Lecture-01

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

Date : 02/10/2023

Introduction to Microprocessors

-> If you want to do programming, you need a processor.

-> From Pentium Series — 80586 (Pentium 1) -1993

We can engage with multimedia using computer. (watch movies)

8085

8086 → We need primary memory (RAM/ROM)

RAM → Volatile

80186

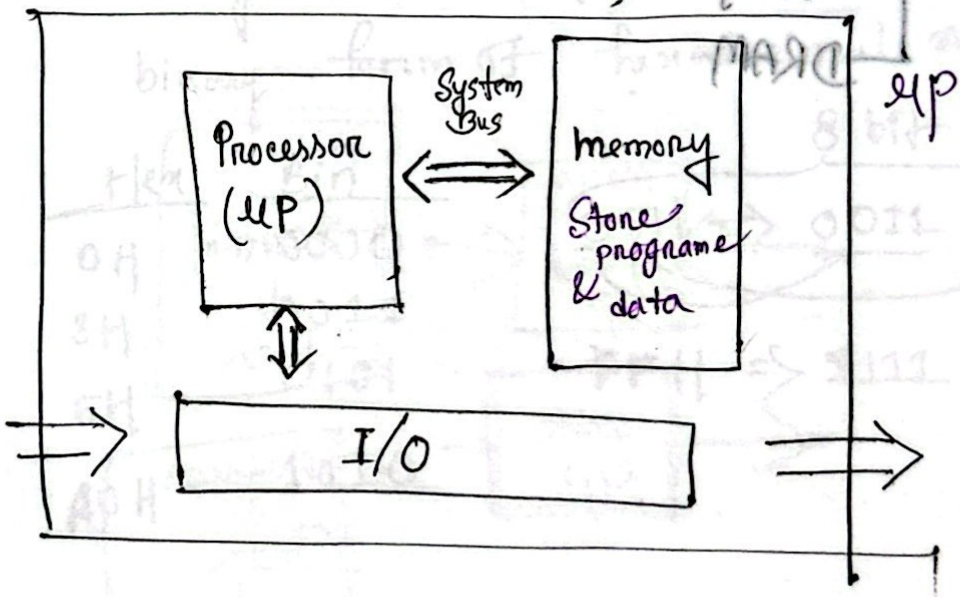
286

386

486

80586 → P_1, P_2, P_3, P_4 ... C2D ... Ci7

Computer System:



μP = Microprocessor

Sat / Sun / Mon / Tue / Wed / Thu / Fri
Date :...../...../.....

- In 8086, memory means,

primary memory → RAM, ROM → (BIOS)

ROM → Read only memory (can only) → permanant information

RAM → Random Access memory (volatile) → can read & write → Temporary → when we stop computer RAM erase everything.

- everything stored in a memory (movie, image) by many 0 and 1.

- RAM
  - SRAM
  - DRAM

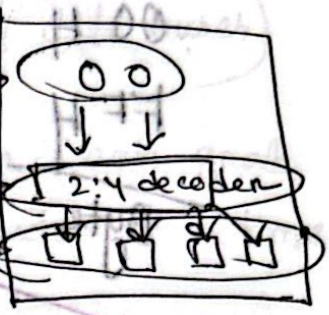
Sat / Sun / Mon / Tue / Wed / Thu / Fri

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

Microprocessor :

instruction cycle

fetch
decode
execute



⇒ necessary of hexadecimal number system in computer:

→ 10 কে decimal দিয়ে represent করা যায় না,

→ 10 কে A দ্বারা hexadecimal এ " " যায়।

9999 →

FFFF → 65,000 (It'n appropriate)

- Inside Computer everything is binary number. And it is the binary form of hexadecimal number.

