

COA
Chinnay
Gir

Chapter-02

Computer Evolution and performance

① Enlist the processors from 4 bit to 64-bit. — 02

1971 → 4004 → 4 bit

1972 → 8008 → 8 bit

1978 → 8086 → 16 bit

1982 → 80286 → 16 bit

1986 → 80386 → 32 bit

1993 → Pentium → 32 bit

2006 → Core-2 → 64 bit

2007 → Core i3 → 64 bit

2009 → Core i5 → 64 bit

2010 → Core i7 → 64 bit

② Mention the categories of IAS instruction set with example (opcode, symbolic presentation and description) — 03

IAS computer had a total of 21 instructions.
These instruction categories are:

(i) Data transfer:

Move data between memory and ALU registers or between two ALU registers.

00001010 \rightarrow LOAD M8 \rightarrow Transfer contents
 of register M8 to the accumulator AC

(ii) Unconditional branch:

Control unit executes instructions in sequence from memory. This sequence can be changed by a branch instruction.

00001101 \rightarrow jump (x, 0:19) \rightarrow Take next instruction from left half of M(x).

(iii) Arithmetic:

operations performed by the ALU.

Contents of the left half registers are: addition, subtraction, multiplication, division, etc.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

iii) conditional branch

The branch can be made dependent on a condition, thus allowing decision points.

$00001111 \rightarrow \text{JUMP } M(x, 0:19) \rightarrow$ If number in accumulator is non-negative, take next instruction from left half of $M(x)$

iv) Arithmetic:

operations performed ~~by~~ by the ALU.

$00000101 \rightarrow \text{ADD } M(x) \rightarrow$ Add $M(x)$ to AC; put the result in AC

v) Address modify: permits addresses to be computed in the ALU and then inserted into instructions stored in memory.

$00010010 \text{ STOR } M(x, 8:19) \rightarrow$ Replace left address field at $M(x)$ by 12 rightmost bits of AC.

3) Describe peripheral devices. iii 03

Peripherals

peripheral devices are external hardware components that connect to a computer or other digital device to enhance its capabilities.

These devices provide additional functionality beyond the core functions of the computer.

Generally peripheral devices, however, are essential for computer to perform its basic task, they can be thought of as an enhancement to the user's experience.

It can be classified into many categories:

(i) Input devices:

Keyboard, mouse, scanner etc.

(ii) Output devices: Monitor, headphones, printers etc.

(iii) Storage devices: Hard disk, Magnetic tape, Flash memory.

(iv) Communication devices:

Modem, Router etc.

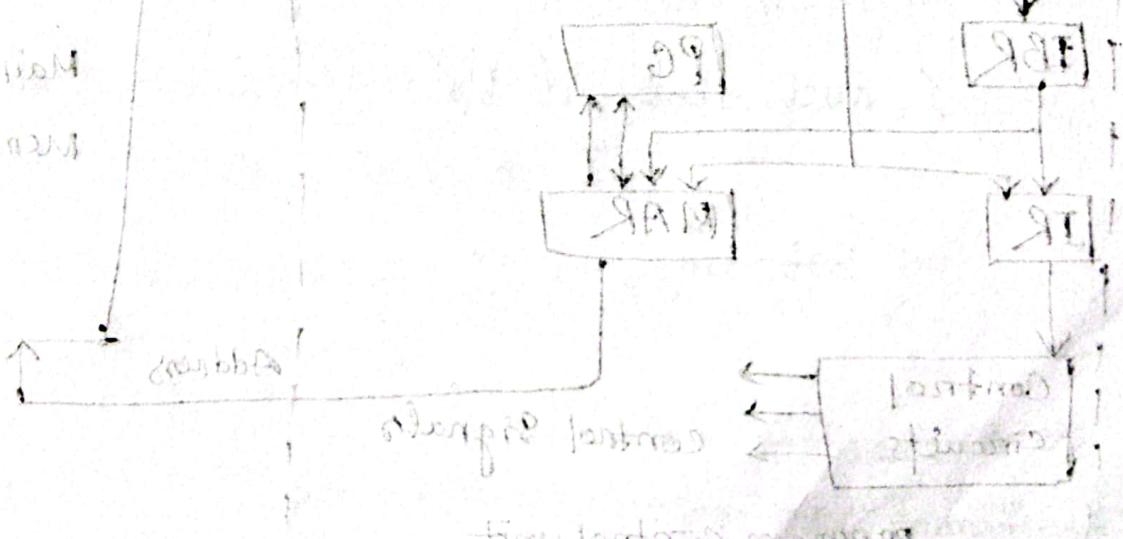
9

Explain Moore's Law.

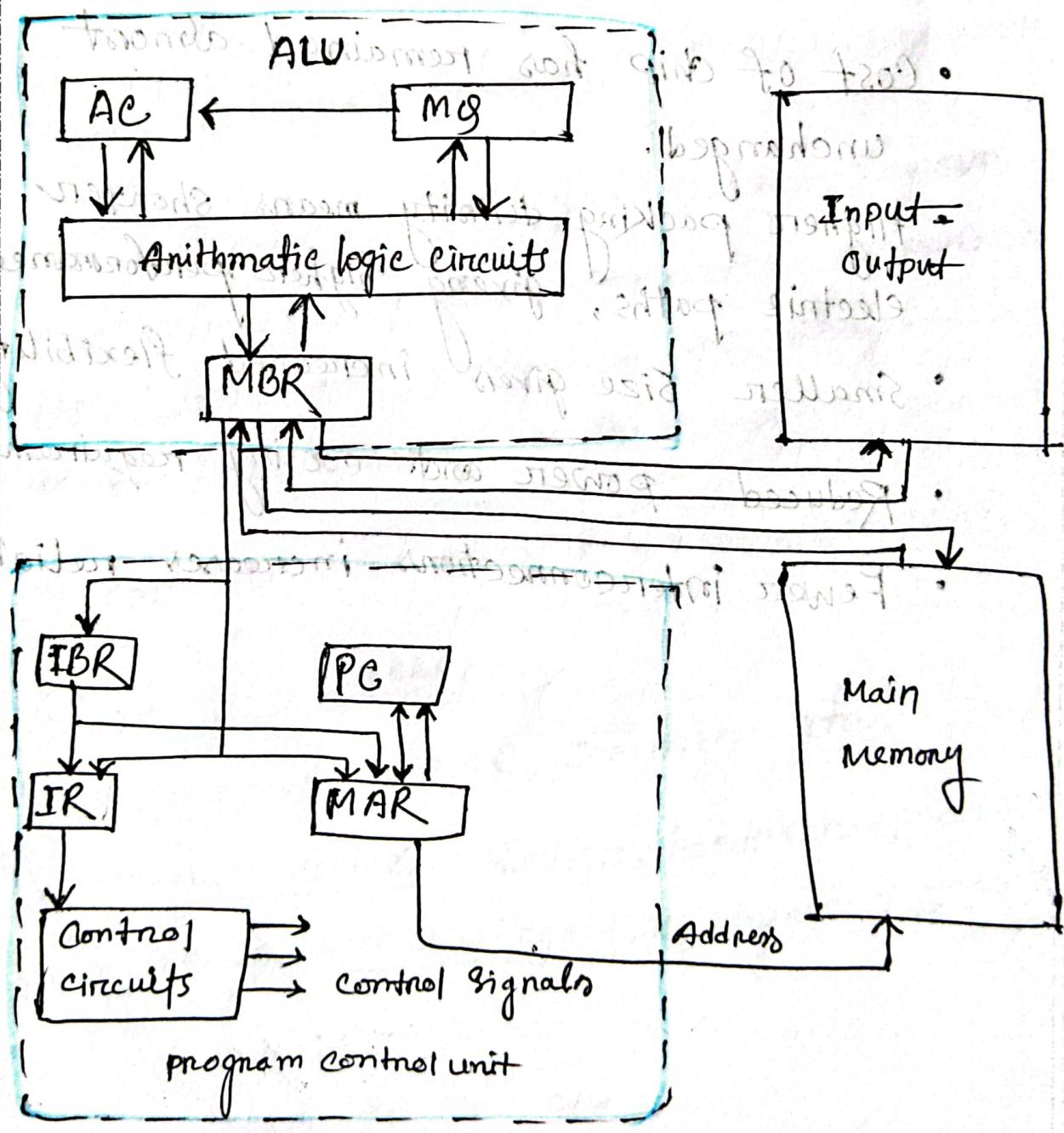
Moore observed that,

the number of transistors that could be put on a single chip was doubling every year and correctly predicted that this pace would continue into the near future.

- Cost of chip has remained almost unchanged.
- Higher packing density means shorter electric paths, giving higher performance.
- Smaller size gives increased flexibility.
- Reduced power and cooling requirements.
- Fewer interconnections increase reliability.



- 5) On the IAS, describe the process that the CPU must undertake to read a value from memory and to write a value to memory in terms of what is put into MAR, MBR, address bus, data bus and control bus. — 03



Sat / Sun / Mon / Tue / Wed / Thu / Fri

 Date: / /

Describe:

MAR \rightarrow memory address identify $\&$ data (data $\&$ location)

MBR \rightarrow identify $\&$ location $\&$ data $\&$ result

IR \rightarrow contains instruction of next execution

IBR \rightarrow contains right hand instruction temporarily

PC \rightarrow next $\&$ IR $\&$ $\&$ location.

(Address of next instruction)

AC/MQ \rightarrow operand, result hold $\&$ arithmetic operation $\&$

Ch-12 (7)

PC contains address of next instruction

Address move to MAR

Instruction Fetch \rightarrow Address placed on address bus

\rightarrow control unit requests memory read

\rightarrow Result placed on data bus, copied to

MBR then to FR

Meanwhile PC incremented by 1

Data Fetch \rightarrow IR examined

Indirect cycle \rightarrow If indirect addressing indirect cycle is performed

\rightarrow Right most N bits of MBR transferred to MAR

\rightarrow control unit requests memory read

\rightarrow Result moved to MBR

- What are the roles of different register of control unit and the ALU

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

- ⑥ Enlist the registers of IAS computer with its functionality.

1.5

i) Memory Buffer register (MBR):

Contain a word to be stored in memory or send to the I/O unit. or is used to receive a word from memory or from the I/O unit.

ii) Memory address register (MAR):

Specifies the address in memory of the word to be written from or read into the MBR.

iii) Instruction register (IR):

contain the 8 bit opcode instruction.

iv) Instruction buffer register (IBR):

hold temporarily the right hand instruction from a word in memory.

v) Program Counter (PC):

contains the address of the next instruction.

vi) Accumulator (AC) and multiplier quotient (MQ):

most significant 40 bits are stored in AC and least significant in the MQ. (If result of multiplying two 40 bit numbers)

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

⑦ Mention the computer generation with its corresponding technology.

• Vacuum tube - (1946 - 1957)

• Transistor - (1958 - 1964)

• Small scale integration - (1965 -)

↳ upto 100 devices on a chip

• Medium scale integration - (- 1971)

↳ 100 - 3000 devices on a chip

• Large scale integration - (1971 - 1977)

↳ 3000 - 100,000 devices on a chip

• ~~Ultra~~ Very large scale integration - (1978 -)

↳ 1,00,000 - 100,00,000 devices on a chip

• Ultra large scale integration

↳ Over 100,000,000 devices on a chip

⑧ Give the characteristics of computer family.

i) Similar on identical instruction set:

Same set of machine instructions is supported on all members of the family. program that execute on one machine will also execute on any other.

ii) Similar on identical operating system:

The same basic operating system is available for all family members.

iii) Increasing speed:

instruction execution increase in going from lower to higher family members.

iv) Increasing number of I/O ports.

v) Increasing memory size.

vi) Increasing cost.

Q) Consider the below Computer Specification:

→ pentium Intel® Core™ i7 4.7 GHz

This is CPU. It is an Intel Core i7 processor running at a clock speed of 4.7 GHz, indicating its processing speed.

→ 2 GB 1333 MHz DDR3 SO-DIMM expandable up to 4 GB memory.

2 GB of DDR3 RAM operating at a speed of 1333 MHz. It also has the option to expand the RAM upto a maximum of 4 GB.

→ 32 KB L1 cache, 256 KB L2 cache and 4 MB L3 cache

L1 cache — 32 KB
 L2 — 256 KB
 L3 — 4 MB } high speed memory used to store frequently accessed data.

→ 1 TB 7200 RPM SATA Hard drive

Computer storage 1 TB. And hard drive operate 7200 revolutions per minute (RPM). It use Serial ATA (SATA) interface for data transfer.

→ (6) High speed USB 2.0 (2 side / 4 rear), (2) Side audio ports : headphone and microphone, (2) PS/2 ports 2

Serial 2

6 high speed USB 2.0 → USB ports for connecting external devices.

Audio jacks for headphone and microphone.

Used for connecting peripheral like keyboard, mouse

2 ports for serial communication.

