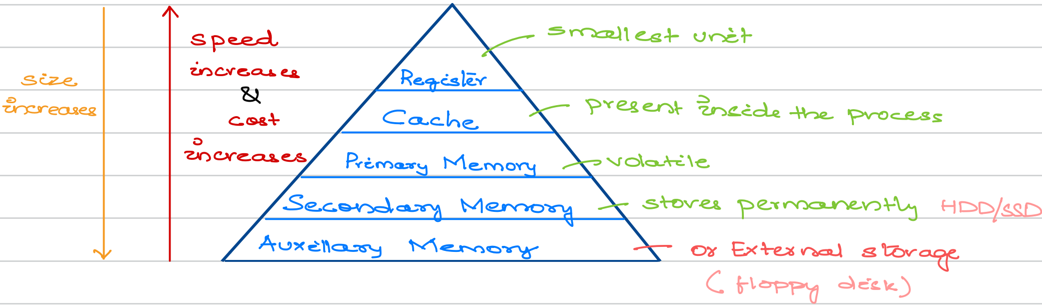
# Memory

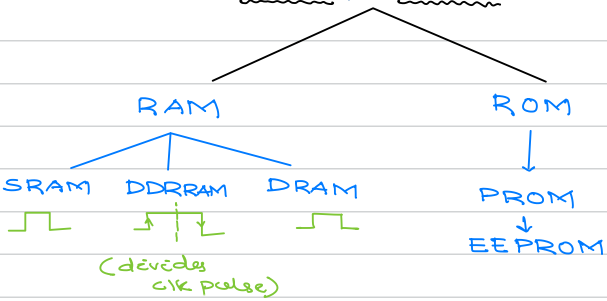
## Memory Hirarchy

represented by Δ structure


size increases
speed increases & cost increases
Register
Cache
Primery Memory
Secondary Memory
Auxellary Memory
smallest unit
present inside the process
volatile
stores permanently HDD/SSD
or External storage (floppy disk)

* secondary memory is a magnetic memory.

## PRIMARY MEMORY



* both are volatile

* both are electronic

* only read → ROM

* Bios is a program which checks all the hard disk & load the OS in the RAM. (Bios is present in ROM).

PROM - Programmable Read Only Memory

EEPROM - Electrically Erasable Programmable Read Only Memory

DRAM - Dynamic RAM

DDRRAM - Dual Data Rate RAM

SRAM - Static RAM

## SRAM

* Static Random Access Memory

Features :-

* faster than DRAM
* used in cache memory
* more expensive
* doesn't need periodic refreshing

→ SRAM uses flipflops (bistable latches) to store each bit of data.

Working Principles :-

1. Write Operation

* data is written by activating the word line (WL).

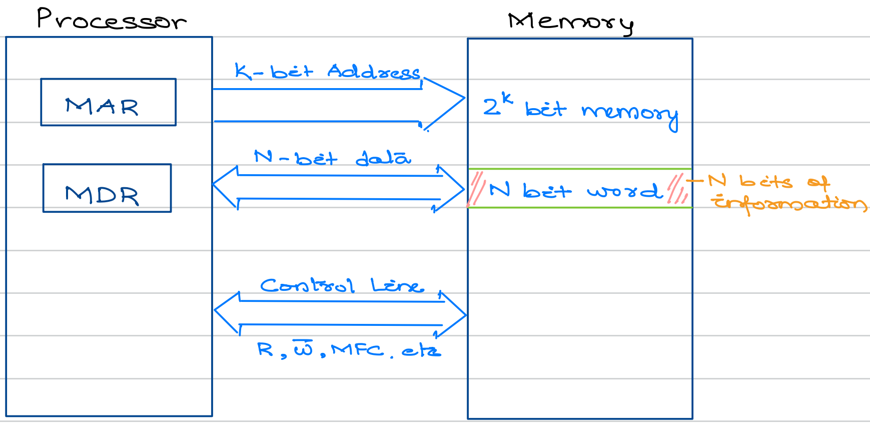
2. Read Operation

* the WL (word line) is activated connecting the stored value to the bit line.

3. Hold / Idle State

* as long as power is supplied, the inverter holds the stored value without refreshing

H.W Same Write for DRAM / not important.



Q- In a computer system MAR holds 28 bits of information/ address & MDR holds 32 bits of data. What is the size of RAM.

