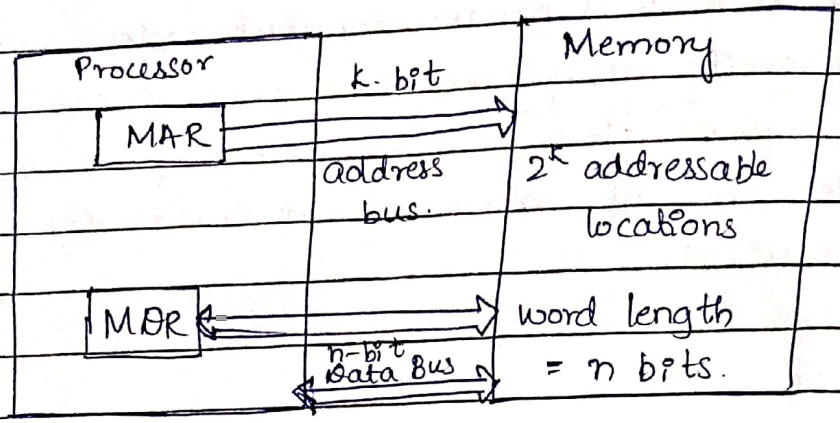
Asha

Date :
Page No :

UNIT-3

MEMORY UNIT

Basic concepts.



Connection between processor & Memory

1. 32-bit = 2^{32} addressable memory locations.
2. Memory Access Time ⇒ Time required to memory operation & its completion.
3. Memory Cycle Time ⇒ Time b/w two successive memory operation.
4. Virtual Memory.

(1), (2), (3) — estimate memory speed

Memory Access Time: Is the time elapsed between operation initiation and its completion.
Ex: Time b/w Read & MFC.

Memory cycle Time:
Ex: Time b/w 2 read operation.
Memory cycle time is the minimum time delay required b/w initiation of two successive memory operation.

Date :
Page No :

Virtual Memory : It translates the address specified by the program into an address that can be used to access the physical memory.

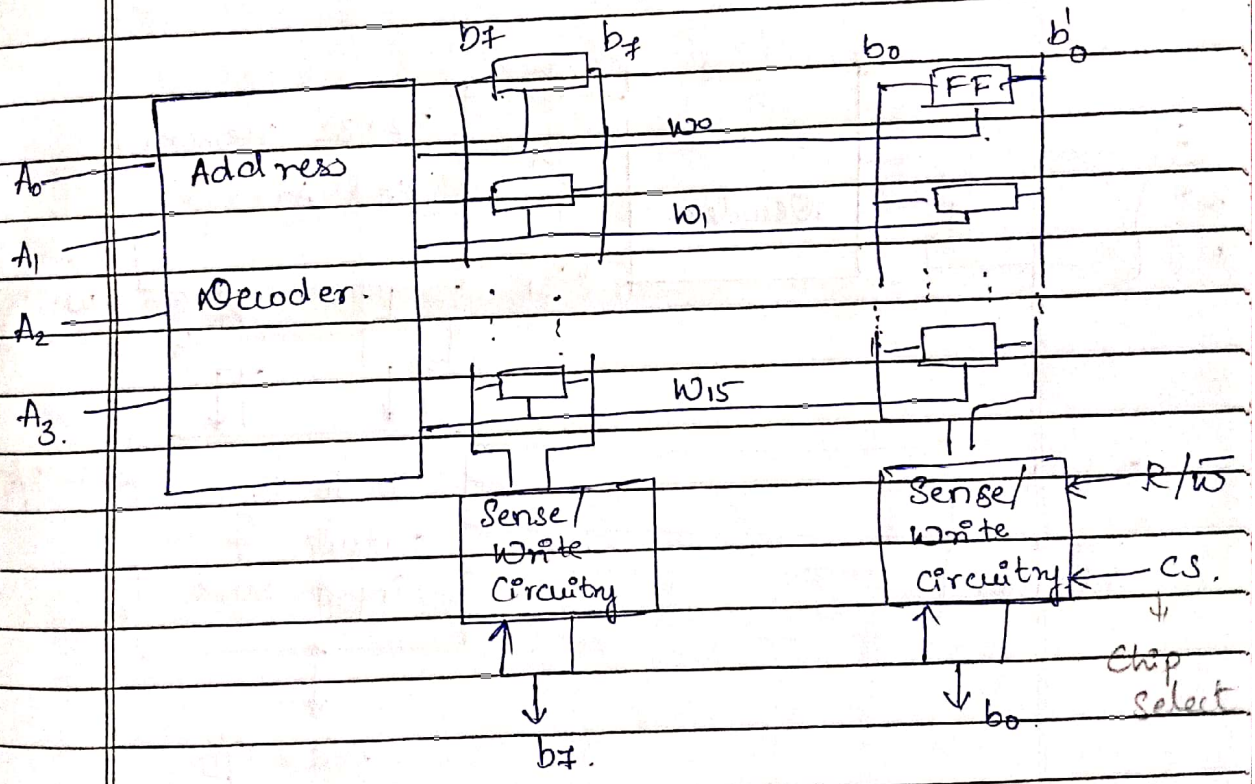
→ The address generated by the processor is called virtual address.

(2) SEMICONDUCTOR RAM MEMORIES.

(i) Internal organization of memory chip.

Question : Write the organization of a 16x8 memory chip.

16x8 → 16 words of 8 bits each



INTERNAL ORGANIZATION OF 16X8 MEMORY CHIP.

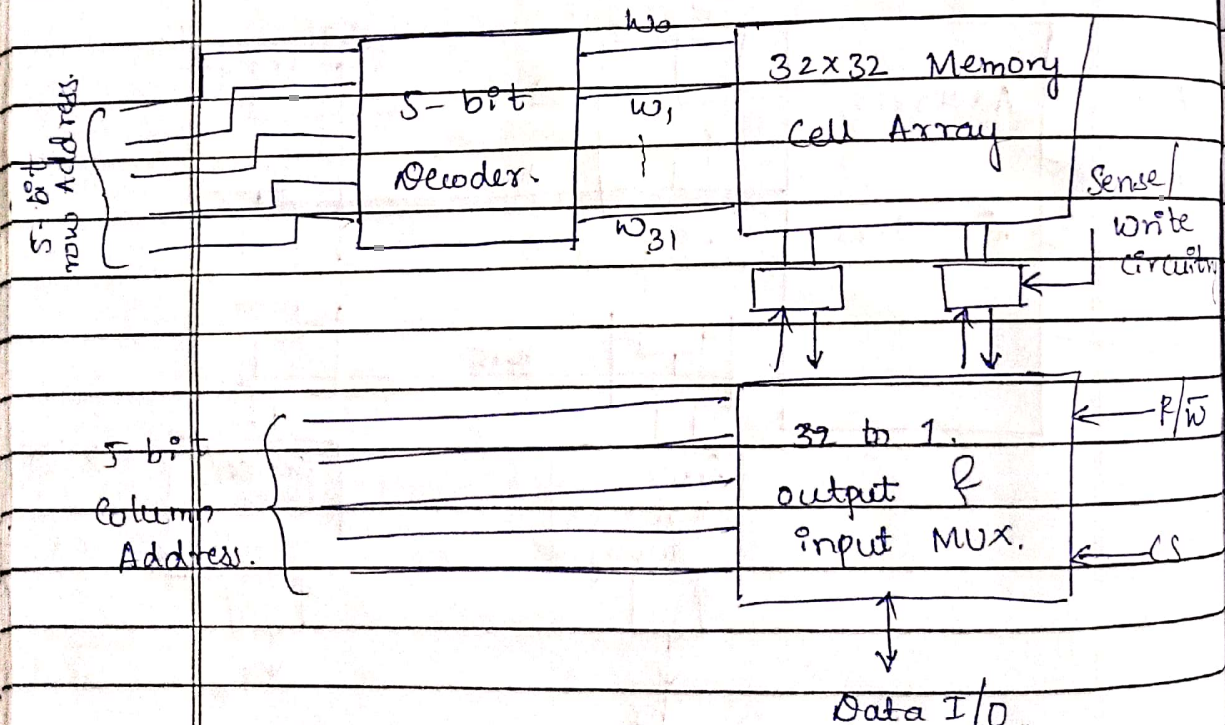
For the memory organization 32x32 Identify length of data bus (32), length of address bus (5) & number of addressable location (32).

32 words of 32 bits each.
length of data bus = 32 bits.
length of address bus = 5
Number of addressable locations = 32 words.

$2^5 = 32$.

⇒ During read operation the sense/read circuits, read information stored in the cells selected by the word line & transmit this information to the output data lines.