→ 19" Inches LCD Monitor, 0.25 mm pitch, 16:9

1280X1024 at 80 MHz.

It's 19" led monitor with an anti glaze (AG) coating, a pixel pitch of 0.25 mm and resolution of 1280X1024 pixels at a refresh rate of 80 MHz.

→ 48X DVD RW optical

optical drive is capable of reading and writing DVDs at a speed of 48x.

→ 256 MB PCI express graphics card.

dedicated graphics card with 256 MB

of video memory using the PCI Express interface

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date :/...../.....

- 56 K data/fax Modem with ~~modem~~ 56 fax communication
- modem that supports data and fax communication at a speed of 56 kbps
- 64 bit PCI sound card
- sound card with 64 bit processing using the PCI interface
- Intel ~~®~~ 82578 DM, 10 M/100 M/1000 M Gigabit Ethernet
- Computer has an integrated Gigabit Ethernet Controller for high speed network connectivity.

Chapter - 03

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date: / /

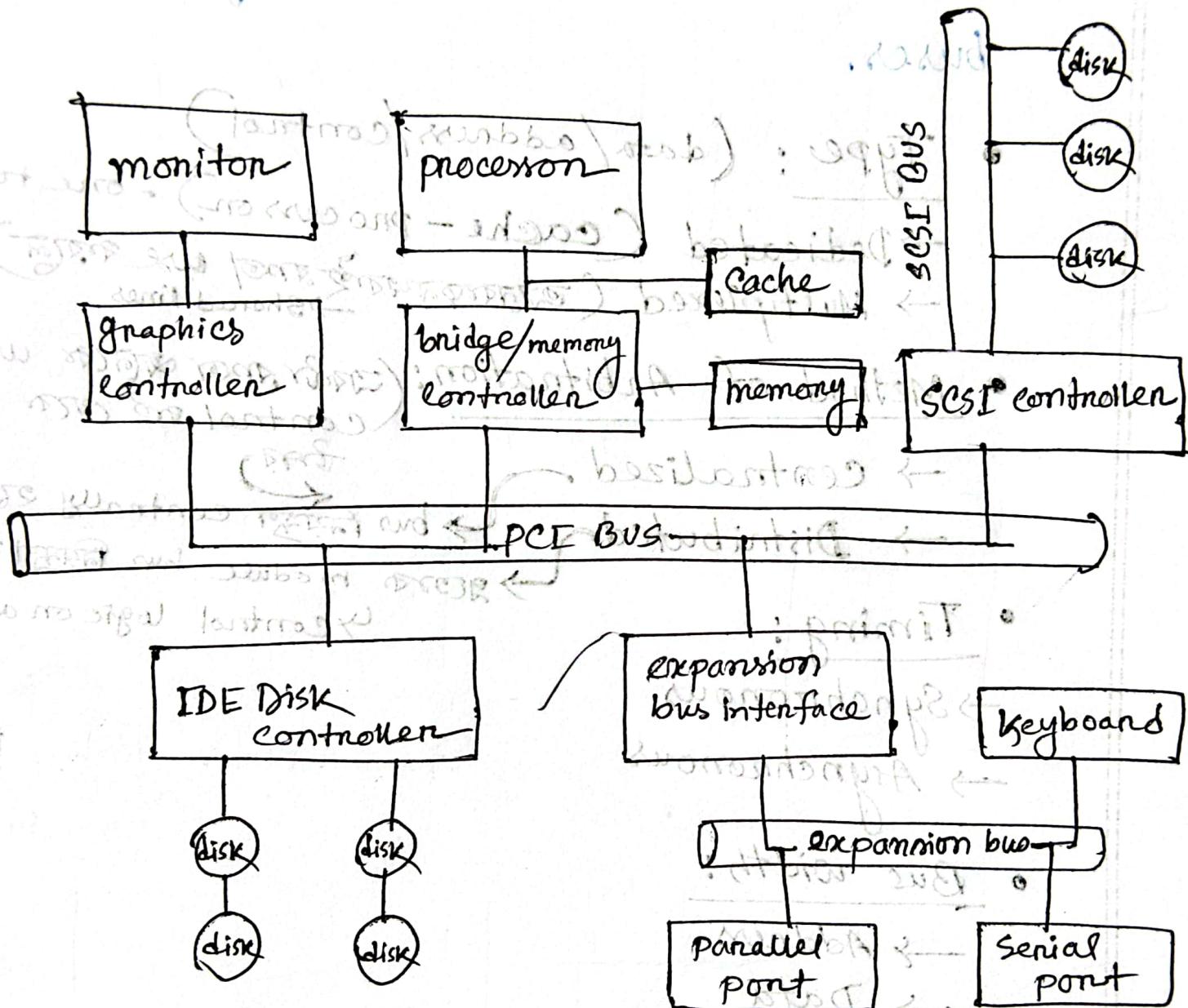
Computer function & Interconnection

① Give the basic elements to design different buses.

- Type: (data / address / control)
→ Dedicated (cache - processor) - one to one
→ Multiplexed (যোগাযোগ একই রূপে ব্যবহৃত) → Shared lines
- Method of Arbitration: (একটি রাজি আনকে আবরণে control করে রাখতে তাৰ সৈমান্য)
→ centralized
→ Distributed
 ↳ bus নিয়ন্ত্রণ কেন্দ্রীয় রূপে।
 ↳ প্রত্যেক module বুস নিয়ন্ত্রণ করতে।
 ↳ control logic on all modules.
- Timing:
→ Synchronous
→ Asynchronous
- Bus width:
→ Address
→ Data
- Data Transfer Type:
→ Read
→ write
→ read modify write
→ read after write
→ Block.

50 - 11. September
 Sat / Sun / Mon / Tue / Wed / Thu / Fri
 Date: / /

② Describe PCI Bus structure.



③ Define 'interrupt':

An interrupt is a mechanism in computer system that temporarily halt the normal execution of a program to handle a specific event or condition.

④ How do the multiple interrupts managed?

Two approach can be taken to dealing with multiple interrupts.

i) First is to disable interrupt while an interrupt being processed. Interrupts remain pending and are checked after first interrupt has been processed. Interrupt handled in sequence as they occurs.

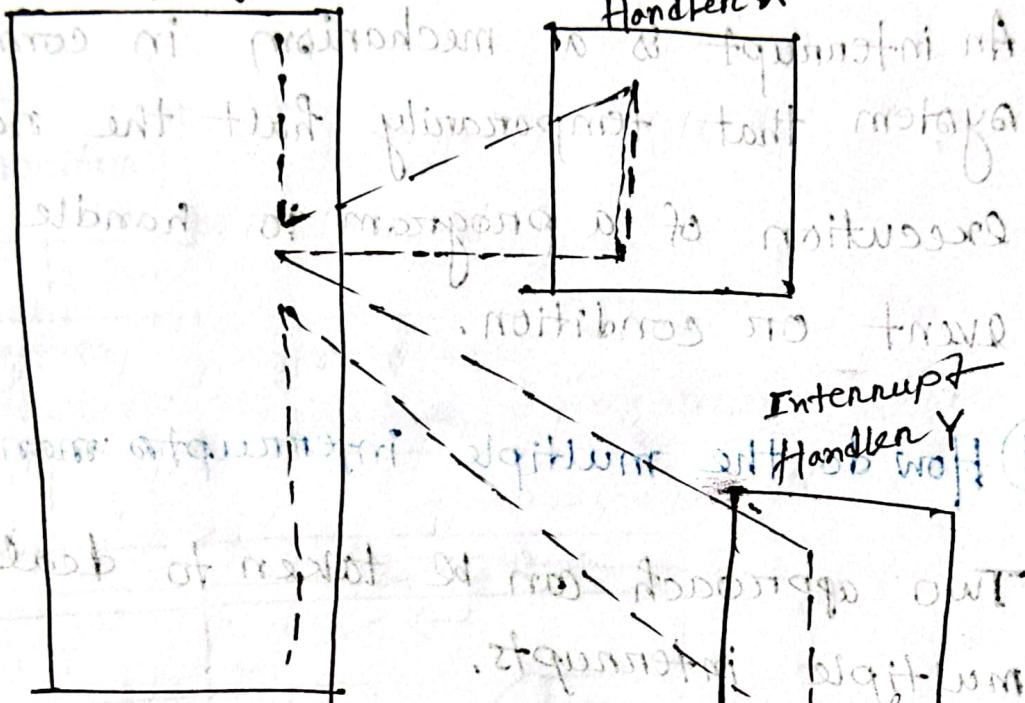
ii) Define priorities:

Low priority interrupts can be interrupted by higher priority interrupts. When higher priority interrupt has been processed, processon returns to previous interrupt.