Ans> Address size = ($2^k$ bit) = $2^{28}$

Data (k bit) = 32

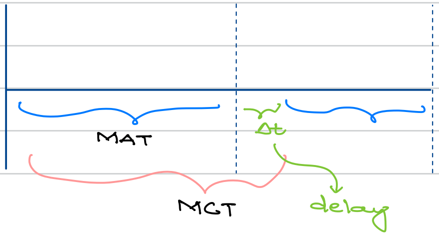
$$2^{28} \times 32 = 2^{20} \times 2^8 \times 32$$
$$= 256 \text{ M.B} \times 32$$

13.02.2025

## Memory Access Time / Memory Latency :-

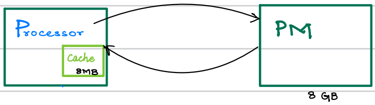
- The time that elapses bet$^n$ initiation of an operation (read or write) & completion of the operation is called Memory Access Time / Latency Time.

i.e the time bet$^n$ read or write control signal sent by the processor & the MFC signal sent by the Memory.



Memory Cycle Time :- The min. time delay req. bet$^n$ 2 successive memory operations.

## CACHE MEMORY



Locality of Reference :-

90% of instruction of the program will execute for the 10% of time & 10% of instruction of program will execute for 90% of time.

Q- How Cache Memory Enhances the Performance :-

* frequently executed instruction will be brought from PM to the cache Memory to satisfy the max time need of the Processor.

Property Based On which Cache Memory Fetches from PM :-

1. Temporal :-

A currently executed instruction is likely to be executed again.

2. Spatial :-

Instruction present next to the current executing instruction is likely to be executed next.

* Information/Instruction always transfer from PM to Cache Memory in the form of a block.

* Both PM & Cache Memory is divided into many no of blocks.

## Terminology

(I) Cache Hit - When the processor request is acknowledged by Cache Memory, it is called Cache Hit.
(processor sends request to Cache Memory)

(II) Cache Miss - When the processor request is not taken/acknowledged by Cache Memory, it is called Cache Miss.

(III) Miss Penalty - The extra time the processor has to spend in case of a Cache Miss is called Miss Penalty.

<IV> Hit Rate - The percentage of Cache Hit over total no of request generated by the processor is called Hit Rate.

<V> Miss Rate - 1 - Hit Rate or $\frac{\text{No of cache miss}}{\text{total no of memory access}}$

$$T_{avj} = \frac{(\text{hit rate} * \text{Cache memory access time}) + \{(1 - \text{hit rate}) * \text{Miss Penalty}\}}{\text{Total no of Memory access (100)}}$$

Average Memory access time

Q- In a comp; the processor takes 200 ns to read a data from cache memory whereas it takes 1000 ns to read the data from PM. Out of 100 memory access. Processor will get the data 80 times from the cache memory, find out the average time the processor will take to fetch a information.

Ans) cache → 200 ns

memory → 1000 ns

$$T_{avg} = \frac{80 * 200 + 20 * (1000 + 200)}{100}$$

$$= \frac{400\cancel{00}}{1\cancel{00}} = 400 \text{ ns} \quad // \text{Ans}$$

Q- In a system the access time of cache memory is 100 ns & main memory is 1000 ns. It is estimated that 80% of memory is for read & 20% request are for write. The hit ratio for read acess only is 0.9. A write through processor is used :-

i) avg access time of the system considering only Memory Read Operation

ii) Avg access time of the system considering both Memory Read & Write Operation

iii) Hit ratio taking into account the write cycle

i) $T_{avg}(\text{read}) = (0.9 * 100) + \{0.1 * (100 + 1000)\}$

$= 90 + 110 = 200$ ns

ii) $$T_{avg}(\text{write} + \text{read}) = \frac{80(0.9 * 100 + 0.1 * 1100) + 20(1000)}{100}$$

$= 360$ ns

iii) Hit rate (read + write) $= 0.8(90) + 0.2(0)$

$= 0.72$

**Mapping** :- bigger address generated by the processor will be converted into smaller address (cache)

* also known as address translation.