Organization of 1K×1 Memory chip:



1K×1 ⇒ 1024×1

128 × 8 = 1024

128 → Address line → 7

8 → data

1024 → capacity.

Address line.

How many external connections are required for 128×8 memory chip.

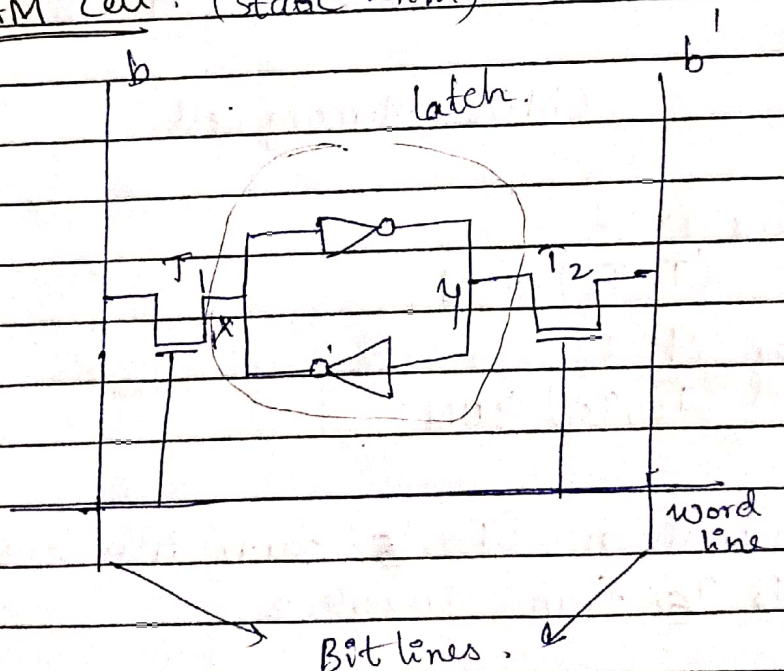
No. of address line + No. of data lines
+ Ground & power supply
$7 + 8 + 2 = \underline{17}$

Design 128×8 Memory chip using 1k×1 format. to reduce the no. of external connections.

⇒ 10-bit address is divided into two groups of 5-bits each to form the row & column addresses for the cell array.

STATIC MEMORY

SRAM Cell: (Static RAM)



⇒ T_1 & T_2 act as switches that can be opened or closed by controling the word line.

⇒ When word line is grounded, transistors are

Asha

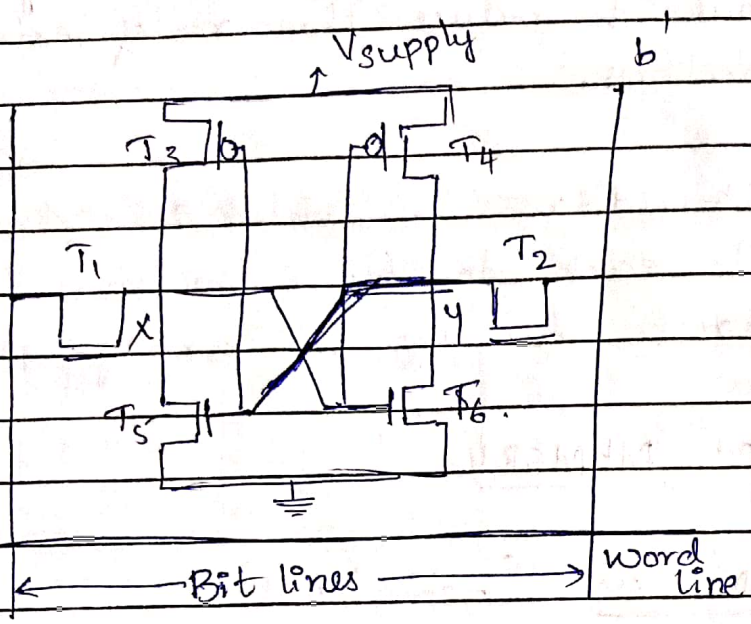
Date :
Page No :

closed and the latch retains its state.

(1) Read - { b=1, b'=0 } or { b=0, b'=1 }
↳ b & b' monitered by sense / write circuits.

(2) Write - Appropriate value set on b & b' & ~~write~~ word line is activated

23/9/19



CMOS Memory cell.

For x = 1
$(T_3, T_6) \Rightarrow$ ON.

For y = 1
$(T_4, T_5) \Rightarrow$ ON.

→ Transistors (T_3, T_5) form inverter 1 and (T_4, T_6) forms inverter 2.

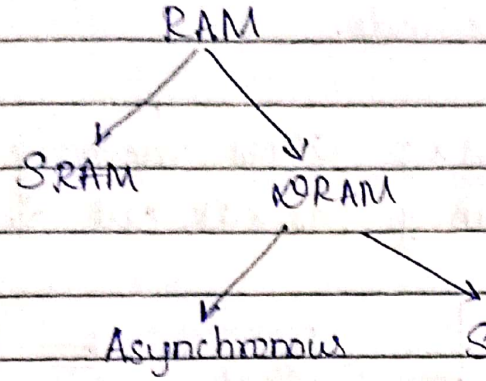
→ Voltage at X is maintained high by having T_3 & T_6 in ON state & T_4, T_5 in OFF state

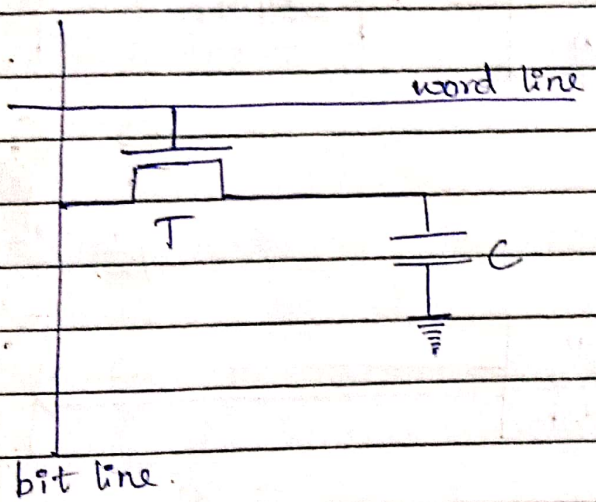
- If T_1 & T_2 are turned on (closed) b will have high signal.

Advantages of CMOS Memory cell

* Low power consumption.
* Can be accessed quickly (in nanoseconds)

ASYNCHRONOUS DYNAMIC RAM (DRAM).





Single Transistor DRAM Cell

If capacitor charge < Threshold ⟹ 0.

If capacitor charge ⩾ Threshold ⟹ 1.

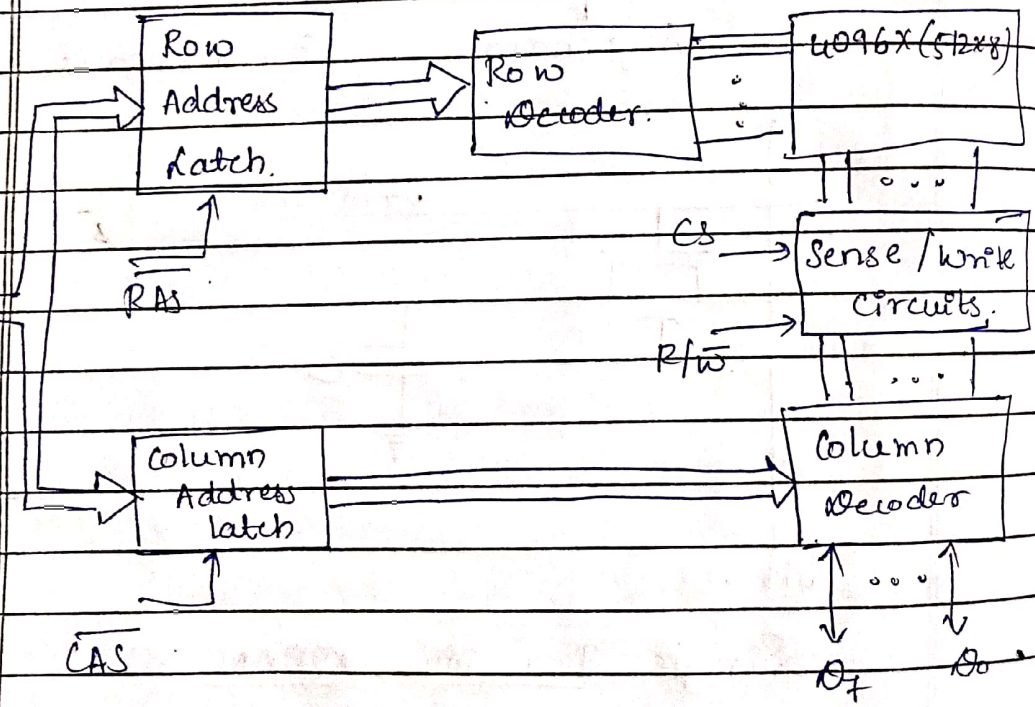
Date :
Page No :

→ DRAM is cheaper.
→ Main memory is made up of DRAM.
→ Cache memory is made up of SRAM.