Hex	Bin
0H	0000
3H	0011
5H	0101
A0H	10 10

8 bit → 8 bit numbers

35H ⇒ 0011 0101

FFH ⇒ 1111 1111

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

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

1024 → 1K (Kilo)

8 bit number	16 bit number
00 H	0000 H
FF H	FFFF H
Byte	Word

$2^{10} = 1K$ (Kilo)

$2^{20} = 2^{10} \times 2^{10} = 1K \times 1K$

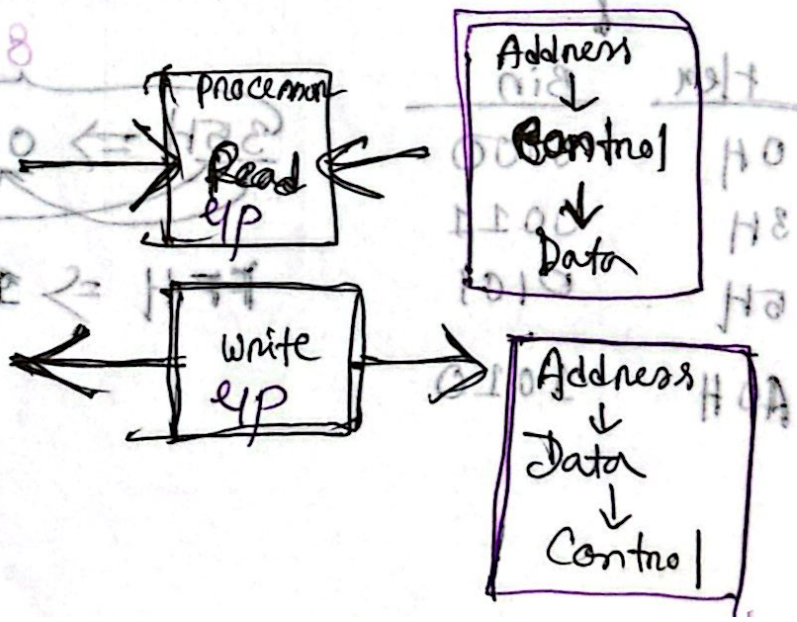
$= 1M$ (mega)

$2^{30} = 1G$ (Giga)

$2^{40} = 1T$ (Tera)

Bus:

- Set of lines
  - Address bus
  - Data bus
  - control bus (Read/write)



Lecture-02

Pipelining

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

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

For processor faster:

→ increase the buses (specially data buses)

↓

processor can grab more data at a time

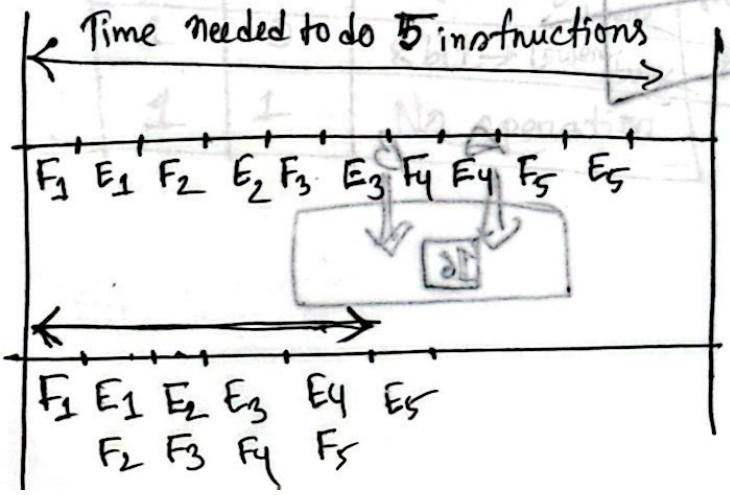
pentium 1 – today's processor

64 bit data bus

but, pentium 1 থেকে বর্তমান processor অনেক fast. So, there is a concept of pipelining to faster processor.

↳ It is the process of Fetching the next instruction while executing the current instructions.

- When we execute, buses are free that time. So, during execution we can fetch the next instruction.