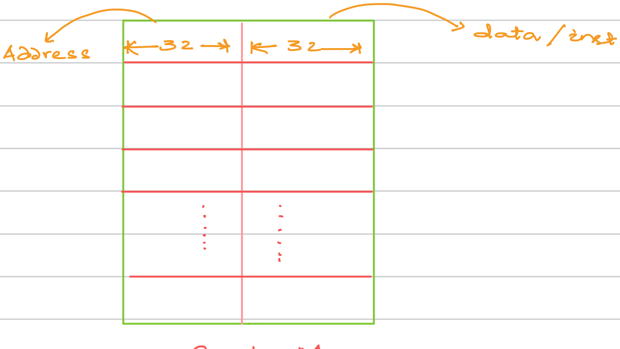
* transferring a block of data from main memory to cache memory is called Mapping.

**Types of Mapping** :-

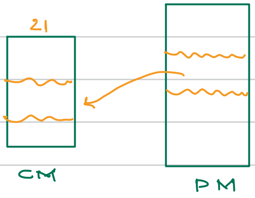
1) Associative Mapping
2) Direct Mapping
3) Set Associative Mapping

**Associative Mapping** :-

* when data or instruction stored in the memory along with the address.



Cache Memory



## Direct Mapping :-

* address generated by the processor is divided into 2 parts :-

i) tag
ii) index

∴ index size is equal to size of cache memory.

∴ main memory address - index = tag

32-bit

| tag | index |
|---|---|
| 11 | 21 |

← 32 →

## Set - Associative Mapping :-

↳ 2 way, 4 way, 8 way, 16 way

* under one tag we can hold multiple index.

" Concept of block "

↳ index is divided into 2 parts

- BLOCK
- WORD

|  | index |  |
|---|---|---|
| tag | block | word |
|  |  |  |

eg:- 12 | 4890

first 48 is searched & then direct 90

" so searching time gets less ".

Q- In a computer system, the size of main memory is 4GB × 32. Size of cache memory is 8 MB. If a block of data contains 512 kb of data. Design the Mapping.

Ans) PM → 4GB

Address Size = 32 bits

| tag | index |
|---|---|
| 9 | 23 |

← 32 →

CM → 8 MB

Address size = 23 bits

tag = 32 - 23 = 9

block size = 512 kb

word address = 19 bits

$$\text{No of blocks} = \frac{2^{23}}{2^{19}} = 2^4 = 16$$

|  | index |  |
|---|---|---|
| tag | B | w |
| 9 | 4 | 19 |

← 32 →

← 23 →

* imp

$$\text{No of tag} = \frac{\text{No of blocks in main memory}}{\text{No of blocks in cache memory}}$$

$$\text{No of set} = \frac{\text{No of blocks in cache memory}}{\text{K way}}$$

→ generalized form

Approach 2

Associative Mapping :-

| tag | word |
|---|---|

Direct Mapping

| tag | block | word |
|---|---|---|

Set- Associative Mapping

| tag | set | word |
|---|---|---|

Q- Consider a cache consists of 128 blocks & MM consists of 4K blocks. Each block having 16 words. How many bits are required for tag block & word field for direct mapping? How many bits are req. for associative mapping? How many bits are req. for tag set word field for 4 way mapping?

Ans> Cache - 128 blocks $\quad 4K = 2^{12} = 4096$

word address = 4 bits

Size of MM = $2^{12} \times 2^{4} = 16$ bits

Size of CM = $2^{7} \times 2^{4} = 11$ bits

Memory Address = $2^{16}$

Associative

| 12 bits | 4 bits |
|---|---|
| tag | word |
| $2^{12}$ | $2^{4}$ |

$\longleftarrow 16 \longrightarrow$

$$tag = \frac{\text{No of block in MM}}{\text{No of block in CM}} = \frac{2^{12}}{2^{7}} = 2^{5} = 5 \text{ bits}$$

Direct

← 16 →

| 5 | 7 | 4 |
|---|---|---|
| tag | block | word |

Set Associative

$$set = \frac{\text{No of blocks in CM}}{\text{K way set ass. map}}$$

$$= \frac{128}{4} = 32 = 2^{5}$$

| 7 | 5 | 4 |
|---|---|---|
| tag | set | word |

← 16 →

Q. A cache consists of a total of of 256 blocks. The MM consists 128 K blocks, each consisting of 32 words. How many bits are there in each of the Tag, Block & word field