→ DRAM doesn't retain its state indefinitely.
↔ Information is stored in the dynamic memory cell in the form of capacitor charge & this charge will maintained only for 10's of miliseconds.

24/9/19

Design a 2M x 8 DRAM memory chip in the form of 4K x 4K cell array.



2M x 8
↳ $2^{21} \times 8$

4K x 4K ⇒ 4096 x 4096.

2^{12}

4096 bits → 512 x 8.

Date :
Page No :

During R/$\bar{w}$ operation row address is applied first to select the word & column address is applied to select the byte.

Design a 16M×8 using 32K×4K Cell Array.

16M

2^{24}

32K — 2^{15}

Design 64M chip can be organized as ~~Memory using~~ 16M×4, 8M×8, 4M×16

16M → 2^{24} ← 2^{12}

2^{12}

16M×4 → 8K×2K 2^{7}

Design a 16×8 Memory chip

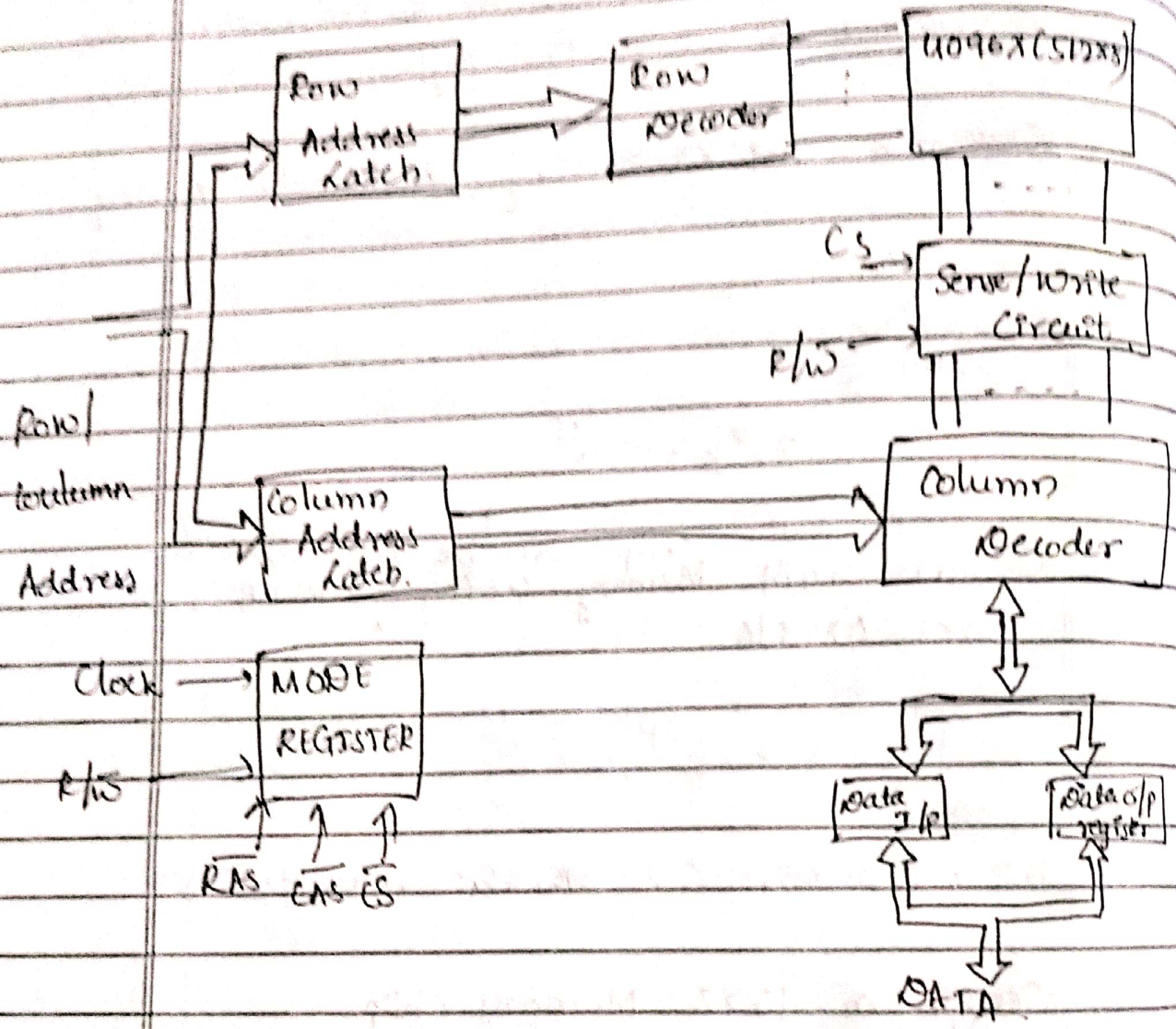
→ DRAM chip is organised to read or write no. of ~~the~~ bits in parallel to reduce no. of memory chip.

FAST PAGE MODE.

↳ Burst (words).

In fast page mode bytes are transferred sequentially by applying consecutive column addresses under the control of successive CAS (Column Address Store)

SYNCHRONOUS DRAM.

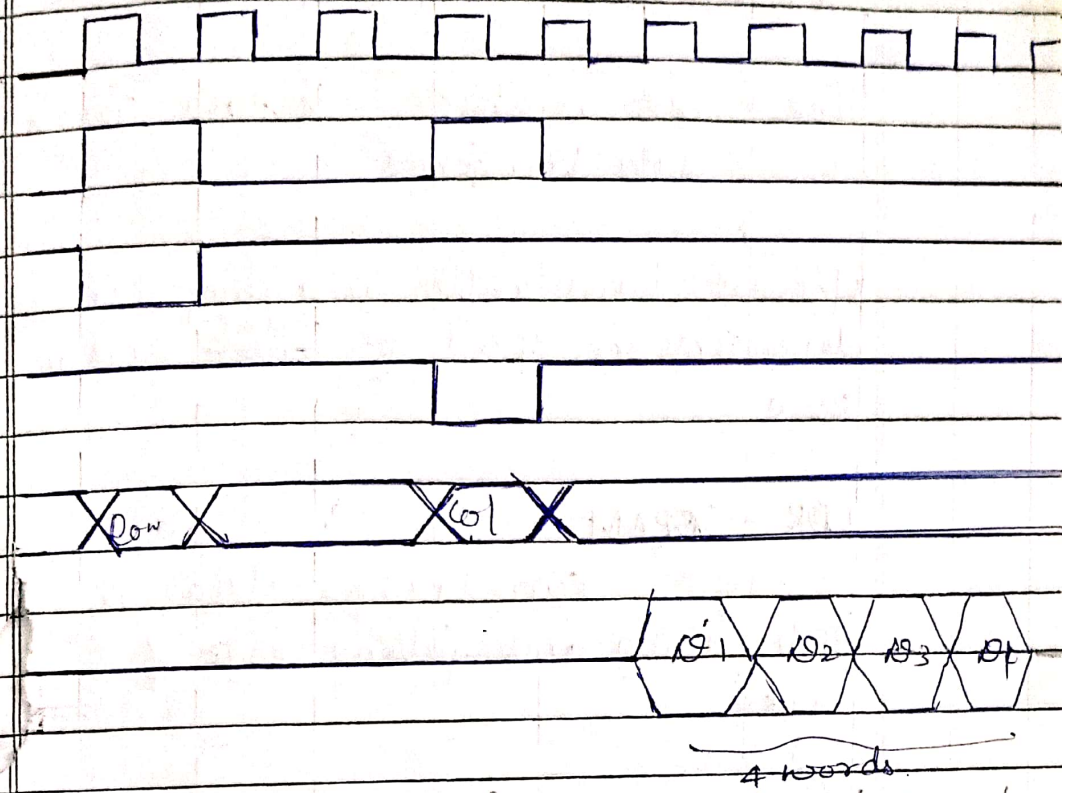


⇒ DRAM's are directly synchronised by clock signal mode register is used to store the control information.

Here, counters are used to store the no. of clock pulses required for data transfer operation.