Multiple interrupt - sequential

user program

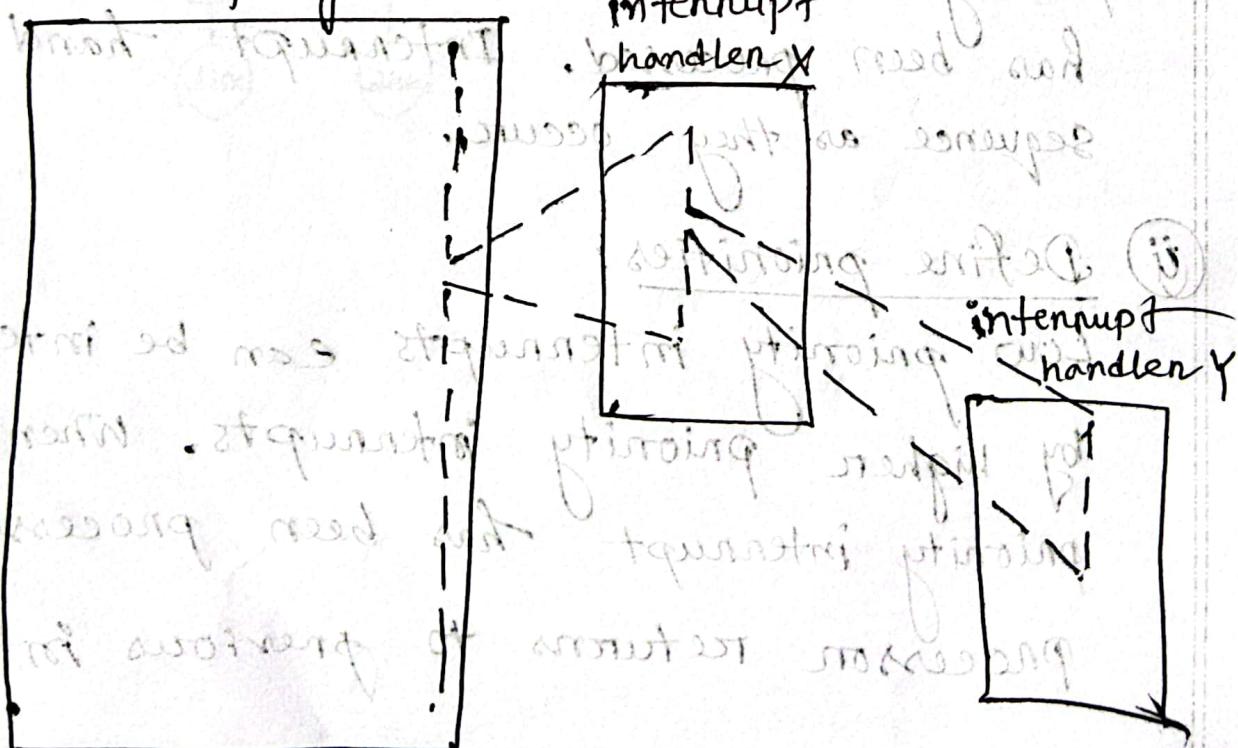
Interrupt
Handler X



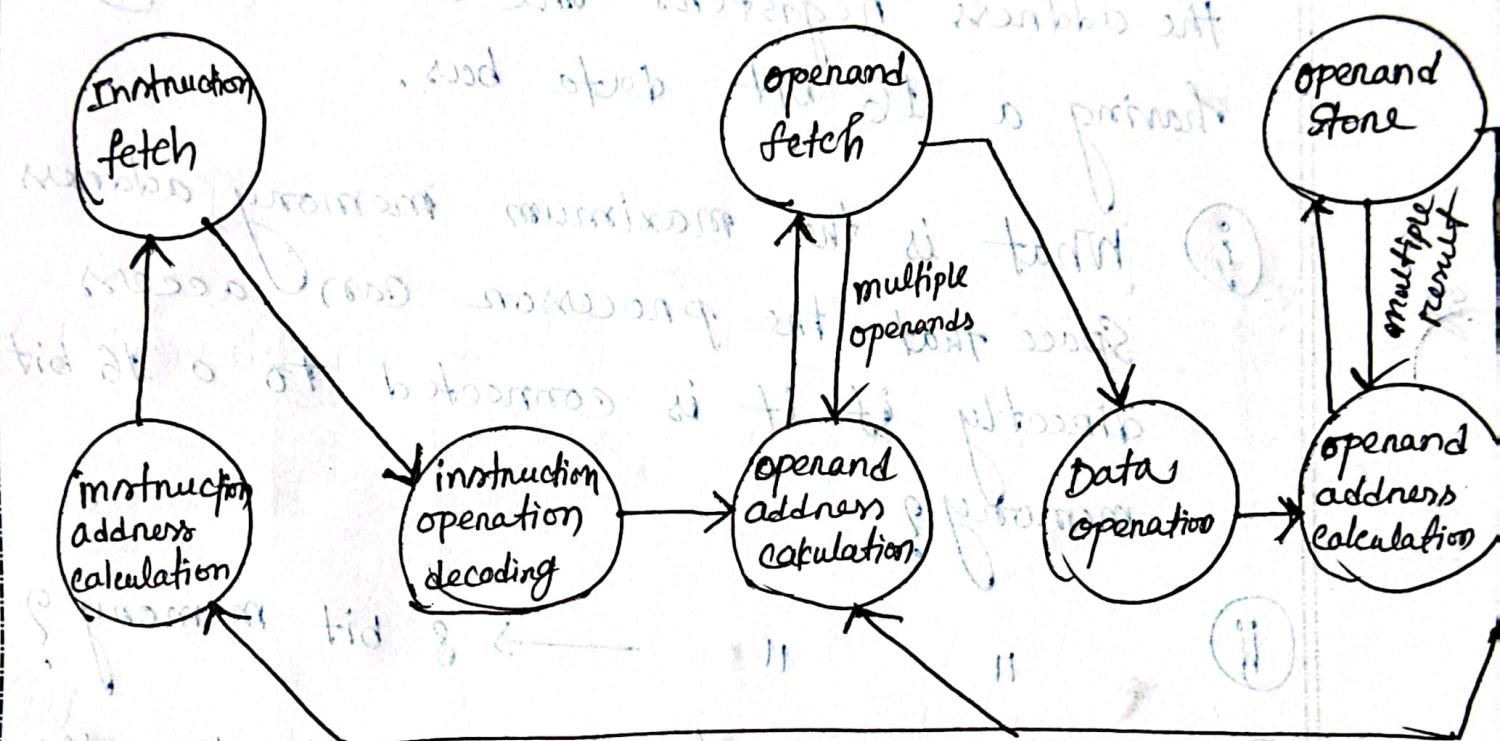
Multiple interrupt - Nested

user program

interrupt
handler X



5. List and briefly define the possible states that define an instruction execution.



instruction compute,
fetch next instruction

Return for
string or
vector data

3.4

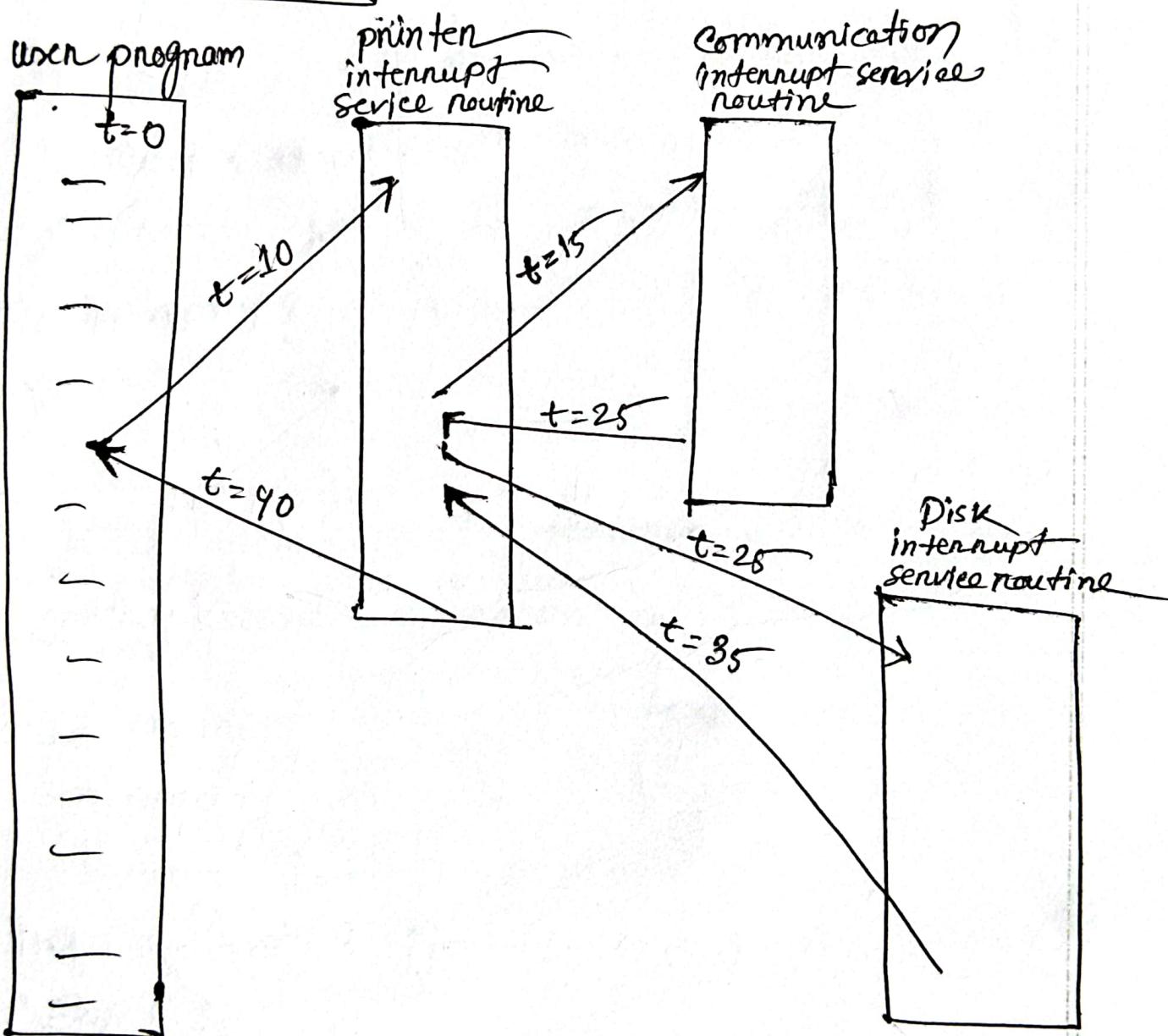
Q. Consider a hypothetical microprocessor generating a 16-bit address (for example, assume that the program counter and the address registers are 16 bits wide) and having a 16 bit data bus.

- i) What is the maximum memory address space that the processor can access directly if it is connected to a 16 bit memory?
- ii) 11 11 → 8 bit memory?
- iii) What architectural features will allow this microprocessor to access a separate I/O space?
- iv) If an input and an output instruction can specify an 8-bit I/O port number, how many 8-bit I/O ports can the chip support? How many 16-bit I/O ports? Explain.

7) Distinguish among sequential, nested, time sequence and priority interrupt.

→ 4 वर्तमान अवस्था
+

Time sequence :



Distinguish:

Diagram
shortly describe
(format)

- Transistor is a three terminal device.
- It has three terminals: Emitter, Base and Collector.
- It has two junctions: Emitter-base junction and collector-base junction.

• It has two stages of amplification.

• It has two types of biasing.

• It has two types of operating conditions.

• It has two types of output.

• It has two types of applications.

• It has two types of operating conditions.

• It has two types of applications.

• It has two types of operating conditions.

• It has two types of applications.

• It has two types of applications.

⑧ Why interconnection is needed in computer?

Or, Intercornection is necessary in computer system - briefly describe this statement.

Almost Same ← OR, "More dedicated interconnection produce faster computer" justify the statement.

The major computer system components (processor, main memory, I/O modules) need to be interconnected in order to exchange data and control signals.

Interconnection in computers refers to the various ways in which different components and devices within a computer system are linked together to communication and data exchange.

Interconnection is necessary for several reason:

- ① Communication between components
- ② Data Transfer

(ii) system integration

(iv) peripheral devices

(v) network communication

(vi) parallel processing

(vii) scalability

(viii) memory hierarchy

(ix) bus system

In summary, interconnection is a fundamental

aspect of computer architecture that enable

components to work together, facilitates data

transfer, support peripheral devices, allow

network communication and contributes to

the overall functionality, efficiency of

computer system.

Almost
Same
Topic

It's not just having more interconnections, but about having efficient, high speed, and purposeful interconnections designed to meet the specific needs of the system.

⑨ "If a large number of device are connected to the buses, performance will suffer" - explain this statement.

For this, there are two main reasons:

- i) In general the propagation delay
- ii) Propagation delay and Coordination:

More devices on a bus mean longer the bus length and increase propagation delays.

When control of the bus passes from one device to another frequently, these propagation delays can noticeably affect performance.

- iii) Bus Bottleneck and Data transfer Demand:

With many devices the bus may struggle

to handle growing data transfer

demands, becoming a bottleneck. Increasing

data rates and widening the bus can

help but may not keep up with the rapidly

rising data rates from devices like

graphics and video controllers.

10 Distinguish among local bus, System bus and expansion bus.

Local Bus	System Bus	Expansion Bus
Connects CPU to RAM and cache.	Connects major component within the motherboard.	Connects peripheral devices.
Inside, the CPU → location	location → on the motherboard.	location → on the motherboard or external connector
Data width → 64 or 128 bits	32 or 64 bits	8, 16 or 32 bits
Faster speed	Moderate speed	Slower compared to local and system
High performance computing system.	General purpose Computing system	connecting additional hardware components.
CPU, cache → component used	CPU, RAM, other major components	peripheral device such as Modem, Network card etc

(11)

A hypothetical machine has 15 instructions. 9/5/6

0001 = Load Ac from memory

0010 = Store Ac to memory

0101 = Add to AC from memory

0011 = Load AC from I/O

0111 = Store AC to I/O

0110 = Sub memory from AC

In this, case 12 bit address identifies a particular I/O device. Show the program execution (using 16 bit instruction format where prefix 4 bit for opcode) for the following program.

i) Load Ac from device 6

ii) Add contents of memory location 555

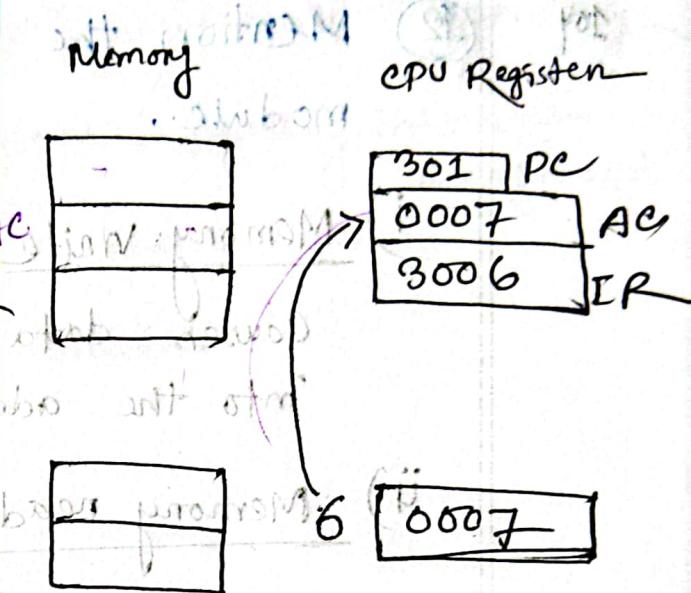
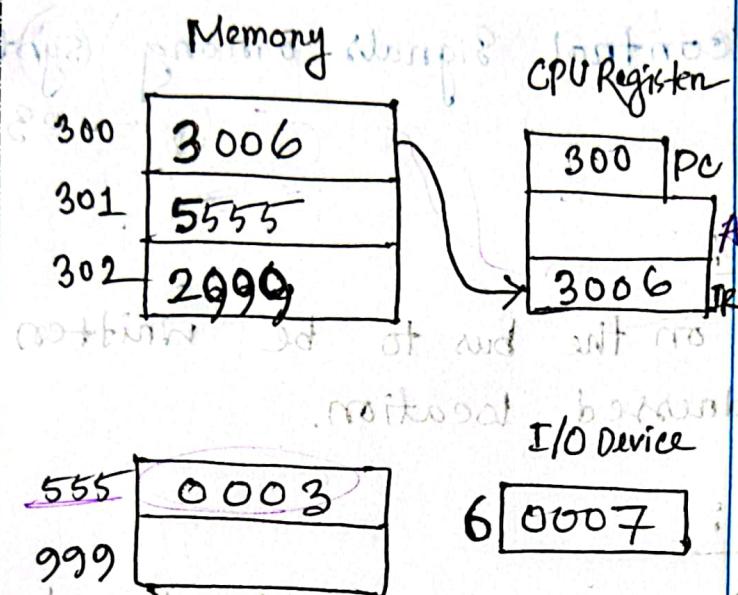
iii) Store Ac to memory location 999

Assume that the next instruction value retrieved from device 6 is 7 and that location 555 contains a value of 3.