Non-Piplining
Eg – 8085

Piplining
Eg → 8086

Lecture-3

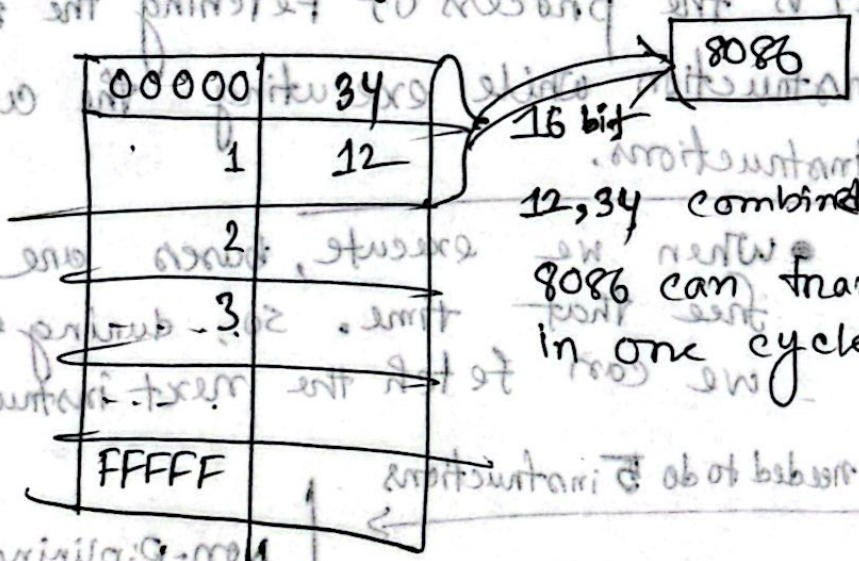
Memory Banking

8086
16-bit, ~~up~~
↓
because it has 16 bit data bus

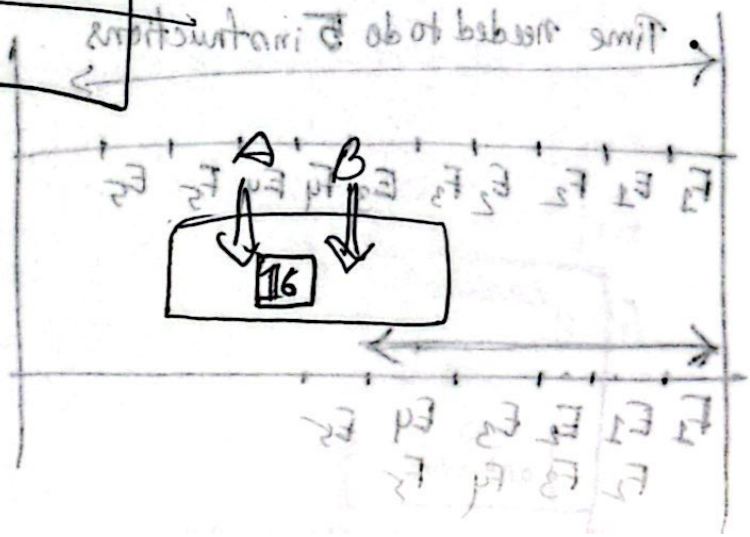
Even addresses → Even Bank → Lower Bank
Odd addresses → Odd Bank → Higher Bank

- All modern processor has 8 bank (64 bit)

▶ 8085, 8 bit processor execute 8 bit data at a time. So, non banking on 8085. → So, access one location at a time



12, 34 combinedly 16 bit
8086 can transfer 16 bit in one cycle.



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

Date .....

JMP

Odd Bank (Higher)	Even Bank (Lower)
00001H	00000H
00003H	00002H
00005H	00004H
⋮	⋮
FFFFFH	FFFFEH