Date :
Page No :

Timing Diagram for read operation for a burst read of length 4 in SDRAM



Write the timing diagram for a burst write read of length 4 in SDRAM.

→ Major Manufacturers like intel produces SDRAM chip in which the memory takes two or three clock cycles to activate the selected row.

Latency: Memory latency is a amount of time taken to transfer one word of data to/from memory. (must be low)

Latency = 5 clock cycle to transfer 1st word.

In block transfers latency is the time taken to transfer first word of data.

Date :
Page No :

Memory Bandwidth: (High)

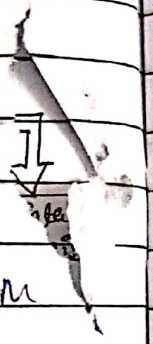
Number of bits / bytes that are transferred to / from memory per second.

B/w = rate at which data transferred × data Bus width

Effective band width in a computer is dependent on speed of ~~burst~~ burst and memory speed.

DDR - SDRAM

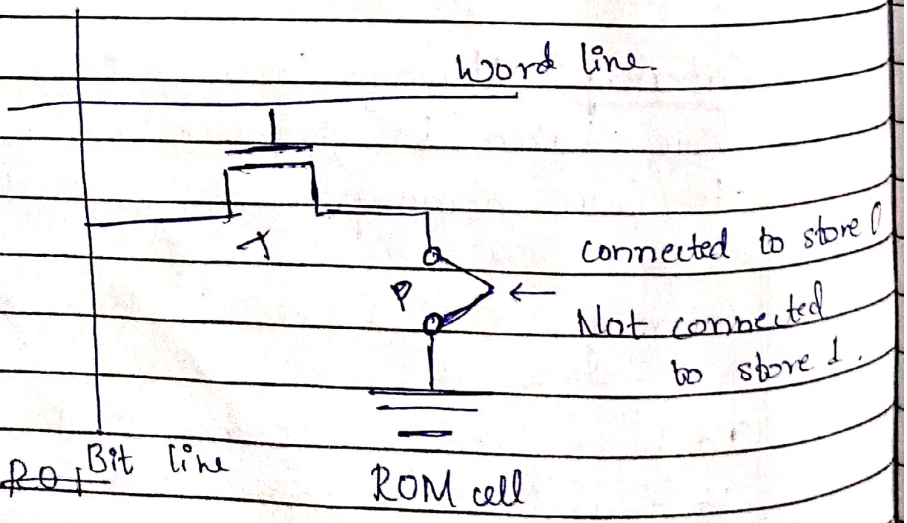
can transfer data on both rising and falling edge of clock.



B/w of DDR-SDRAM = 2× B/w of SDRAM

→ It is used in graphic terminals & machine learning systems.

ROM: (Read only- Memory)



→ ROM is non-volatile.

→ Small amount of non-volatile memory called ROM is provided that holds the instructions whose execution results in loading the boot program from the list.

→ Bit line ~~is~~ is connected to the power supply through a register.

→ Cells circuit at the end of bit line generates proper output.

27/9/19

Types of ROM:

- PROM
- EPROM
- EEPROM.
- Flash Memory (Flash Card & Drives)

PROM: Programmable Read only Memory.

1. Data loaded by user.
2. Programmability achieved by inserting Fuse at "p".
3. Memory contains 0's before programming.
4. User inserts 1's by burning out Fuse using high current pulse.
5. Irreversible process.

Advantages:

1. Flexibility, faster.
2. Less expensive.

→ Programmable ROM is used in cell phones where small no. of ROM's are required.

Date :
Page No :

EPROM: Erasable PROM

1. Allows stored data to be erased by exposing chip to uv light.
2. Charge injected into special transistor.

Advantages:

1. Flexibility, Retains information for long time

Disadvantage:

1. Chip must be physically removed from circuit for reprogramming.
2. Entire contents erased.

→ The EPROM chip allows stored data to be erased & new data to be loaded.

EE PROM: Electrically Erasable PROM.

1. Can be programmed & erased electrically.
2. Erasing can be done selectively.

Disadvantage:

1. Requires different voltages for reading, writing erasing stored data.

→ EEPROM doesn't have to be physically removed & cell contents can be electrically erased

Flash Memory:

Flash Card:

1. Larger module to mount flash chips on small card.
2. Card is plugged into accessible slot.
3. 1 minute of music → 1 MB
   1 hour of music → 64 MB

Date :
Page No :

Flash Drive

1) Emulates harddisk
2) Shorter seek & access times
3) Low power consumption
4) Weakneens after it has been written 1 million times

Examples: Pendrive, External hard drive

⇒ hand-held computers, cell-phones, digital cameras
→ MP3 players ⇒ Stores Music.
→ Digital cameras ⇒ Image data.

Disadvantage of flash drive over HDD (Hard Disk Drive)

⇒ Small capacity and higher cost per bit.

SPEED, SIZE, COST.

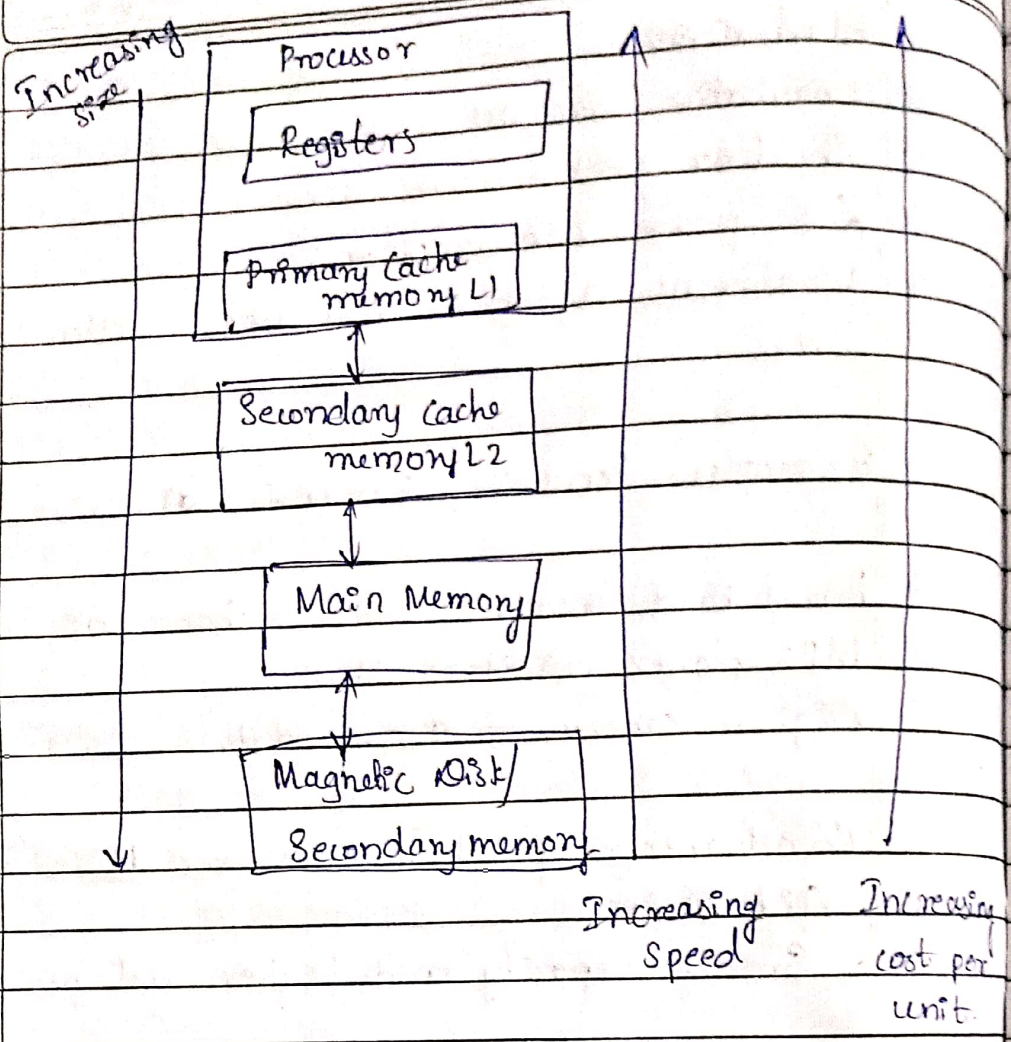
① SRAM	① Magnetic disk	① SRAM.
② DRAM	② SRAM & DRAM	② DRAM
③ Magnetic disk		③ Magnetic disk.

Expensive

Imp MEMORY HIERARCHY.