Ans Cache = 256 blocks

MM blocks = 128 K = 128 × 1024

block size = 32 words

WORD = $2^{5}$ = 32 = 5 bits

CACHE = 256 blocks = $2^{8}$ = 8 bits

MEMORY BLOCKS = $2^{17}$

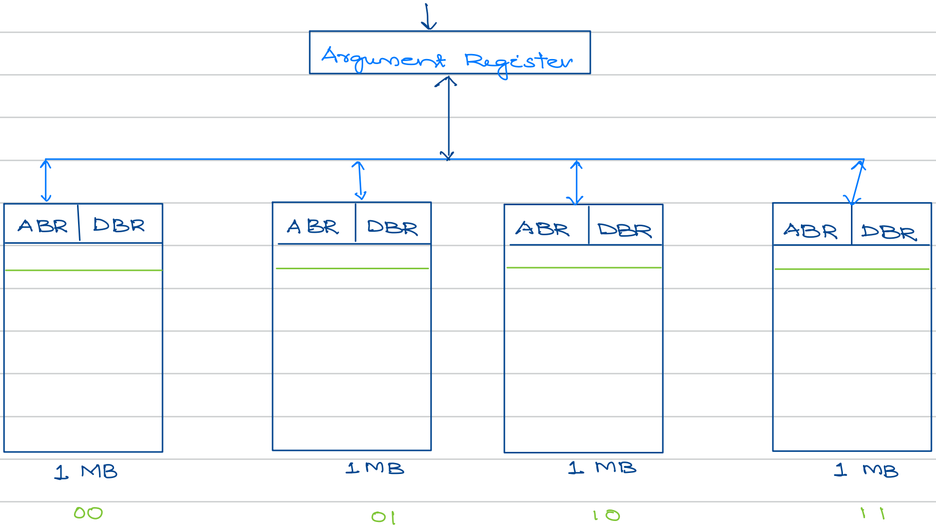
## Memory Interleaving

* Ram optimizing technique

2 types :-

* High Level M.I
* Low Level M.I

} based on how we interpret the address.



ABR → Address Buffer Register

DBR → Data Buffer Register

High Level Interleaving → MSB is considered

↳ no advantage

Low Level Interleaving :- LSB is considered

| 21 | |
|---|---|
| 0 0 0 0 . . . 00 | 00 |
| 0 0 0 0 . . . 00 | 01 |
| 0 0 0 0 . . . 00 | 10 |
| 0 0 0 0 . . . 00 | 11 |
| 0 0 0 0 . . . 01 | 00 |
| 0 0 0 0 . . . 01 | 01 |
| ⋮ | |
| 1 1 1 1 . . . 11 | 11 |

} stored in each location

* Cache Coherence (Memory Write)

* is a memory updation technique.

2 diff approach :-

1) Write Through

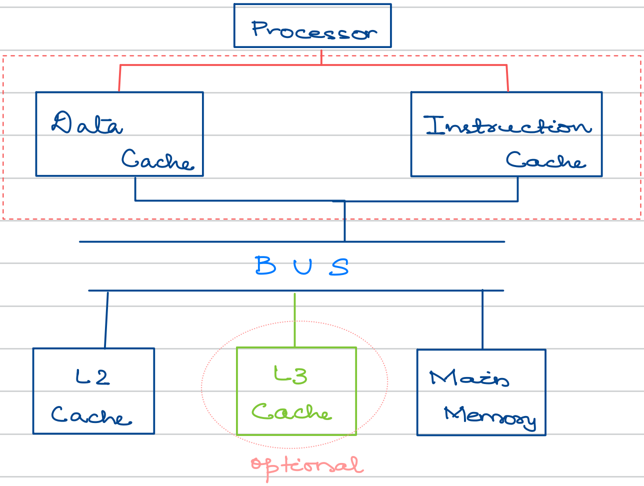
2) Write Back

Write Through - When processor generates a write request to the information will be written in the respective locations of all the memory present in the hierachy simultaneously.

* time consuming

Write Back :- Under write back, the data will be written only inside the cache memory & the locations are marked as 1 (dirty bit). When a particular block of cache memory need to be replaced with a new block of M.M.; First the modified data's are reflected on the higher memory in the hierarchy & the new block is placed in the lower memory.