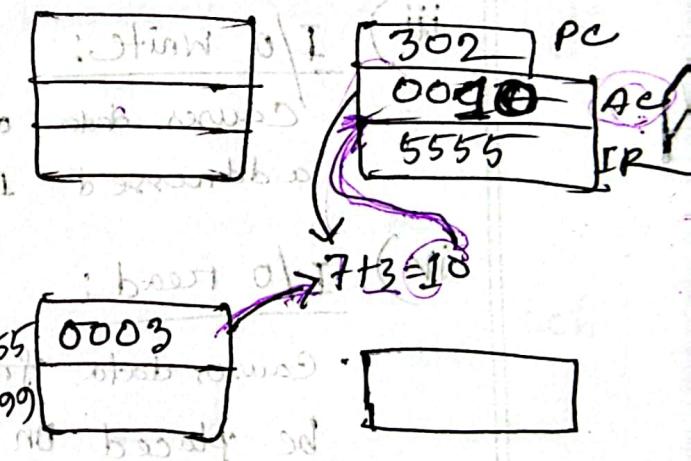
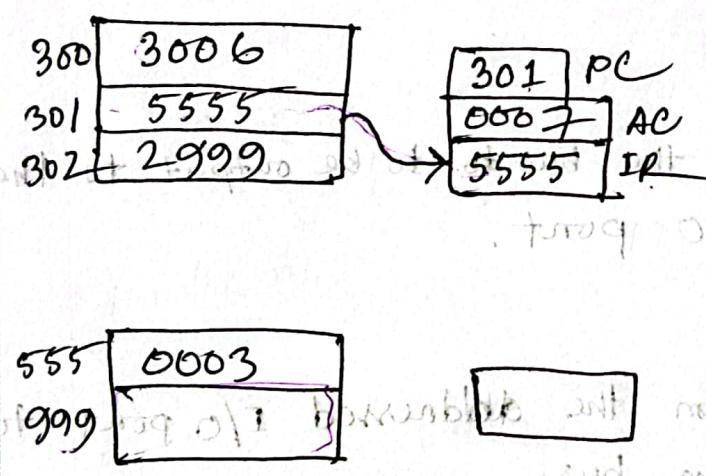
Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

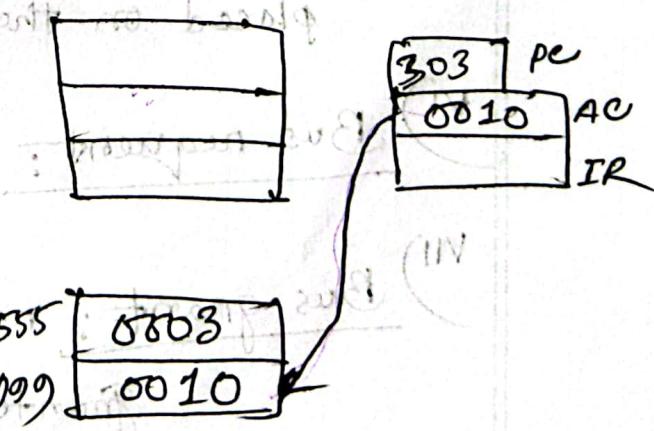
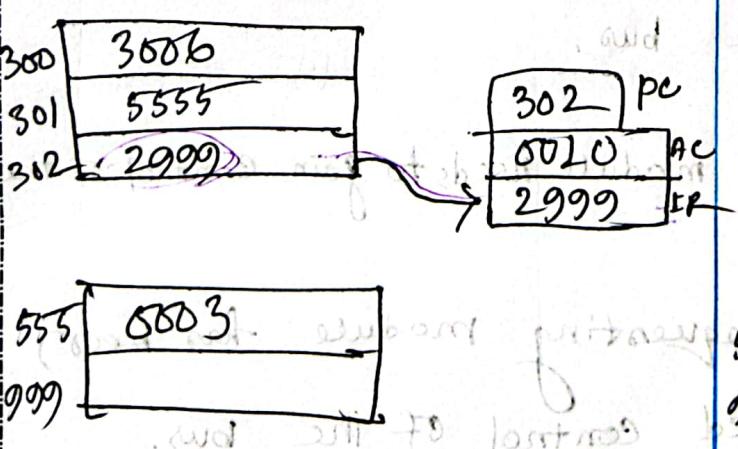
i)



ii)



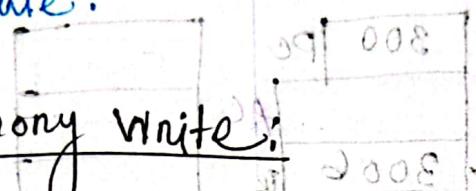
iii)



104

12) Mention the control signals among system modules.

i) Memory Write:

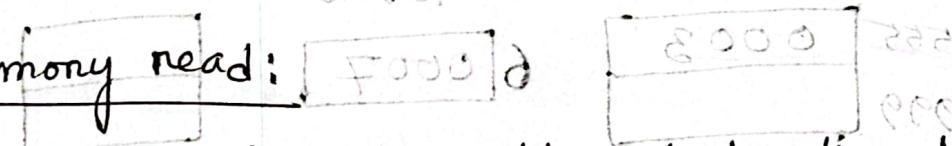


0000 100
0001 101
0010 110
0011 111
0100 000
0101 001
0110 010
0111 011

0000 000
0001 001
0010 010
0011 011

Causes data on the bus to be written into the addressed location.

ii) Memory read:



0000 100
0001 101
0010 110
0011 111
0100 000
0101 001
0110 010
0111 011

0000 000
0001 001
0010 010
0011 011

Causes data from the addressed location to be placed on the bus.

iii) I/O write:

Causes data on the bus to be output to the addressed I/O port.

iv) I/O read:

Causes data from the addressed I/O port to be placed on the bus.

v) Transfer ACK:

Indicate data have been accepted from or placed on the bus.

vi) Bus request:

Module needs to gain control of the bus.

vii) Bus grant:

Requesting module has been granted control of the bus.

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

- VIII) Interrupt request: Indicate interrupt is pending
- IX) Interrupt ACK: Acknowledgement pending interrupt recognized.
- X) Clock: used to synchronize operations.
- XI) Reset: Initializes all modules.

Q13 What is computer system?

A computer system consists of a processor, memory, I/O, and the interconnections among these major components.

We can describe a computer system by

- i) describing the external behaviour of each component.
- ii) describing the interconnection structure and the controls required to manage the use of the interconnection structure.

14 Distinguish between Synchronous and asynchronous timing operation. (XII)

Synchronous Timing Operation: (IX)

Synchronous timing required a global clock

signal to synchronize the operations of various components in a system.

The synchronous ensures that components perform operations at specific clock edges, leading to a generally higher performance compared to asynchronous systems.

Asynchronous timing operation: (i)

Asynchronous timing operates without a global clock, allowing components to trigger events independently.

While prioritizing flexibility, asynchronous timing may result in various performance due to the lack of global synchronization. Components operate independently, introducing timing variations.

process structure & Function

① what general roles are performed by processor?

Fetch instruction: The processor reads an instruction from memory.

Interpret instruction: Instruction is decoded to determine what action is required.

Fetch data: Instruction may require reading data from memory or I/O.

Process data: Instruction may require performing some arithmetic or logical operation on data.

Write data: Results of an execution may require writing data to memory or an I/O module.

② What general roles are performed by processor registers?

User Visible registers: (আমাদের প্রযোজনীয় ব্যবহারে programming মাধ্যমে)

Enable the machine or assembly language programmer to minimize main memory references by optimizing use of registers.

→ A user visible register are the registers visible to programmers.

Ex: • general purpose register (8-32)

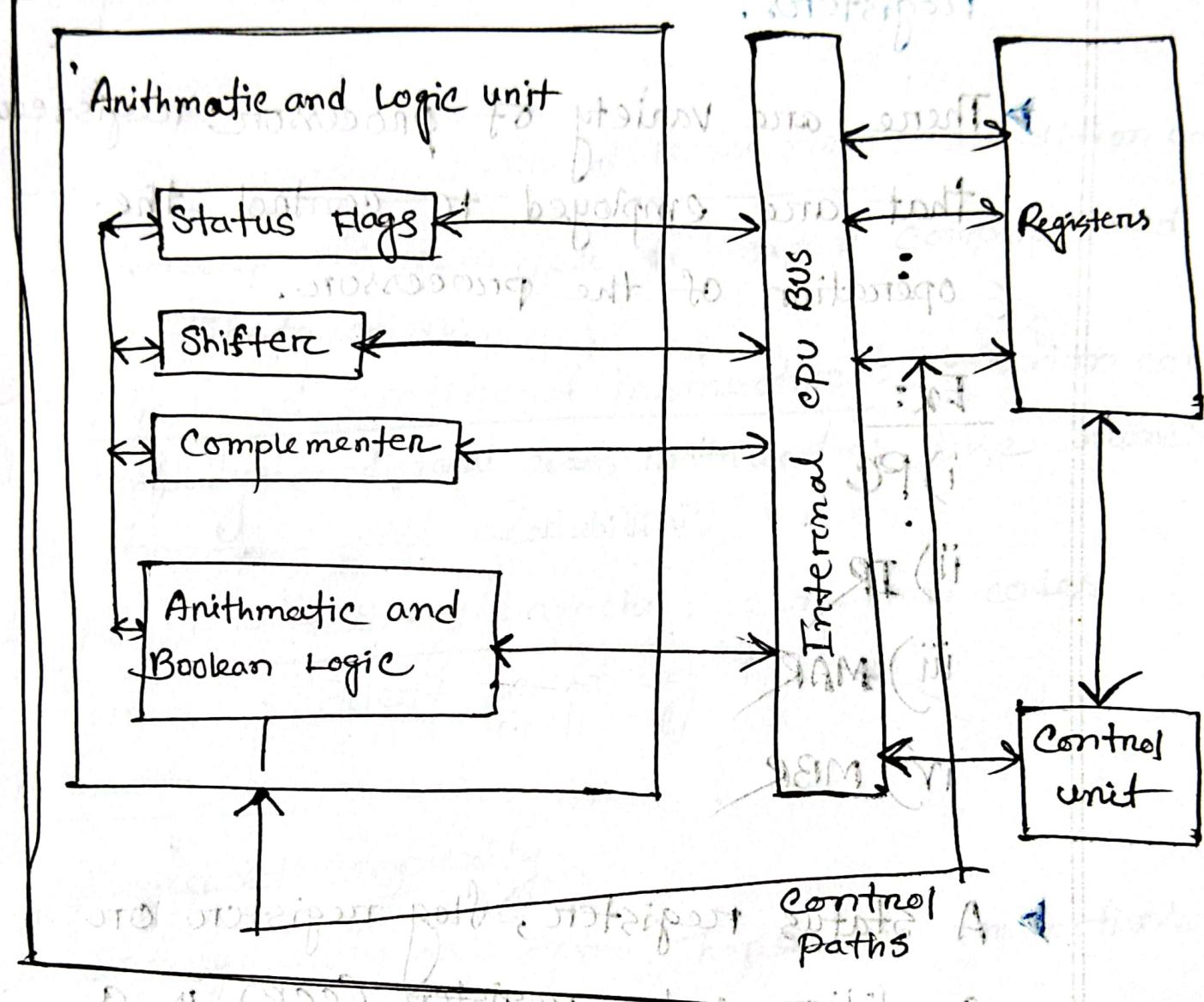
- Data register → (Accumulator)
- Address register → (segment)
- Flag register (condition codes)

Control and Status register: (আমাদের প্রযোজনীয় ব্যবহারে computer নিয়ের প্রযোজনীয় ব্যবহার)

Used by the control unit to control the GDT register operation of the processor and by privileged operating system programs to control the execution of programs.

(3)

Draw the Internal Structure of CPU.



Day 1 (iv)

Assignment 2 (iv)

Ques (i) Ans

Ques (ii)

Ques (iii)

Ques (iv)

④ Write down the control, and status registers.

► There are variety of processor registers that are employed to control the operation of the processor.

Ex:

i) PC

ii) IR

iii) MAR

iv) MBR

► A status register, flag register or condition code register (CCR) is a collection of status flag bits for a processor.

Ex:

i) sign

ii) zero

iii) Carry

iv) Overflow

v) Interrupt Enable/Disable

vi) Equal

vii) Supervision

155 page ← 5 ⑤ Advantages and disadvantages of Condition codes → (Flag Register) - 02

Advantages:

1. Instruction overhead reduction: condition codes decrease the need for ~~dedicated~~ compare and test instructions.
2. Simplified conditional instructions: Condition codes simplify conditional ~~code~~ instructions like branches, improving code readability.
3. Facilitate multiway branches: condition codes enable efficient multiway branches.