⇒ To speed up the program execution, instructions and data will be stored in cache.

Date :
Page No :



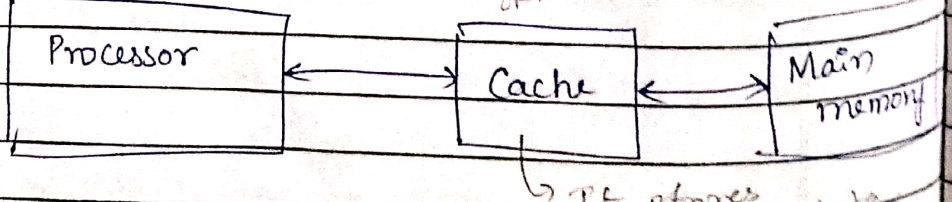
CACHE-MEMORY

- → Mapping function.
- → Replacement algorithms

① Locality of Reference → temporal
→ Spatial

② Block – Set of instructions.

③



④ Mapping function

Date :
Page No :

(5) Replacement Algorithm
(6) Read Hit
(7) Read Miss
(8) Write Hit
(9) Write Miss
(10) Write through protocol
(11) Write - Back protocol / Copy - back
(12) Load through / Early restart

(1) Locality of reference is a concept that many instructions in the localised area of the program are executed repeatedly during same time period & the reminder of the program is accessed relatively infrequently.
Temporal: Recently executed instruction is likely to be executed very soon.
Spatial: Instructions closed to the recently executed instruction are likely to be executed soon.

(2) Block: Block refers to set of contiguous address location of same size.

(4) Mapping function: It specifies the correspondance b/w main memory blocks & the cache block

(5) Replacement Algorithm: are used to replace cache block when a word not in cache is referenced.

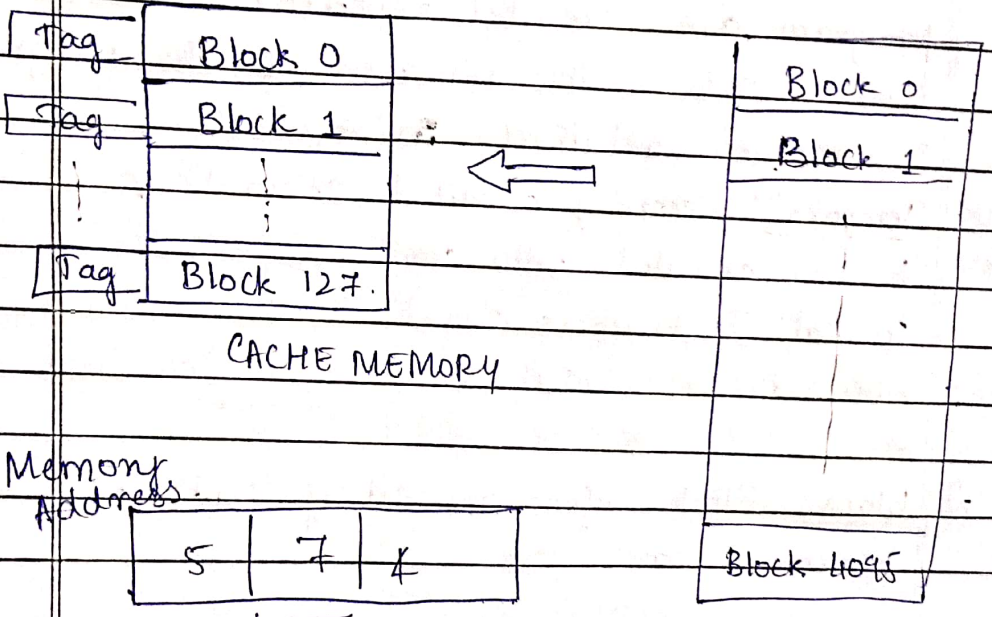
(10) Write through procal: Cache location & main memory ~~access~~ are updated simultaneously.

(ii) Write back protocol: Only cache location is updated & it is marked with dirty or modified bit, main memory location is updated later when the cache block is being replaced.

MAPPING FUNCTIONS

1. DIRECT MAPPING.
2. ASSOCIATIVE MAPPING.
3. SET ASSOCIATIVE MAPPING.

Direct Mapping:



Main memory ⇒ 4096 blocks.

1 blocks = 16 words ⇒ Total Memory = 4096 × 16 words.

Cache memory ⇒ 128 Blocks. = 128 × 16 = 2048 words.

Memory address size ⇒ 16 = 2^{16} = 65536

Date :
Page No :

j mod 128

j → Block.

19 → 54 → 6

10011 011011O 0110

Processor wants to read 6th word from 54th block present in 19th tag.

j mod No. of cache blocks

→ Tag bits are used to identify whether the addressed word is present in cache.

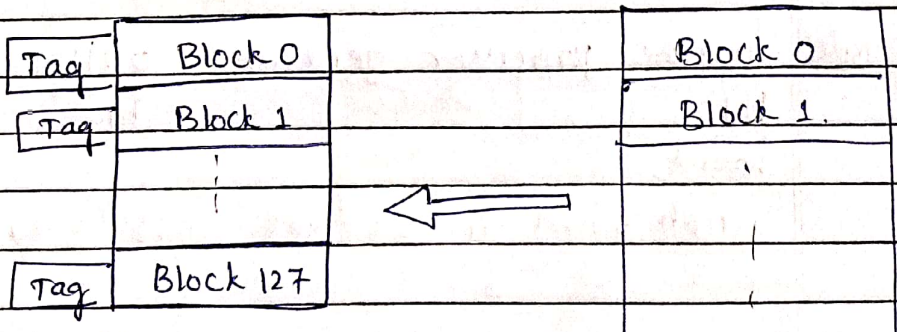
→ Direct mapping is easy to implement but not very flexible.

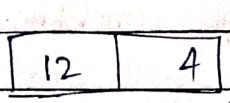
→ Memory wastage happens because 128 blocks has to be store in block 0. even if other cache blocks are empty.

1/10/19

MAPPING FUNCTION

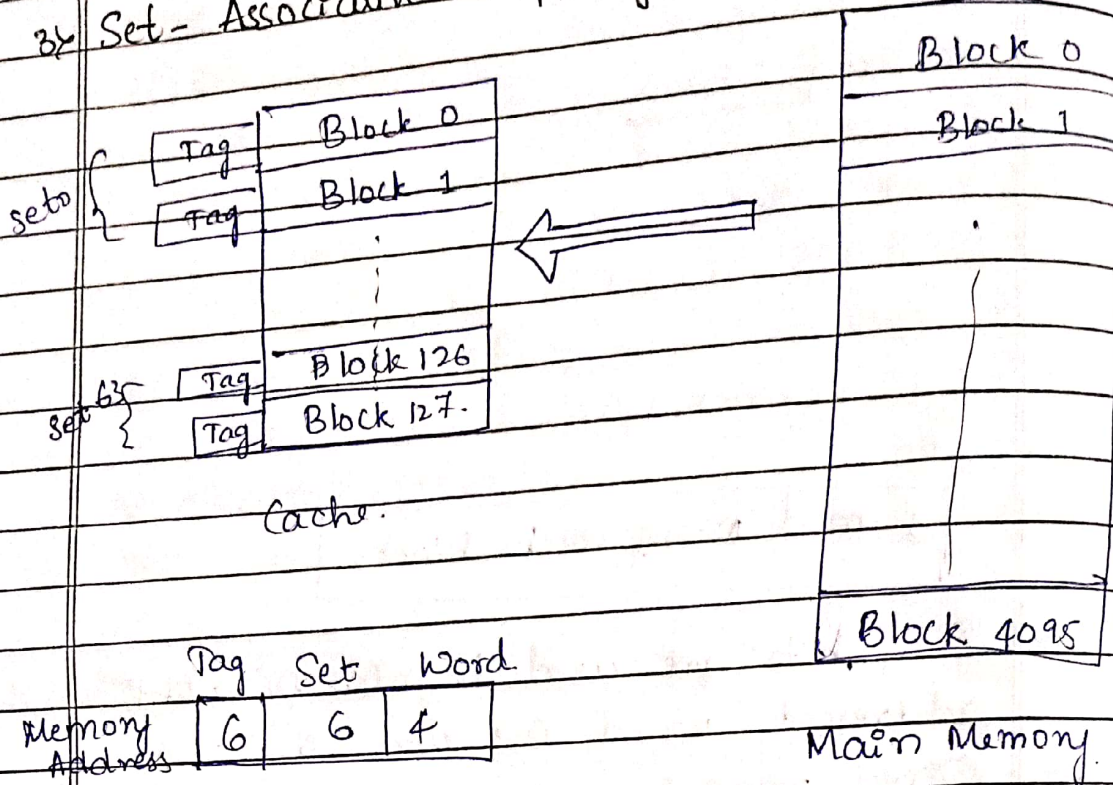
2) ASSOCIATIVE MAPPING.