A_3 A_2 A_1 A_0 → A_0 দিয়ে শুরু।

00000	0H
00001	1H
00010	2
00011	3

তাই A_0 বাদে সবগুলো নিলে

0,1 same

2,3 same

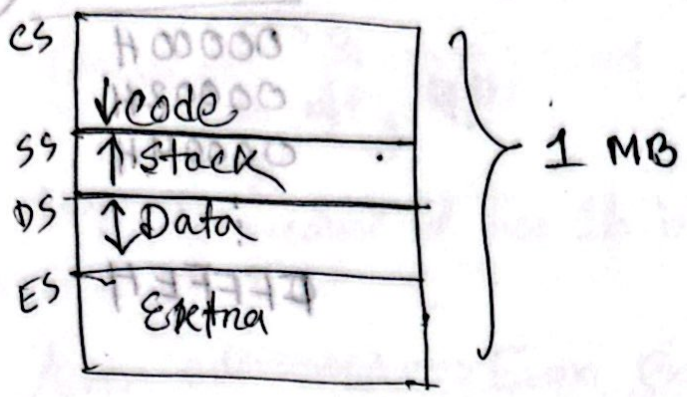
তাই এগুলো align

So, একসাথে access করা যাবে।

BHE	A_0	operation	
0	0	16 bit	MOV CX[2000H]
0	1	8 bit → higher bank	MOV CH[2001H]
1	0	8 bit → lower bank	MOV CL[2000H]
1	1	No operation	

Lecture - 04
Memory segmentation

Sat / Sun / Mon / Tue / Wed / Thu / Fri
Date :...../...../.....



segmentation prevents overnight.

$2^{20} = 1MB \rightarrow$ Physical address

20 bit $\rightarrow$ address | bus — 1MB

16 bit $\rightarrow$ address $\rightarrow$ compatible

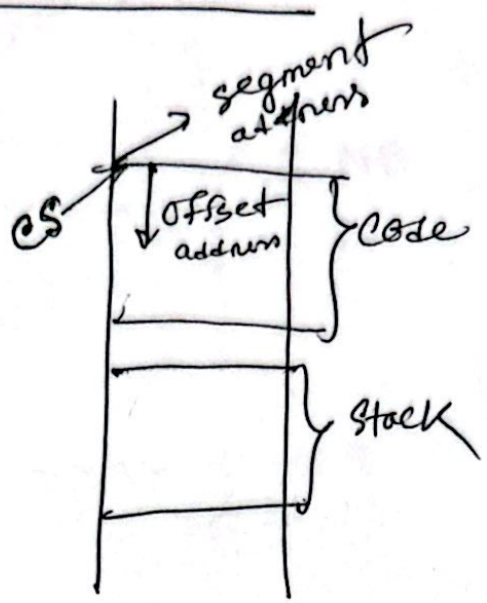
Class-05

8086 Architecture

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

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

8080 Architecture :



~~log~~

- location 564 page

Chapter-5

↳ Segment address

64 → offset address

$PA = Seg \times 10_H + offset$

$PA = CS \times 10_H + IP$

↳ It is the real address

↓

Calculated by processor

BIU → Fetch

↓

Calculate PA

EU → execute

MOV BL, CL → opcode

MOV BL, 25H

opcode — operand

↓

we decode opcode

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

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

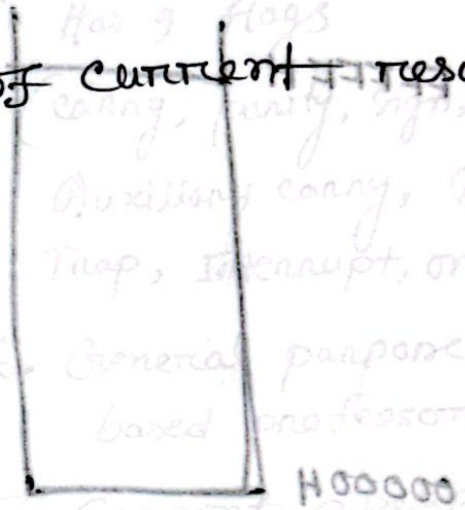
MOV: Bx, Cx ; Bx ← Cx

MOV BL, 04H
MOV CL, 05H

ADD BL, CL ; BL ← BL + CL
CL ← CL + BL

Flag

↳ contain the status of current result.