Disadvantages:

1. Increased complexity.
2. Irregularity and extra hardware connections.
3. Need for special instructions.
4. Pipelined Implementation Challenges.

What are the functions of condition codes.

Condition codes are bits, set by the processor hardware as the result of operations.

For example an arithmetic operation may produce a positive, negative, zero or overflow result. In addition to the result itself being stored in a register or memory, a condition code is also set. The code may subsequently be tested as part of a conditional branch operation.

Q61 → (7) Describe the data flow of fetch and indirect cycle.

During the fetch cycle an instruction is read from memory. The PC contains the address of the next instruction to be fetched. This address is moved to the MAR and placed on address bus. The control unit requests a memory read, the memory result is placed on the data bus and copied into the MBR and then moved to IR. Meanwhile the PC is incremented by 1, preparatory for the next fetch.

Once 'fetch' is over, control unit examines the contents of IR to determine if it contains an operant specifier using indirect addressing. If so, an indirect cycle is performed. The right most N bits of the MBR, which contain the address reference, are transferred to the MAR. Then the control unit requests a memory read, to get the desired address of the operand into the MBR.

(8)

What is pipelining?

Pipelining is a technique in computer architecture that involves organizing the execution of multiple instruction simultaneously.

(9)

What is instruction pipelining?

Q. নির্মাণ যুক্তি

RAM

10) Describe the Six stage CPU Instruction Pipelining with timing diagram. — 03

Six stage CPU instructions Contain:

- (I) Fetch instruction (FI)
- (II) Decode instruction (DI)
- (III) calculate operands (CO)
- (IV) Fetch operands (FO)
- (V) Execute Instruction (EI)
- (VI) Write operand (WO)

every stages take more nearly equal duration.

In this six stage CPU pipelining process
When Decode instruction running on Instruction 1,
Instruction 2 start fetching its instruction.

Let's see the timing diagram for six stage CPU instruction pipeline operation.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instruction 1	FI	DI	CO	FO	EI	WO								
2		FI	DI	CO	FO	EI	WO							
3			FI	DI	CO	FO	EI	WO						
4				FF	DI	CO	FO	EI	WO					
5					FF	DI	CO	FO	EI	WO				
6						FF	DI	CO	FO	EI	WO			
7							FI	DI	CO	FO	EI	WO		
8								FF	DI	CO	FO	EI	WO	
9									FI	DI	CO	FO	EI	WO

(On) breakups strip (IV)

This six stage pipeline can reduce the execution time for 9 instructions from 54 time units to 14 time units.

$9+6-1$

$14 = 14$

54 19

16 14

11 Pipeline processor with K stages is $\frac{1}{K}$ time faster than non pipeline processor. Justify this

→ Timing diagram (10 $\frac{2}{3}$) draw
 Let $T_{K,n}$ be the total time required for a pipeline with K stages to execute n instruction. Then

$$T_{K,n} = [K + (n-1)] \tau$$

From timing diagram,

$$14 = [6 + (9-1)]$$

Now, consider a processor with equivalent functions but no pipeline, and instruction cycle time is $K\tau$. The speedup factor for the instruction pipeline compared to execution without the pipeline is define as

$$S_K = \frac{T_{1,n}}{T_{K,n}} = \frac{nK\tau}{[K + (n-1)]\tau}$$

$$\frac{nK}{K + (n-1)}$$

As might be expected at the limit ($n \rightarrow \infty$) we have a K fold speedup.
 So, pipeline processor with K stages is $\frac{1}{K}$ time faster than non pipeline processor.

12 Pipeline process is inefficient for branch instruction. Explain the mechanism to deal the brancher in pipelining. — 03

The effect of a conditional branch on Instruction pipeline operation.

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	FI	DI	CO	FO	EI	WO								
2		FI	DI	CO	FO	EI	WO							
3			FI	DI	CO	FO	EI	WO						
4				FI	DI	CO	FO							
5					FI	DI	CO							
6						FI	DI							
7							FI							
15								FI	DI	CO	FO	EI	NO	
16									FI	DI	CO	FO	EI	WO

Here Instruction 3 is a conditional branch to instruction 15. until instruction 3 execution there is no way knowing which instruction will come next.

The pipeline in this example simply loads the next instruction in sequence (instruction 4) and proceeds. This is not determined until end of time unit 7. So, instruction 4, 5, 6, 7 are not useful and the pipeline must be cleared, them.

So, we say pipeline processor is inefficient for branch instruction.

Mechanism to deal the branches in pipelining:

(i) Multiple Streams \rightarrow have two pipelines

(ii) prefetch Branch Target along with RA

(iii) Loop buffer \rightarrow check buffer before fetching from memory

(iv) Delayed branching \rightarrow Don't Jump until reading instructions

(v) Branch prediction

13 Explain performance and performance

penalty of instruction pipelining.

(10) নং প্রশ্নের Timing Diagram draw করবে।

→ In this figure, branch is not taken and we get the full performance.

(11) নং প্রশ্নের Condition Branch র টাইম ডায়াগ্রাম করা হবে।

→ In this figure, branch is taken. This is not determined until the end of time unit 7. At this point, the pipeline must be cleaned of instructions that are not useful. During time unit 8, instruction 15 enters the pipeline. No instructions complete during time units 9 through 12 (instructions 4, 5, 6, 7). This is the performance penalty incurred because we could not anticipate the branch.

14

Assume a pipeline with 4 stages :

Fetch Instruction (FI) ✓

— 04

Decode " ~~DE~~

Calculate address (DA) ✓

Fetch operand (FO) ✓

Execute (EX) ✓

Draw a timing diagram for instruction pipeline operation for a sequence of

9 instruction, in which the fifth instruction is a branch that is taken and in which there is no data dependencies.

	1	2	3	4	5	6	7	8	9	10	11	12
I1	FI	DA	FO	EX								
I2		FI	DA	FO	EX							
I3			FI	DA	FO	EX						
I4				FI	DA	FO	EX					
I5					FI	DA	FO	EX				
I6						FI	DA	FO				
I7							FI	DA				
I8								FI				
I9									FI	DA	FO	EX

Chap-09Unsigned binary multiplication:

$$11 \rightarrow 1011 \text{ — multiplier}$$

$$13 \rightarrow 1101 \text{ — multiplier}$$

$$\underline{143}$$

C	A	S	M	
0	0000	1101	1011	Initial values
0	1011	1101	1011	Add (A+M)
0	0101	1110	1011	Shift
0	0010	1111	1011	Shift
0	1101	1111	1011	Add (A+M)
0	0110	1111	1011	Shift
1	0001	1111	10110	Add (A+M)
0	1000	1111	1011	Shift

(143)

- Right shift (Logical) $\rightarrow C, A, S$

- $S_0 = 1$ शर्त (A+M) ADD करें

Otherwise, only shift.

M+A

 $\rightarrow A$

FFFF

0000

1010

128
15
143

B7D2

FFFF

0000

0300

B2D2

FFFF

1010

1011

A3

FFFF

1111

1110

A3

• 32-bit

C, A, B₋₁

initial value always zero

Sat / Sun / Mon / Tue / Wed / Thu / Fri



Date : / /

Booth Algorithm

Signed - Unsigned multiplication

Two's complement multiplication

$$(7 \times 3) \rightarrow 21$$

multiplicand
multiplicand

$$\begin{array}{r}
 \begin{array}{r} A \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{r} B \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{r} B_{-1} \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{r} M \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{r} Product \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \\
 \hline
 \end{array}$$

Multiplicand

A	B	B ₋₁	M	Product	Initial values.
0000	0011	0	0111	0000	
1001	0011	0	0111	0000	A \leftarrow A - M
1100	1001	1	0111	0000	Shift
1110	0100	1	0111	0000	Shift
0101	0100	1	0111	0000	A \leftarrow A + M
0010	1010	0	0111	0000	Shift
0001	0101	0	0111	0000	Shift

21

Sign bit କାହାରେ ବିକାଶ ହେବାରେ

- Right shift (arithmetic) $\rightarrow A, S, S_{-1}$

δ_0	δ_{-1}	
1	0	Subtraction (A-M)
0	1	Addition (A+M)
0	0	
1	1	only Right Shift

प्रस्तुत
प्राचीन

1101 = M

biting
- ~~poisonous~~
+ B. B. B.

$\frac{1}{4} = 0.25$

→ line
points

0111 0000 1011

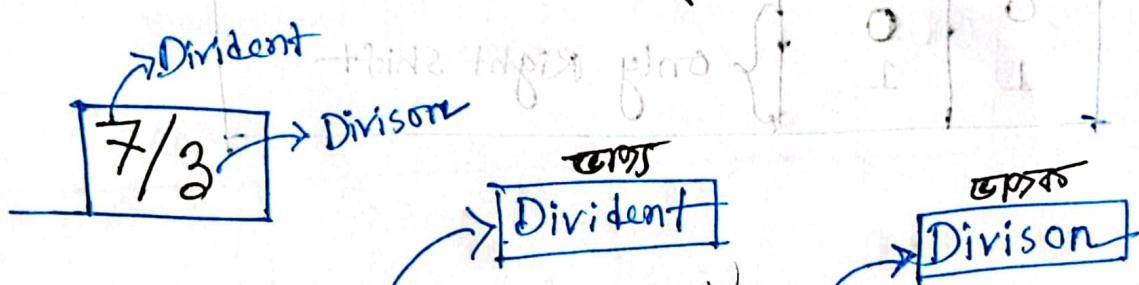
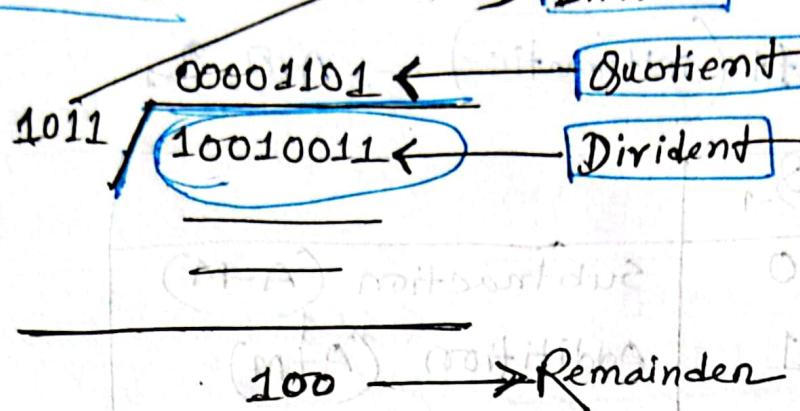
00 R 1000.

DOLE 1000

0000
0000

OBELISK

Division:



A	Q	M
0000	0111	$M = 0011$ initial values
0000	1110	Shift \rightarrow left (logical)
1101	.	Subtract
0000	1110	Restore
0001	1100	Shift
1110	.	Subtract
0001	1100	Restore
0011	1000	Shift
0000	.	Subtract
0000	1001	Set $Q_0 = 1$
0001	0010	Shift
1110	.	Subtract
0001	0010	Restore

Reminder = 1 Quotient = 2

- Divident ଭିଲାମ୍ବର ରୂପ

ଆଶଳେ A ଏଇ initial value $\rightarrow 1111$

- Divident ଭିଲାମ୍ବର ରୂପ

A ଏଇ initial value $\rightarrow 0000$

- Shift Left (logical) $\rightarrow A, Q$

~~→ M ଏଇ 2's Complement କାହିଁ କ୍ଷେତ୍ର ଏବଂ~~

\rightarrow A ଏଇ sign bit } Same ରୂପ \rightarrow Subtract (A-M)
 M ଏଇ sign bit } କିନ୍ତୁ ରୂପ \rightarrow Add (A+M)

⊗ A ଏଇ Value shift ଏଇ ଅନ୍ତର୍ବର୍ତ୍ତର ବିନ୍ଦୁ,

\rightarrow Shift value ଏଇ sign } • Same ରୂପ \rightarrow operation successful
 Subtract n " sign } • କିନ୍ତୁ ରୂପ \rightarrow Restone.