Memory address.

3. Set- Associative Mapping.



Memory Address	Tag	Set	Word
	6	6	4

→ In Associative Mapping, ~~here~~ main memory can be placed into any cache block position.

→ 12 tag bits are used to identify memory block when it is resident in the ~~m~~ cache memory.

Example: Suppose processor generates 001101 (tag) 010111 (set) 1010 (word)

10th word in 23rd set, 13th tag.

→ In Set associative mapping, combines direct & associative mapping.

→ Cache blocks are grouped into sets.

→ In the example given two tags in set 23 searched to see if either one matches tag 13.

→ Valid Bit is provided for each word indicating whether the block contains valid data.

Valid Bit = 1 ⇒ Data in cache & main-memory are consistent.

→ Cache coherence problem need to ensure that two different entities use the same copies of data.

DMA → Data Memory Access.

REPLACEMENT ALGORITHM:

1) LRU (Least Recently Used)

Block 0
Block 1
Block 2
Block 3.

Replacement algorithms are not required for direct mapped cache.
The aim is to keep track of the no. of blocks that are referenced. LRU algorithm overwrites the cache block that are not been referenced for longest time.
So a counter is kept with each block to

Date :
Page No :

track the no of references to the cache block.

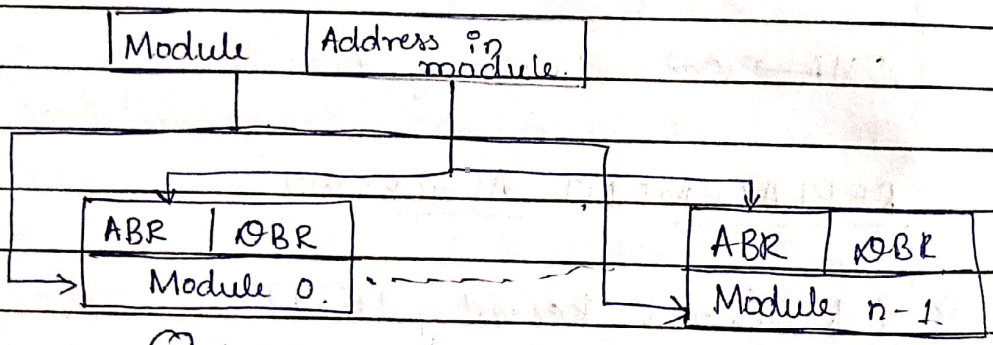
(2) Oldest Block is replaced

In oldest block algorithm, the block that is there in the cache for the longest time is removed.

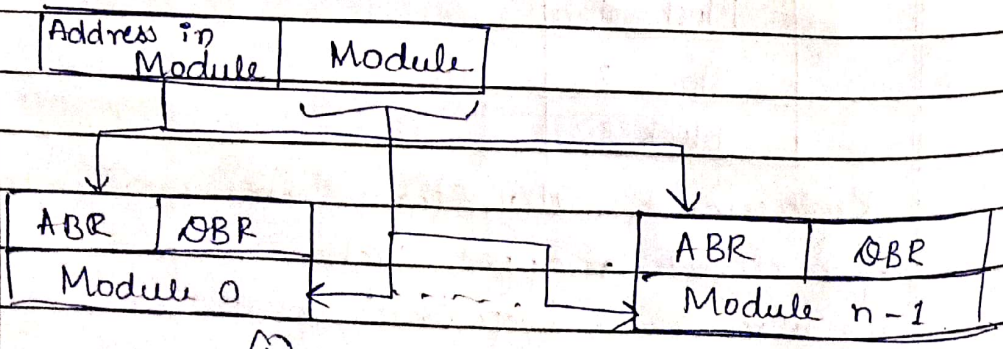
The simplest algorithm is to randomly choose the block to be overwritten.

Performance Considerations:-

1) Interleaving



(a) Without interleaving.
→ Consecutive words in a Module



(b) With Interleaving
Consecutive words in consecutive modules.

ABR → Address Buffer register.
DBR → Data Buffer register.

⇒ In the diagram main memory is structured as a physically separate module.

Example for (a).
1000:- 1st word. } 1 module.
1004:- 2nd word }

Example for (b).
1000:- 1st word ⇒ Module 1
1004: 2nd word Module 2

→ In memory interleaving consecutive words are located in successive modules so that several modules are kept busy at any given time as all the consecutive locations will be accessed parallely.

Illustrate the benefit of interleaving with an example.
Assume cache with 8 word blocks. 1 clock cycle is to send address to main memory
1st word accessed in 8 clock cycles.
1 Clock cycle needed to send 1 word to cache
Compute time (in clock cycles) needed to transfers a block of data from main memory to cache with / without interleaving.

Without interleaving:

1 cache block = 8 words.

$1 + 8 + (7 \times 4) + (1 \times 8) = 45$

- 1 → Clk cycle to send address to main memory.
- 8 → Clk cycle to access 1st word.
- (7×4) → Clock cycle to transfer remaining 7 words
- (1×8) → Clock cycle to transfer block of 8 words from Main memory to cache

With interleaving:

$1 + 8 + (\overset{1}{\cancel{7 \times 1}}) + 1 = 11$

⇒ Interleaving reduces the block transfer time by more than a factor of two.

2) **Hit rate & Miss-penalty:**

Hit rate = $\frac{\text{Number of hits}}{\text{Attempted Memory Accesses.}}$ ⩾ 0.9

Miss rate = $\frac{\text{Number of Misses}}{\text{Attempted Memory Accesses.}}$

⇒ Performance is affected by hit rate & miss rate.

⇒ Miss-penalty indicates extra time needed to bring the desired data into the cache memory.

Date :
Page No :

Average access time experienced by processor $= hC + (1-h)M$

where, $h \Rightarrow$ Hit rate
$C \Rightarrow$ Time to access information in cache
$M \Rightarrow$ Miss penalty.

* Suppose 1 Memory Read access = 10 clock cycles.
Time needed to load a block into cache = 17 cycles.
Assume 30% of instructions perform read/write operations
$\Rightarrow$ 130 Memory accesses
h for instructions = 0.95
h for data = 0.9
Compute performance improvement due to cache.

$\Rightarrow$ Time without cache = No. of memory access $\times$ No. of clock cycles for each access
$= 130 \times 10$

Time with cache = (No. of memory access for instructions / 100) $\times (hC + (1-h)M)$ for inst + (No. of memory access for data / 100) $\times (hC + (1-h)M)$ for data

$M = 17$ cycles.
$C = 1$

No. of memory access for data. 30.

$hC + (1-h)M + hC + (1-h)M$
$= 0.95 \times 1 + (1-0.95) \times 17 + 0.9 \times 1 + (1-0.9) \times 17$
$= 0.95 + 0.05 \times 17 + 0.9 + 0.1 \times 17$
$= 4.4$

$\Rightarrow$ Answer $= \dfrac{130 \times 10}{100 \times (0.95 + 0.05 \times 17) + 30 \times (0.9 + 1.7)}$

$= 5.04.$

Date :
Page No :

* Calculate performance improvement of memory with & without cache given the following data:-

1 cache blocks = 8 words
Hit rate for instructions = 0.95
Hit rate for Data = 0.9
Number of processor cycles to transfer 8 words Block to/from main memory = 60
Number of processor cycles to access single word in main-memory without cache = 36
Number of instructions accesses = 100 } out of
Number of Data accesses = 30 } 130.

130 memory access 30% are read/write transfer of data

Time without cache = ~~No of 60~~ x 36
= ~~2160~~ 4680

Time with cache = 100 x (0.95 + (1-0.95) 60)
+ 30 x (0.9 + (1-0.9) 60)
= 100 x 3.95 + 30 x 6.9
= 602

Time without cache / Time with cache = 4680 / 602 = 7.77.

This indicates that the memory with cache performs 7.77 times faster than memory without cache

Assume system bus clock is 4 times slower than processor clock

Hit rate Improved by:

1. Cache larger ⇒ Expensive
2. Transfer large blocks b/w main memory & cache.
   ↳ Miss penalty may increase.
3. Load through protocol. (Processor doesn't wait for entire block to be loaded)
   ⇒ Processor stats reading word immediately after it is brought into cache.

Cache on processor chip.