## Multi Level Cache :-



$$T_{avg} = h_1C_1 + (1-h_1)h_2C_2 + (1-h_1)(1-h_2)M$$

$h_1$ → hit rate of $L_1$ cache

$c_1$ → $L_1$ cache memory access time

$h_2$ → hit rate of $L_2$ cache

$c_2$ → $L_2$ cache memory access time

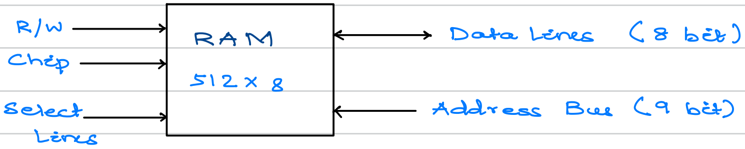
M → Main memory access time

Q→ Hit Rate :- 70%

$T_{avg} = (100 \times 0.7) + (1-0.7)\ 0.9 \times (100+400) +$
$(1-0.7)(1-0.9)\ 2500$
$= 70 + 135 + 75$
$= 280$ ms

excluding L2 cache

$T_{avg} = 100 \times 0.7 + (1-0.7)\ 2100$
$= 70 + 630$
$= 700$ ms



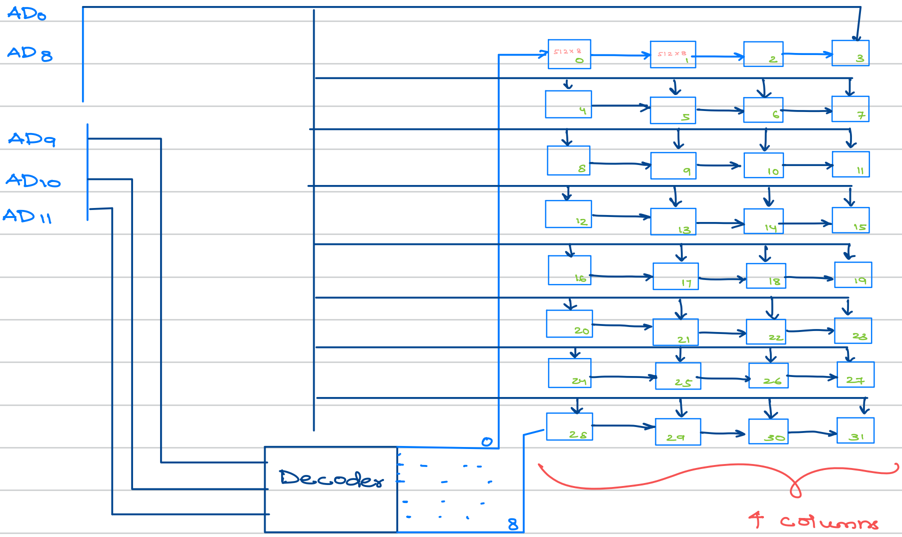
## RAM Analysis

$$\boxed{= \frac{M \times N}{P \times Q}}$$

$$= \frac{\overset{M}{4K} \times \overset{N}{32}}{\underset{P}{512} \times \underset{Q}{8}}$$

$$= \frac{2^{12}}{2^{9}} \times 4$$

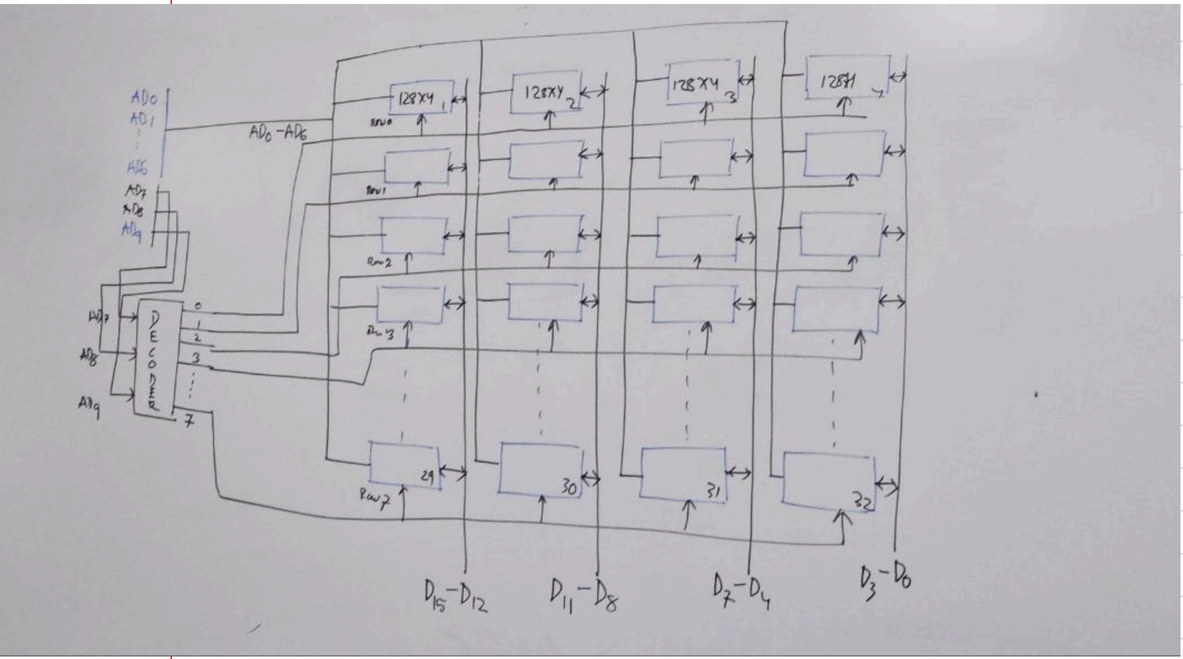
$$= \underset{\text{vertically 8 row}}{8} \times \underset{\text{horizontal 4 columns}}{4}$$