\rightarrow operation ଯାଦି

Successful ରୂପ | (~~ଯାଦି କିମ୍ବା~~)

⊗ \rightarrow 0,0 ରୂପ Success ରୂପ, $Q_0 = 1$ କଷଟକୀୟ

\rightarrow 1,1 ରୂପ Success ରୂପ, $Q_0 = 0$ କଷଟକୀୟ।

\rightarrow Divisor, Divident opposite sign ରୂପ

କିମ୍ବା 2's Complement ବାବୁ ଯେବେଳେ ପାଇଁବାରୀ କାହାରେ,

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

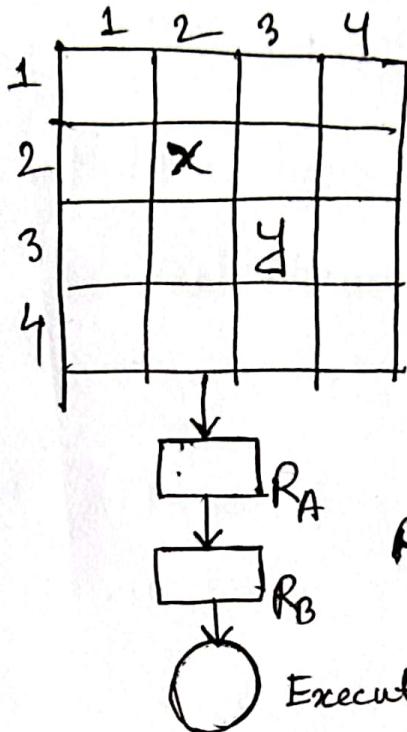
CISC & RISC

CISC

- I) Complex Instruction set computer.
- II) Large number of instruction.
- III) Variable length instruction format.
- IV) Large no. of addressing modes.
- V) Cost is high.
- VI) More power.
- VII) Several cycle instructions.
- VIII) Manipulation directly in memory.
- IX) Microprogrammed control unit
- X) Mainframes, Motorola 6800, Intel 8080, Intel 80486, Pentium, Core-i, AMD

RISC

- I) Reduced instruction set computer.
- II) Less no. of instructions.
- III) Fixed length instruction format.
- IV) Few no. of addressing modes.
- V) less cost.
- VI) Less powerful
- VII) Single cycle instructions.
- VIII) manipulation only in Register.
- IX) Hardwired control unit.
- X) MIPS, ARM, SPARC

CISCRISC

Register set

Execution unit

MULT 2:2, 3:3

$$x = x * y$$

LOAD A, 2:2

LOAD B, 3:3

PROD A, B

STOR 2:2, A

Previous question

$$x = x * y$$

1. Give the characteristics of RISC and CISC.
2. Difference between CISC and RISC.
3. ~~Give~~ Give some example of processors of each category.
4. Explain RISC architecture.

Micro-Operation :

- Sheet
1. Illustrate and list the micro-operations to perform addition between two operands from memory. -02/02
2. Analyze the relationship between instruction and micro-operation.



Operations executed on data stored in registers are known as microoperation.

→ Shift, Clear, count, Count, Load, Mul

Types

- (i) Data Transfer
- (ii) Arithmetic
- (iii) Logical
- (iv) Shift

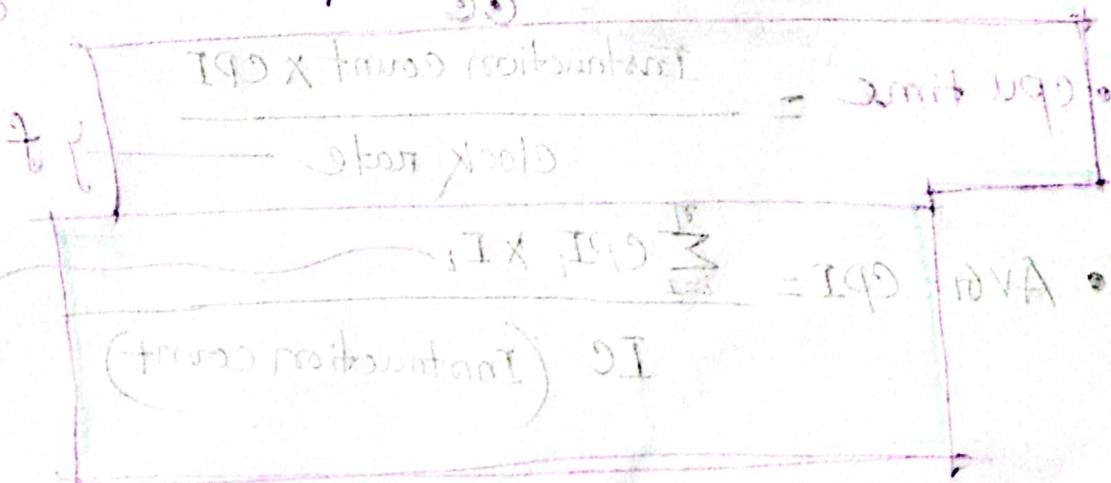
Sat / Sun / Mon / Tue / Wed / Thu / Fri

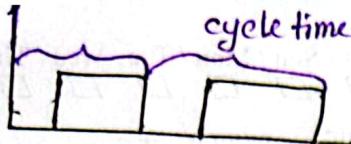
 Date: / /

Relationship between instruction and microoperations

1. When instruction fetch and decoded by the CPU, its broken down into a sequence of micro-operations.
2. Control unit generates control signal based on decode instruction, which initiate the necessary microoperations to execute the instruction.
3. Each instruction implemented by a sequence of micro-instructions.
4. Multiple micro-operations within an instruction may be executed in parallel.

→ Instructions are composed of sequence of micro operations.





Sat / Sun / Mon / Tue / Wed / Thu / Fri

 Date : / /

performance

measurements

- $CPU \text{ time} = CC \times CT = IC \times CPI \times CT$

$$= \frac{CC}{f}$$

$CC = \text{cycle count}$

$CT = \text{cycle time}$

$$CT = \frac{1}{f}$$

$f \rightarrow \text{clock frequency}$

$CPI = \text{Cycle per instruction}$

$IC = \text{Instruction count}$

- $CPI = \frac{\text{cpu clock cycle for the program}}{\text{Instruction count}}$

- $CPU \text{ time} = \frac{\text{Instruction count} \times CPI \times \text{Clock cycle time}}{f}$

- $cpu \text{ time} = \frac{\text{Instruction count} \times CPI}{f}$

- $AVG \text{ CPI} = \frac{\sum_{i=1}^n CPI_i \times I_i}{IC \text{ (Instruction count)}}$

→ No. of
cycle per
second

► MIPS (Million Instruction per second)

$$\bullet \text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

$$\bullet \text{MFLOPS} = \frac{\text{NO. of floating point operation in a program}}{\text{Execution time} \times 10^6} = \frac{\text{IC}}{\text{IC} \times \text{CPI}} \times 10^6 = \frac{\text{f}}{\text{f}} = \frac{\text{f}}{\text{IC} \times \text{CPI} \times 10^6}$$

$$\bullet \text{Arithmatic mean} = \frac{1}{n} \sum_{i=1}^n \text{Execution time}_i$$

$$\bullet \text{Geometric mean} = \sqrt[n]{\prod_{i=1}^n \text{Execution time}_i}$$

$$\bullet \text{Harmonic mean} = \frac{n}{\sum_{i=1}^n \frac{1}{\text{Execution time}_i}}$$

$$\text{CPI} = \frac{\text{CCT} + \text{ET}}{\text{f}} = \frac{\text{CCT}}{\text{f}} + \frac{\text{ET}}{\text{f}}$$

$$\text{CPI} = \frac{\sum \text{CPI}_i \times I_i}{\text{IC}}$$

$$\text{MIPS} = \frac{\text{f}}{\text{CPI} \times 10^6} = \frac{\text{f}}{\text{CPI} \times 10^6} = \frac{\text{f}}{\text{CPI} \times 10^6}$$

$$\text{MIPS} = \frac{\text{f}}{\text{CPI} \times 10^6} = \frac{\text{f}}{\text{CPI} \times 10^6}$$

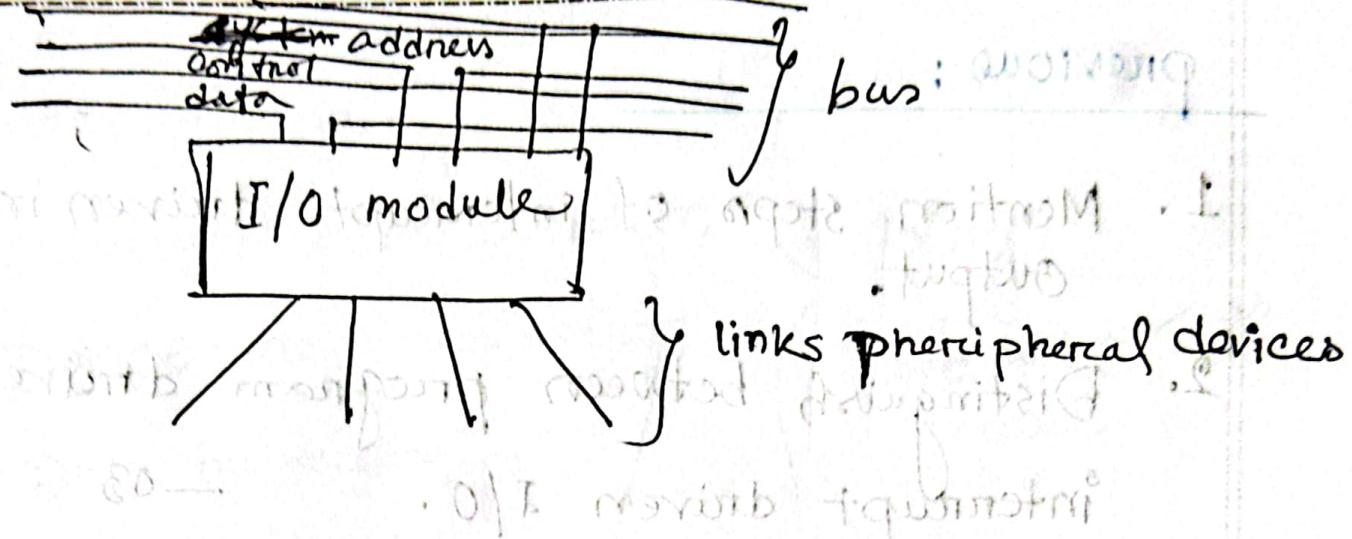
Chapter - 07

Input - Output

Atiq
Sir

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /



→ External devices

→ Keyboard / monitor

→ Block diagram of an I/O module

→ Programmed I/O

→ I/O command

→ I/O instructions

→ Addressing format

→ Addressing format
memory addressing

previous :

1. Mention steps of interrupt driven input-output. — 03
2. Distinguish between program driven and interrupt driven I/O. — 03
3. Why I/O module is used in Computer system? Classify the external devices.
4. Describe the function of input-output module.
- 5.

↳ Human Readable
Machine Readable
Communication

4

Function of an I/O module:

- i) Control and timing
- ii) Processor communication
- iii) Device communication
- iv) Data buffering
- v) Error detection

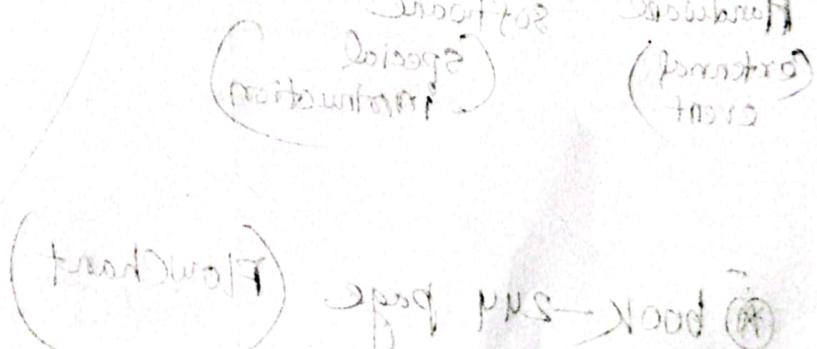
Why

I/O module is

used

Three techniques:

- details
- I) programmed I/O
 - II) Interrupt driven I/O
 - III) Direct Memory Access (DMA)



Programmed I/O Interrupt

3 techniques of I/O

- programmed I/O
 - interrupt driven I/O
 - DMA
- To exchange information between user and CPU.