Due to space requirements cache size is limited and pentium processor uses L1 & L2 cache to increase the memory access speed.

L1 cache } pentium, ARM
L2 cache }

$$t_{ave} = h_1C_1 + (1-h_1)h_2C_2 + (1-h_1)(1-h_2)M$$

t_{ave} ⇒ access speed.

where,
h_1 ⇒ Hit rate of L_1
C_1 ⇒ Time to access information in L_1 cache
h_2 ⇒ Hit rate of L_2
C_2 ⇒ Time to access information in L_2 cache
M ⇒ Miss penalty
t_{ave} ⇒ average access time by processor in system with 2 cache

Including L2 cache reduces the average access time of memory by the processor.

Other Enchancement.

1. Write buffering.
2. Pre fetching.
3. Lock-free cache.

Explain the other methods for improving the performance of memory.

→ Write buffering: The write buffer stores the write requests temporarily & doesn't wait for memory function complete (MFC)
⇒ Read requests are high priority than write requests.

→ To avoid processor waiting prefetching of data into cache is done by compiler by inserting prefetch instructions in the program

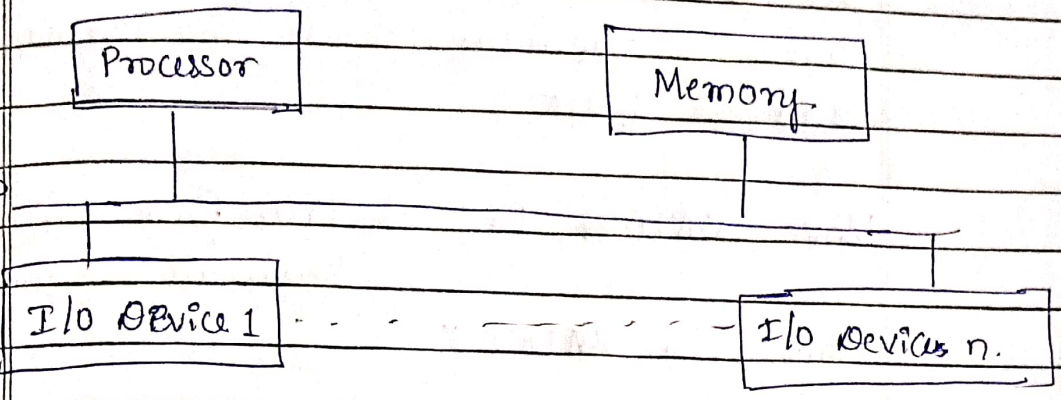
→ Lock free cache allows the processor to access the cache when a miss is being serviced.

UNIT - 4

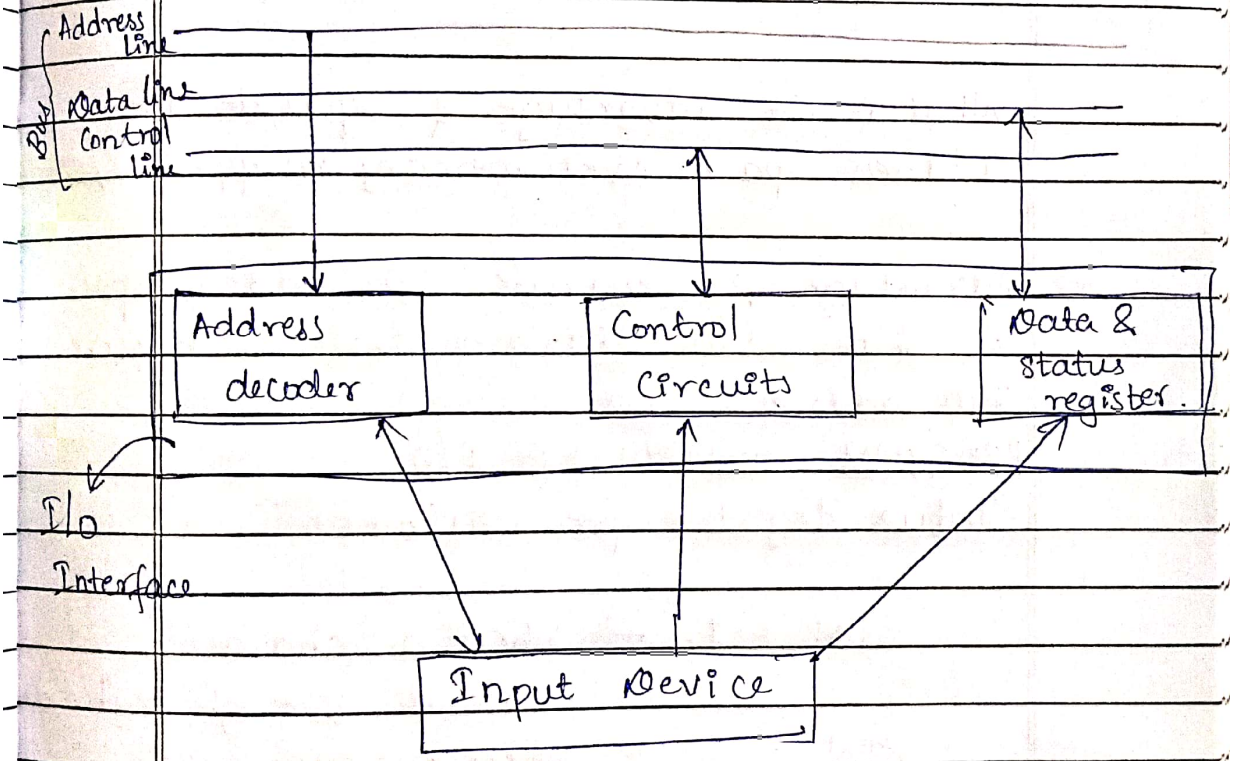
I/O Organization & Arithmetic

Date :
Page No.

Accessing I/O Devices



Single - Bus structures



I/O Interface for an I/O Device

→ The I/O device examines the lower ordered bits of address line to determine whether they should respond

Date :
Page No :

→ Memory mapped I/o :- Memory & I/o device share same address space. It is an arrangement where the I/o devices & memory shade the same address space.

Move DATAIN, R0 ⇒ Transforms from keyboad to processor

Move R0, DATAOUT

DATAIN ⇒ I/p buffer

What is the advantage of separate I/o address space over memory mapped I/O

⇒ Advantage of separate I/o address space is that I/o addresses deals with pure I/o address lines.

PROGRAM - CONTROLLED I/O.

Status Register for keyboard

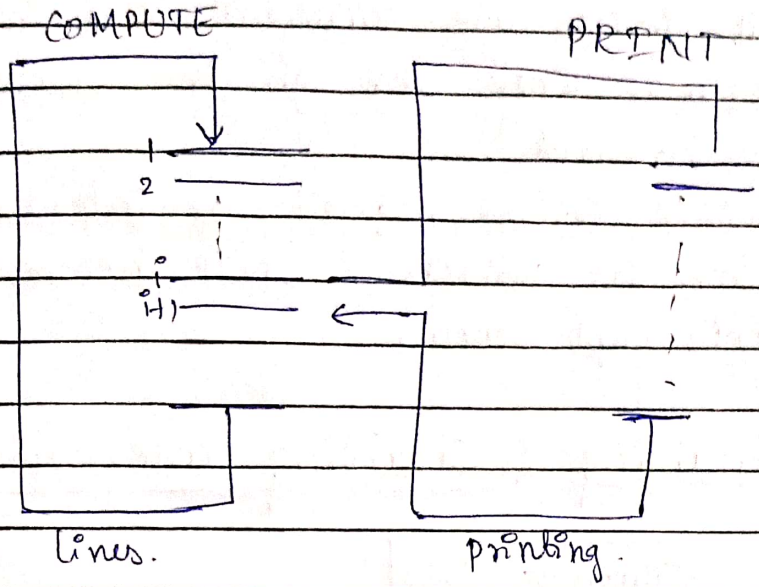
SIN = 1. ⇒ when a character entered through keyboard

SIN = 0 ⇒ when processor reads character.

Interrupts :

⇒ A program controlled I/O repeatedly reads the status register which leads to

slower execution. To overcome this Interrupts are used in which the I/O device sends a special signal when it is ready for a data transfer.