Q- A comp. uses RAM Chip of 128 x 4 capacity. Design a memory of 1K x 16 by using available chip.

Ans> $\dfrac{2^{10} \times 2^{4}}{2^{7} \times 2^{2}} = 2^{3} \times 2^{2} = 8 \times 4$

vertically 8, horizontal 4

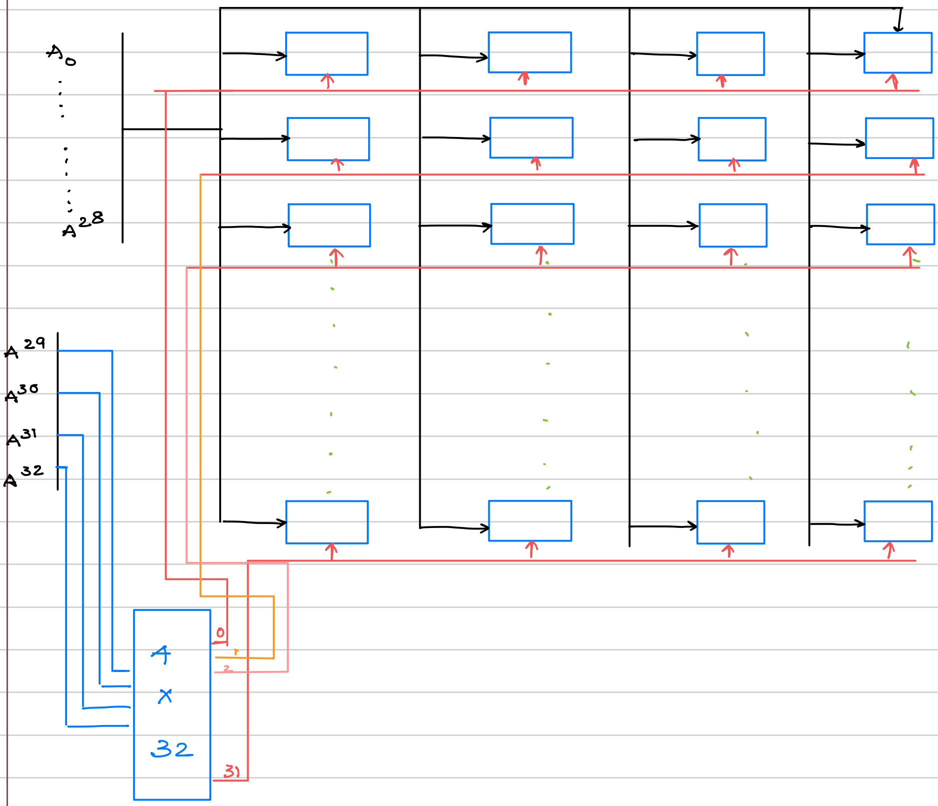


Design a RAM of 8GBX64 using RAM of size 512MBX16

$$= \frac{8 \times 2^{30}}{512 \times 2^{20}}$$

$$= \frac{2^{3} \times 2^{30}}{2^{9} \times 2^{20}}$$

$$= 16 \times 4$$


A0
A28
A29
A30
A31
A32
4
x
32
0
1
2
31