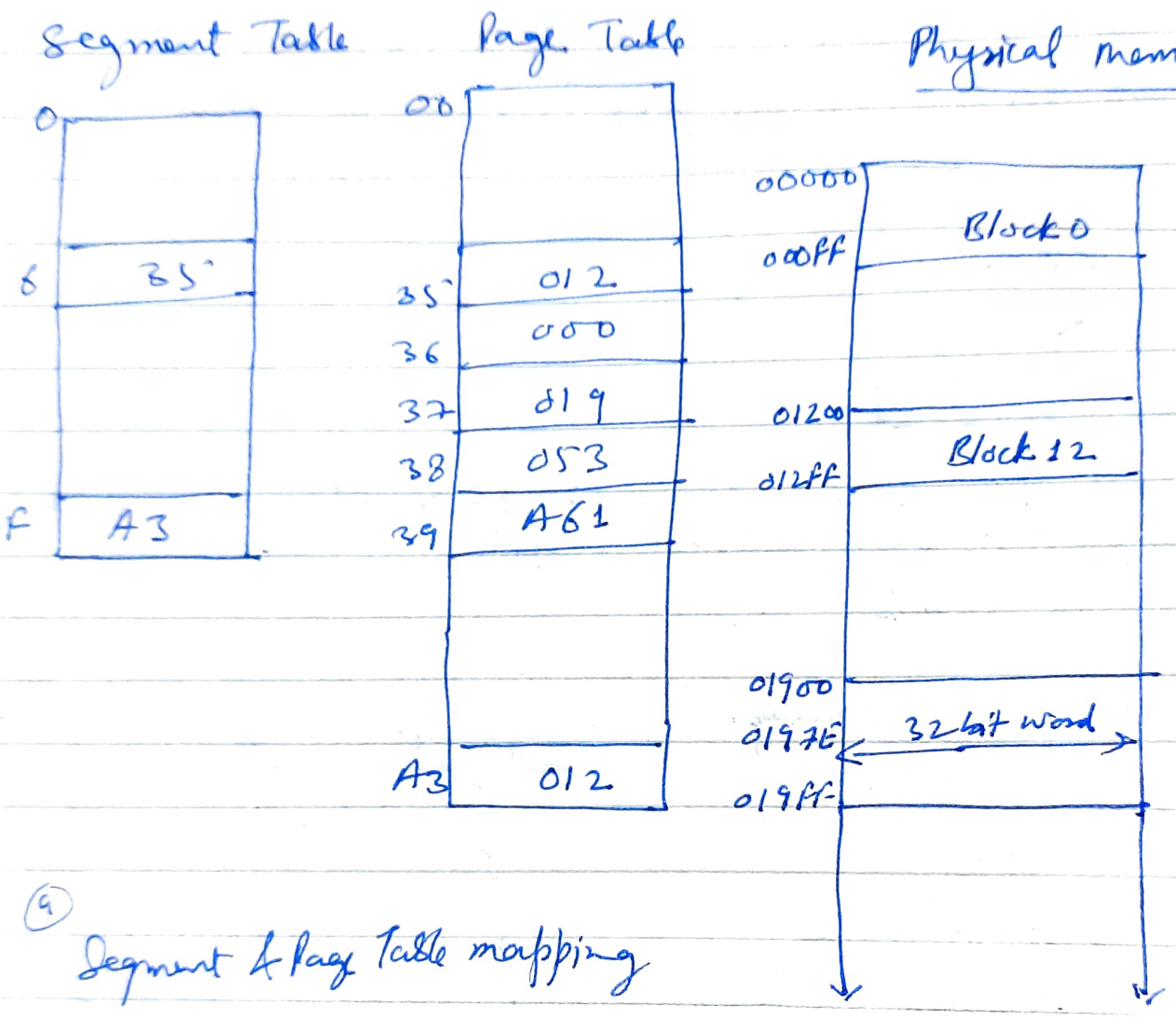
Hexadecimal Address	Page Numbers.
60000	Page 0
60100	Page 1
60200	Page 2
60300	Page 3
60400	Page 4
:	:
604ff	~~Page 5~~

(d) Segment - page versus memory block assignment

Segment	Page	Block
6	00	012
6	01	000
6	02	019
6	03	053
6	04	A61

Logical Address (in hexadecimal)

6	02	76



(a) Segment & Page Table mapping

Segment	Page	Block
6	02	019
6	04	A61

(b) Associative memory (TLB)

Q.1. (a) How many 128x8 RAM chips are needed to provide a memory capacity of 2048 bytes?

(b) How many lines of the address bus must be used to access 2048 bytes of memory? How many of these lines will be common to all chips

(c) How many lines must be decoded for chip select? Specify the size of the decoder.