Think X

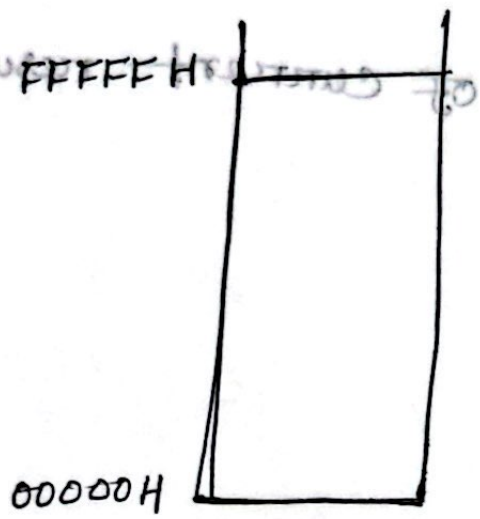
Real Mode memory addressing :

operation

up addresses memory using Logical addresses to the memory space

1 MB

Conventional (DOS memory system)



Windows — X
Real mode X

Question solve:

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

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

8085	8086
1. 8 bit microprocessor	1. 16 bit microprocessor
2. 8 bit data	2. 16 bit data bus
3. 16 bit address line	3. 20 bit address line
4. only 64 KB of memory can be used. (2^{16})	4. 1 MB of memory can be used. (2^{20})
5. Has 5 flags (Carry, Parity, Sign, zero, Auxiliary carry)	5. Has 9 flags (carry, parity, sign, zero, Auxiliary carry, Direction, Trap, Interrupt, overflow
6. Accumulator based processor.	6. General purpose register based professor.
7. Does not support pipelining	7. Support pipelining
8. Does not support memory segmentation	8. Supports memory segementation.

Von Neumann machine:

A machine that stores the instructions of a program in the memory system.

Scanned with CamScanner

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

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

How is the local descriptor table addressed in the memory system?

Fetch
Decode
Execute

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

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

(#) Which register locates the global descripton table?

8086
Assembly Instructions

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

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

MOV AX, 1234H

Consume 3 byte

CS:IP

Logical address

```
 01000100
 10111011
        +1
 -------
 10111100

 00100010
 10111100
 -------
 ~~00110~~
 -------
 11011110

 00100001
        1
 -------
 00100010
```

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

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

1 → Single key input

2 → Single Character output

9 → Character String output

lea

load effective address

int (interrupton)