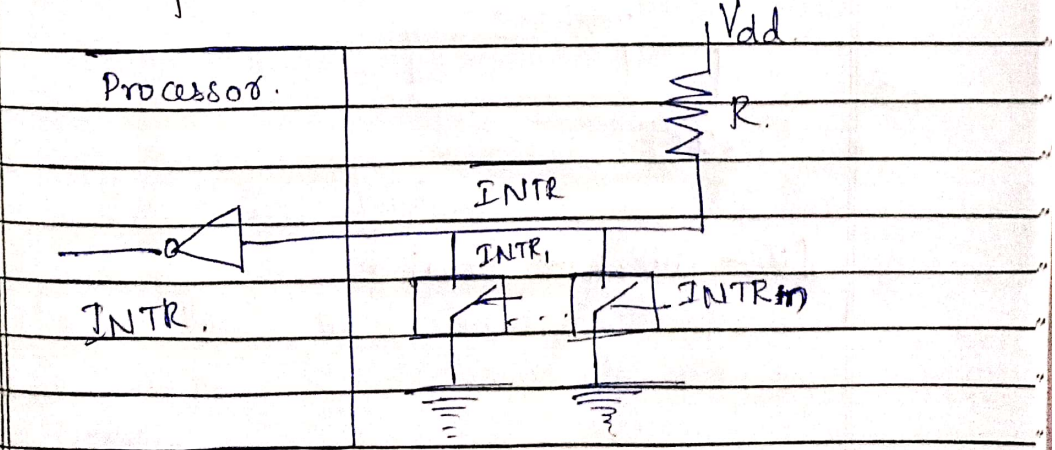


ISR → Interrupt service Routine.

⇒ ISR is a program that is executed when interrup occurs

Interrupt latency is the delay b/w time & interrupt request is recieved and the start of execution of ISP

Interrupt hardware.



Date :
Page No :

Interrupt request line implementation using open drain Bus

$INTR = INTR_1 + INTR_2 + \cdots + INTR_n$

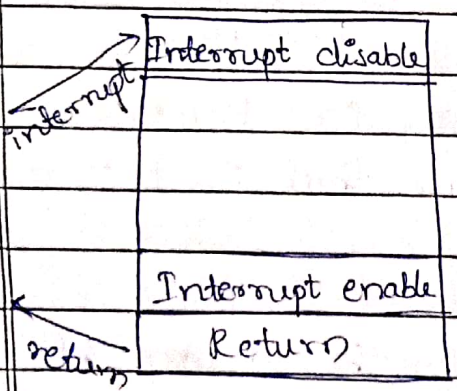
→ All devices are connected to a interrupt request line through the switches to the ground.

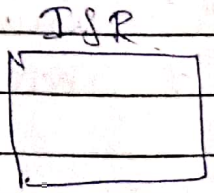
→ When device requests an interrupt it closes its switch & INTR becomes 0 & interrupt occurs.

11/10/19

Enabling & Disabling interrupts

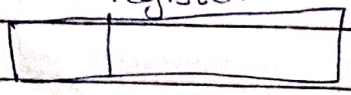
1)





2)

Processor status (PS) register



IE

IE - Interrupt enable

Date :
Page No :

→ In option 1 processor uses interrupt disable instruction has 1st instruction in the ISR.

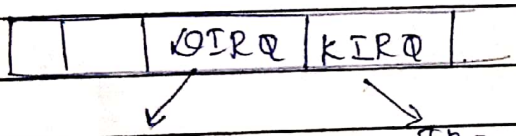
→ In Option 2 the IE bit is said to 1 in the PS register to enable further interrupts.

IE = 1 ⇒ Processor enables interrupts
IE = 0 ⇒ Processor disables interrupts

Handling Multiple devices.

- Polling.
- Vectored interrupt.
- Priority (Interrupt Nesting)
- Daisy chain (Simultaneous requests).

⇒ Polling



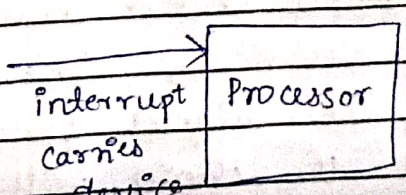
IRQ bit for Display.

IRQ bit for keyboard.

(IRQ → Interrupt Request)

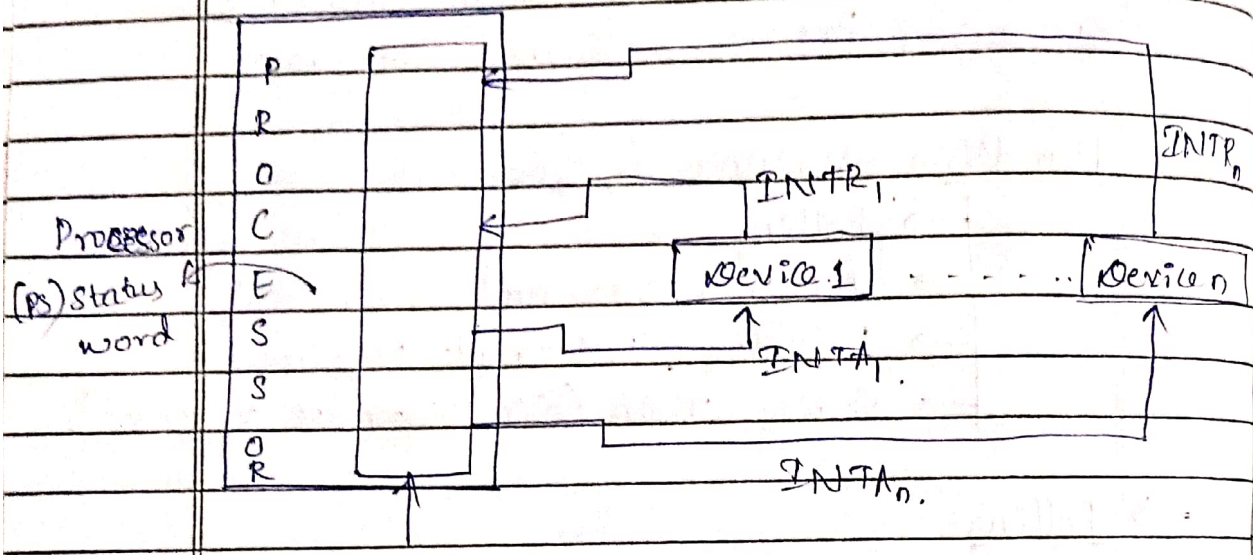
* In polling scheme.
* The first device encountered with its IRQ bit set is the device that should be services.

⇒ Vectored interrupt:



* In vectored interrupt scheme device requesting interrupt identifies itself to the processor.

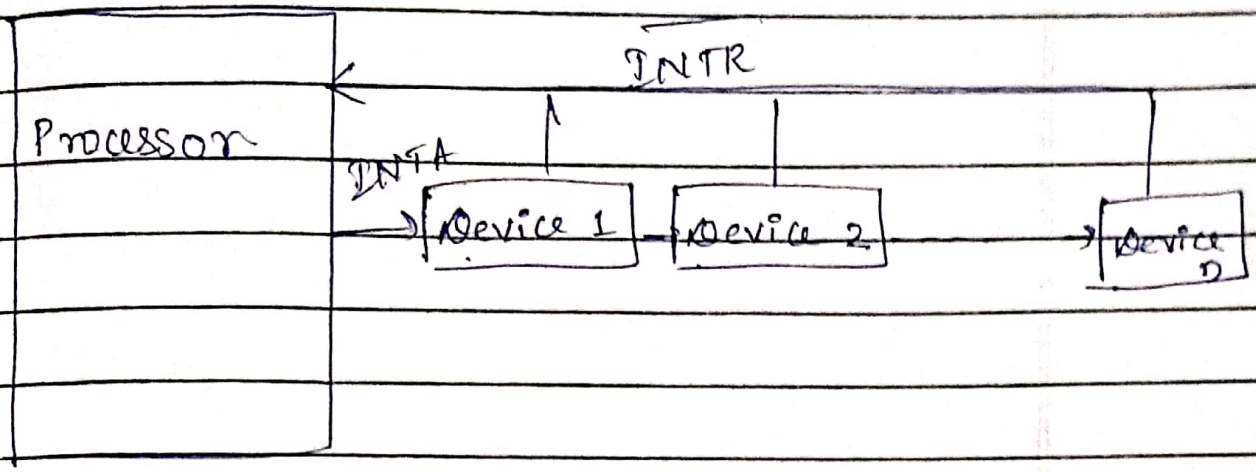
⇒ Priority (Interrupt Nesting)



* In a multiple priority scheme processors priority level is changed according to the device priority that it is servicing.
* In this scheme request is accepted only if it has a higher priority level than that is currently assigned to the processor.

⇒ Daisy chain

* In daisy chain arrangement the device that electrically closest to the processor has the highest priority. The second device along the chain will have second highest priority.



* Devices can be organized into groups and each group is connected at a different priority level.
Within a group devices are connected in daisy chain.