user

CPU

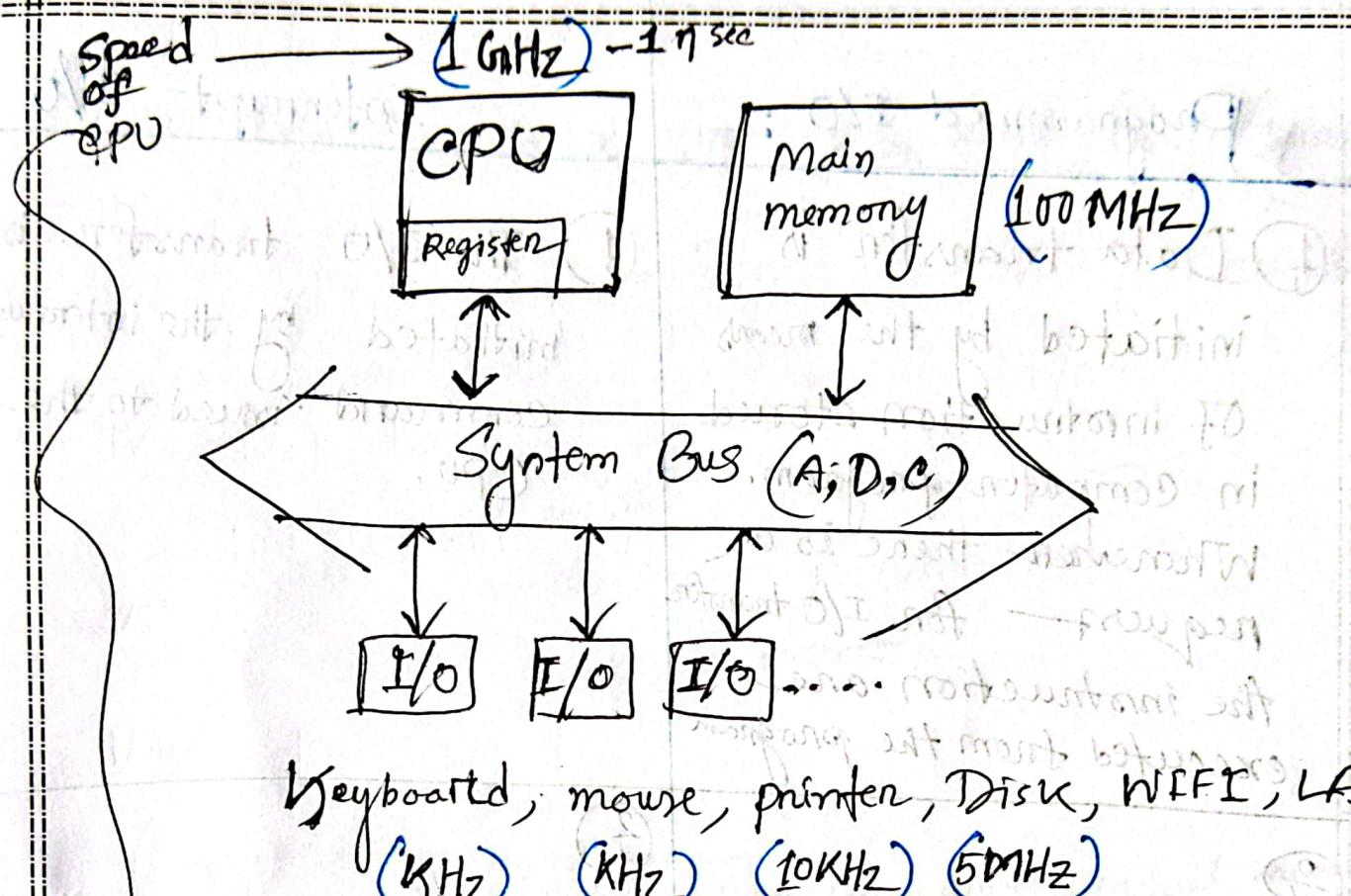
I/O

memory

Interrupt : Suspension of process

Hardware (external event) software (special instruction)

book - 244 page (Flowchart)



④ Keyboard is faster than CPU 1 billion speed higher.

I/O transfer modes

- Programmed I/O
- Interrupt Driven I/O
- DMA

Sat / Sun / Mon / Tue / Wed / Thu / Fri

□ □ □ □ □ □ □

Date :/...../.....

Programmed I/O

- ① Data transfer is initiated by the means of instruction stored in computer program. Whenever there is a request for I/O transfer the instruction are executed from the program.

②

Interrupt I/O

- ① The I/O transfer is initiated by the interrupt command issued to the CPU.



②

informant of I

I/O being respond

I/O moving, after wait

Arms

Virtual memory

Sat / Sun / Mon / Tue / Wed / Thu / Fri
□ □ □ □ □ □ □

Q Hard disk मुद्रित नो. Support नियंत्रित करता है। (Virtual memory)

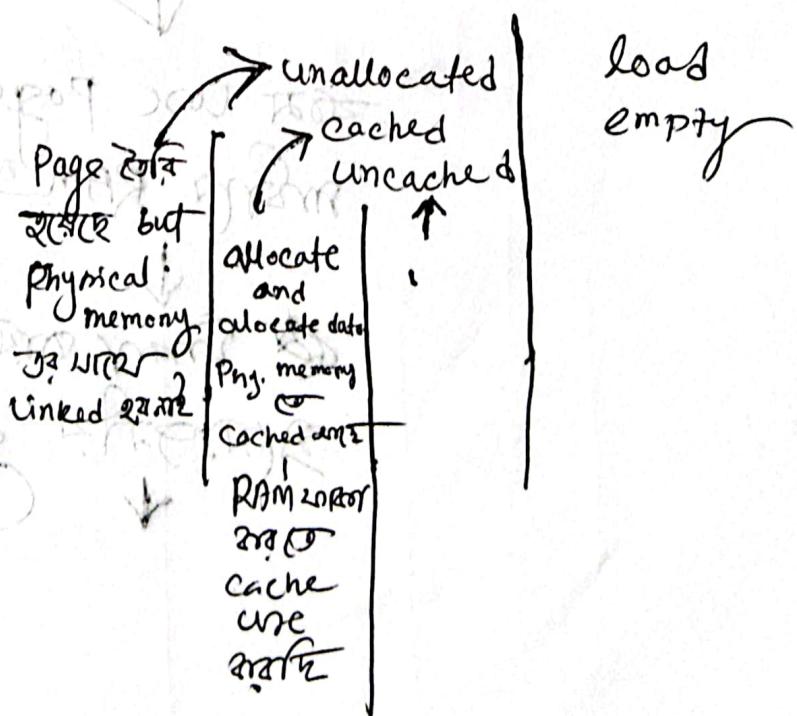
৪) CPU দ্বারা যে চিকিৎসা নির্বাচন করে Virtual address.

→ ~~বিভিন্ন~~ portion এবং এটা
dedicatedly RAM এর মাঝে
link up হবে।

physical (ram)

ପ୍ରାଣ / ଜୀବଶିତ୍ରର shift ଓ ଏବଂ Virtual memory ଟା

- CPU 64 bit ରୀତି
ଏହା ପାଇଁ page ନା 64 bit ହାବାରୁ (Page size 8 byte)



Sat / Sun / Mon / Tue / Wed / Thu / Fri



Date:/...../.....

➤ CPU \rightarrow RAM ~~for most data~~ for

RAM ~~for~~ data for

Request miss

RAM ~~for~~ document handle disk

for CPU, Support Apps

use Page ~~for~~ Select

page



use Page handle disk

from RAM \rightarrow memory management

for memory management

data ~~for~~ for

\downarrow (disk \rightarrow DRAM)

Chap - 06

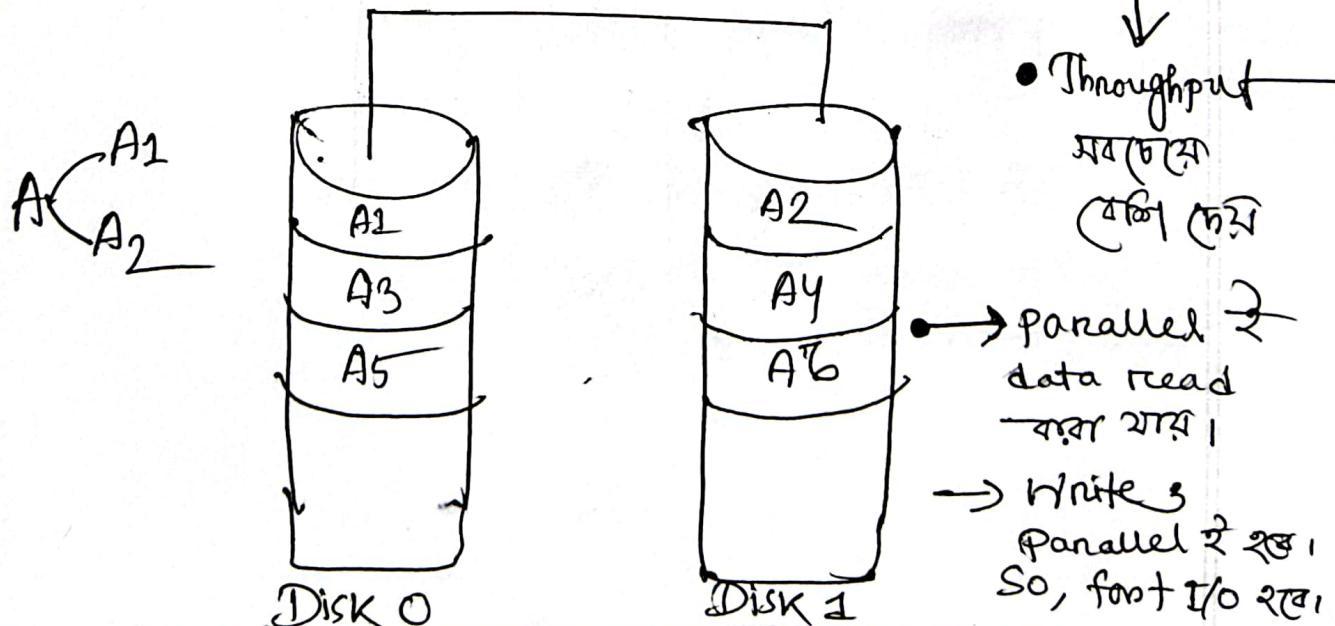
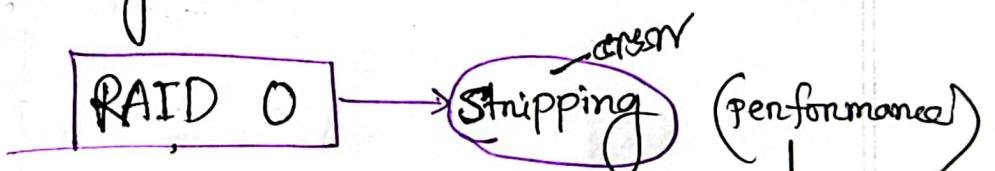
External Memory

- previous
- ① Describe RAID working mechanism. - 03
 - ② How are data read from magnetic disk. - 03
 - ③ Mention the key feature of RAID 0 - RAID 6 - 03

RAID

Redundant Array of Independent Disks
 Redundant Array of Inexpensive Disks
 for Storing data

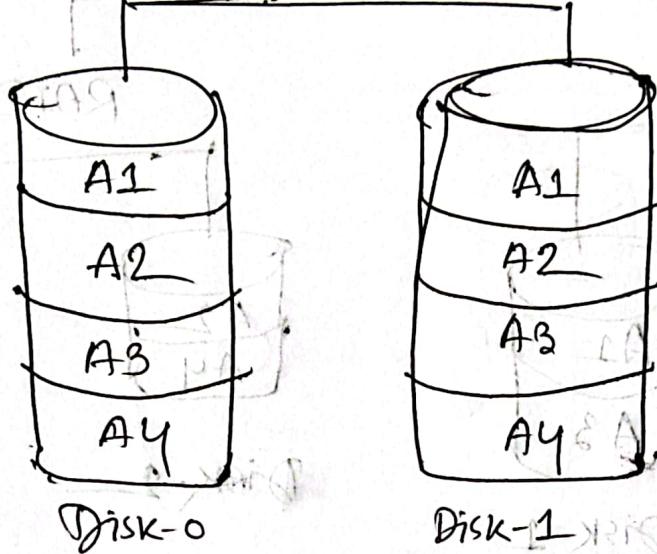
Redundant Array of Inexpensive Disks



Sat / Sun / Mon / Tue / Wed / Thu / Fri
□ □ □ □ □ □ □

Date : / /

RAID-1 → Mirroring



Performance
নিয়ে বিবরণ আছে
Availability
নিয়ে আছে

→ cost (কাম).
↳ disk price কম

- If one disk fails others disk work.
- Geographically Company ৩/৪ disk রাখত।

• data secured ৰে

RAID-1+0

- * Data কর একই parallel ~~প্রক্রিয়া~~
~~input~~ · read, write করত।
- * কর সকলে কিন্তু copy করে different different place
করে।

Combination of RAID 0 and RAID 1

Sat / Sun / Mon / Tue / Wed / Thu / Fri

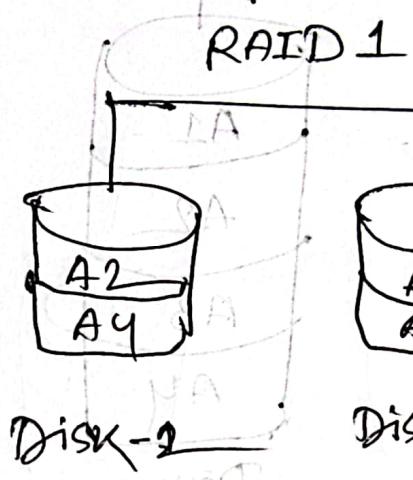
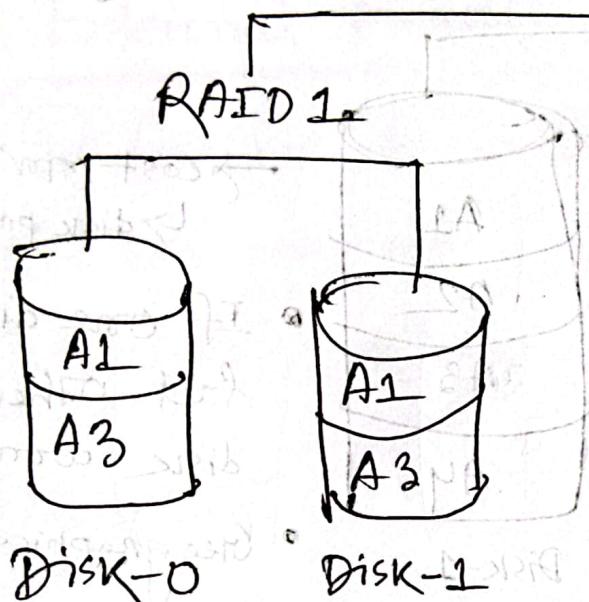


Date : / /

periority

RAID 0

* Real life
usable



RAID-2

RAID - 2

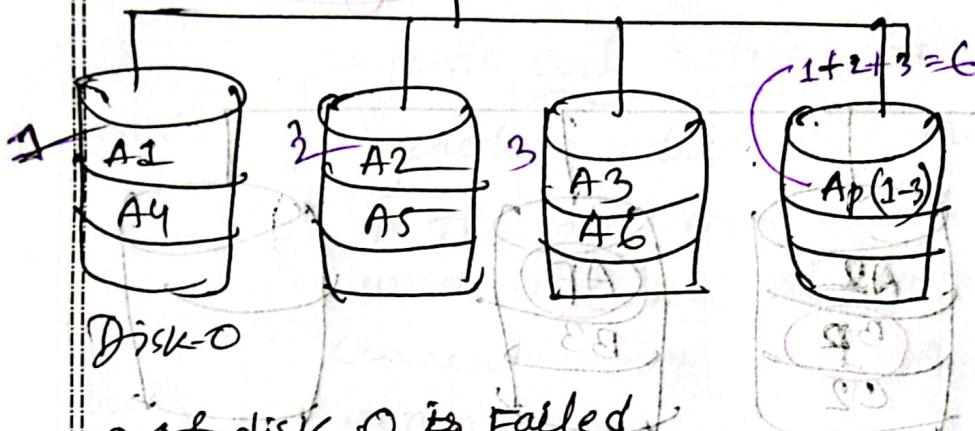
Hamming code

1. Disk option, bad. Error
2. Error detect and for
3. Error correctable
4. Error detect and correctable

RAID has 0 1 2 3 4 5 6 7 8

RAID 3

abit interrelated parity



- If disk 0 is failed, so, we do = parity - (disk1+disk2)

$$\begin{aligned}
 &= 6 - (2+3) \\
 &= 1
 \end{aligned}$$
 - If one of disk failed, we can recover by Parity

- Only one disk fail এলে আর কানুন।

• problem: parity disk problem এখন কোন এক না।
 \rightarrow n_1 দ্বিতীয় উৎস এবং

RAID · 4

block parity

* To overcome the disadvantage of RAID 3.

RAID 3, RAID architecture almost same

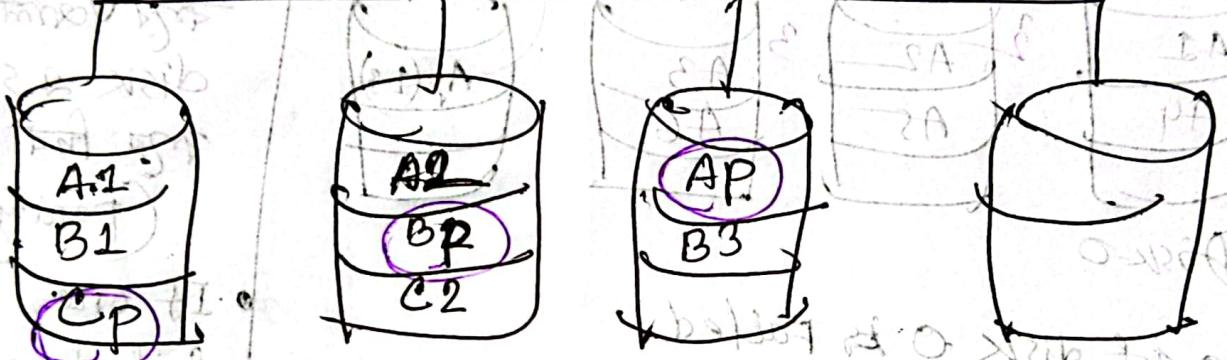
RAID 3 sobre 274 MB, almost RAID 3
almost RAID 4

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

RAID-5

Block level
distributed
parity



* parity \rightarrow disk 1 \rightarrow 2nd 2nd
disk 2 distribute 2nd 2nd

RAID-6

→ Dual redundancy

Same as RAID-6

→ but open disk store 2 parity

→ 2 for disk fail 2nd 3

data recover 2nd 3rd

RAID 10

► How data read from magnetic disk:

Read/write m/s

Start conducting coil (head)

magnetic coil rotate for reading

produce electrical current

in the coil. When surface rotate under the head it generate a

current with same polarity before generate.

book }
204 page }

Chapter-04

- 1 set associative mapping → Replacement algo in cache
- 2 Cache speed up: performance
- 3 Characterize memory of your computer.
- 4 direct associative diagram
- 5 cache / main memory with block diagram
- 6 basic elements of cache
- 7 direct mapping with diagram
- 8 relationship time, memory cost, capacity
- 9

• Cache-element

↓
mapping function

- Direct
- Associative
- set associative

$\frac{1100}{1101}$

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

Dividend

Divisor

7/3

Dividend

Divisor

A
0000

Q
0011

M = 011

initial values

0000
1101
0000

1100
1100
1100

shift
sub

Restore

0001
1110
0001

1100
1100
1100

shift
sub

Restore

0011
0000
0000

1000
1000
1001

shift
sub

Q₀ = 1

0001
1110
0001

0010
1001
0010

shift

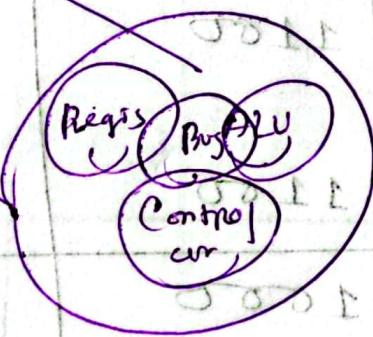
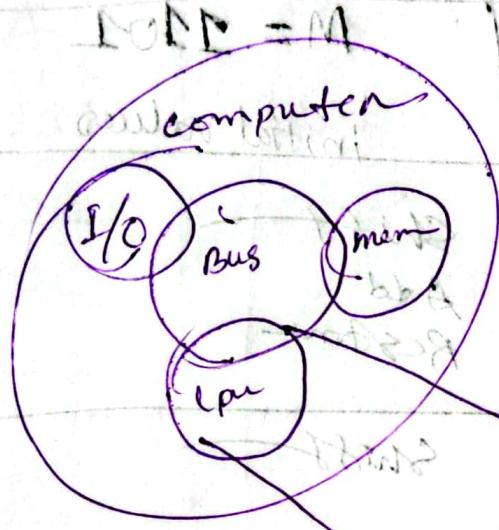
sub

Restore

Dividend
Divisor
Quotient to 2

Chap-09

7/3



Dividend
Division

7/3

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

0010
0011

0010

011

A	0	$M = 1101$ initial values
0000	0111	Shift Add Restore
0000	1110	Shift
1101	1110	
0000	1110	
0001	1100	
110	1100	
0001	1100	
0011	1000	
0000	.	
0000	1001	
0001	0010	
110	0010	
0001	0010	

reminder

1

2

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

dividends

divisor

-7/3

- A

01111 = M

Q

1001

~~8/7~~ M = 0011

initial values

1111 : 0010

0010

1111

0100

0100

1100

1000

1001

0100

0000

0000

0010

0001

1001

0100

0100

0100

~~1111~~

~~1111~~

0100

0100

0100

1000

0100

0011

1111

1111

1111

0100

0100

1111

1111

0010

1111

1001

0010

0010

1111

0010

1111

1001

0010

0010

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

~~-7/-3~~

~~0010
0011~~

1100-11	0	1001	M = 1101
1111	1001	0100 initial	0100
1111	0010	0000	1111
0010	0010	0000	0111
1111	0100	0000	1000
1110	0100	0010	0111
0001			
1110	0100	0000	0011
1100	1000	1000	1111
1111	1001	1000	1111
1111	0010	0100	0100
0010		0100	1111
1111	0010		

2
 8
 9 5th
 2.2.20
 2.2.20
 2.2.20

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

$$f = -19$$

$$M = 3$$

$$\begin{array}{r}
 10011 \\
 01100 \\
 +1 \\
 \hline
 1101
 \end{array}$$

8941

A	1110	M	G	B	M = 0011 A
1111	1111	1111	01101	1010	01111
1111			1010	01111	01111
0010			1010	01111	10100
1111			1010	01111	01111
1111			0100	00100	10111
0010			0100	00100	00100
1111			0100	00100	10111
1110			1000	00100	11010
0001			1000	00100	01000
1110			1000	00100	11010
1101			0000	00000	01101
0000			0000	00000	01101
1101			0000	00000	10111
1110	11010			01000	11010
00001				01000	01000
11110	11010			01000	10010
① 11101	10100		④ 11111	11100	10011
00000			④ 11111	11111	10011
11101	10100		④ 11111	11111	00110
② 11011	01000		⑤ 11111	00010	00110
11110	01001		⑤ 11111	11111	00110
11110					11010

19/7

11001
00110
01010
10011

01110
00111
00011

A 11001
11111

M 01101
11010

Q 10110
Initial values

M = 0111 A

11110
00101
11110

11010
11010

01001
01001

11111
01000

11101
00100
11101

10100
10100

00110
00110

11111
01000

11011
00010
11011

01000
01000

00011
00011

01111
10000

10110
11101
11101

10000
10001

01000
00011

10111
01000

11011
00010
11011

00010
00010

01011
00011

01111
10000

00100
+1

00101

Reminder = 5

Quotient
= 2

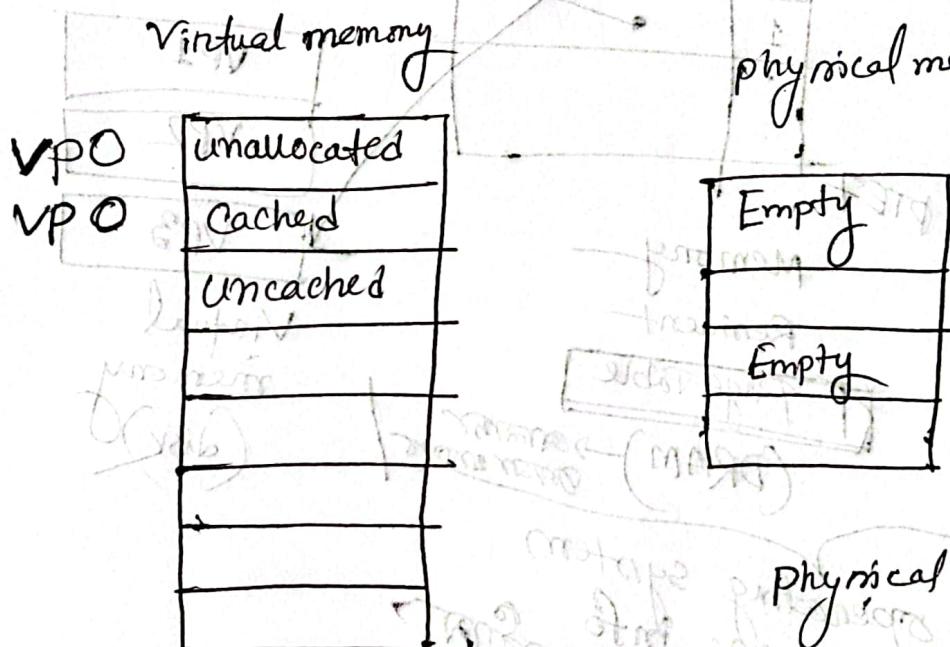
VM

Cache → RAM

FIFO
LRU
LFU

(I) Cache

(II) Memory management

(III) No corruption

Virtual pages

Cached in DRAM

→ Content of the array on disk
are cached in main memory.

→ VM systems handle this by partitioning
the virtual memory into fixed sized
blocks called virtual pages.

→ Physical memory → Physical Pages

Page table:(Virtual page physical
page relation define ~~map~~)page
Table
Entries