int 21h

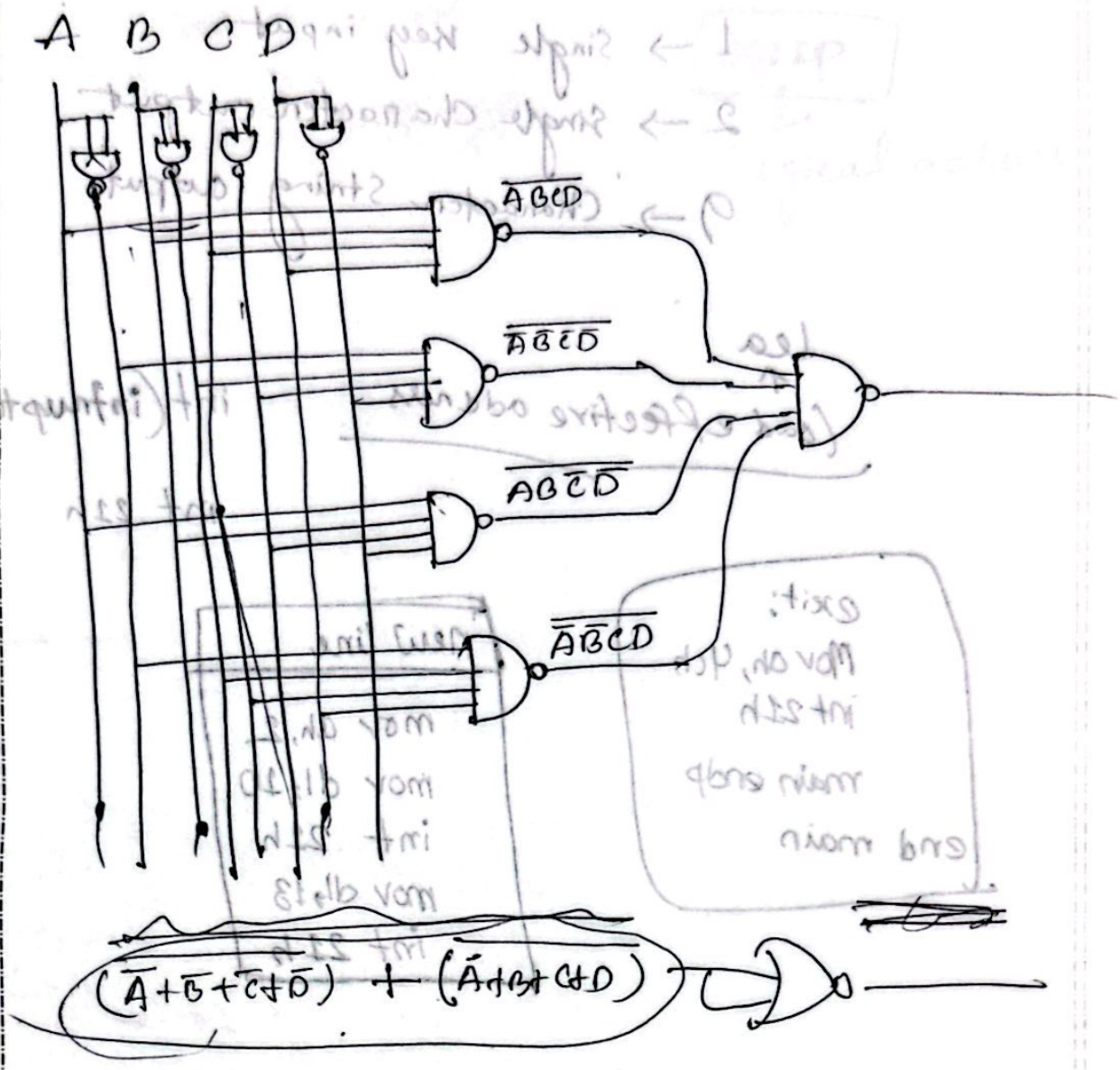
```
exit:
mov ah, 4ch
int 21h
main endp
end main
```

new line

```
mov ah, 2
mov dl, 10
int 21h
mov dl, 13
int 21h
```

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

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



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

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

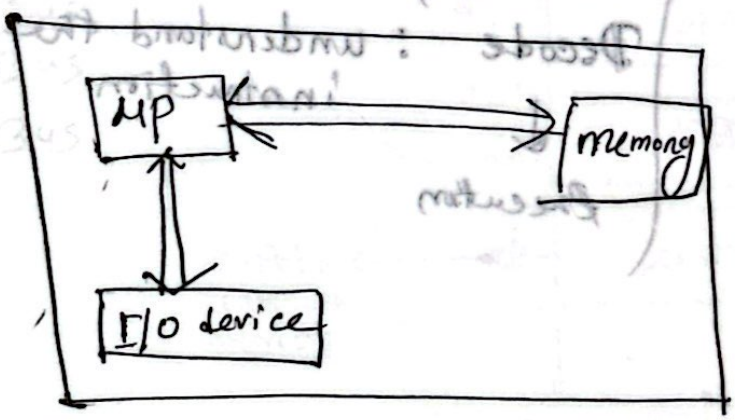
Microprocessor Introduction

Microprocessor যেসকল device এ use করা হয়, যেসকল device এ programming করা হয়।

Microprocessor program execute করে।

8080 → major modification
8085 ← 5V stable supply

86 Family:
- *8086
- 80186
- 80286
- *80386
- 80486
- 80586 → P_1, P_2, P_3, P_4 --- Core 2 duo — core is/ছিল
  - P_1 → multimedia — Support দেয়



computer system

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

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

memory

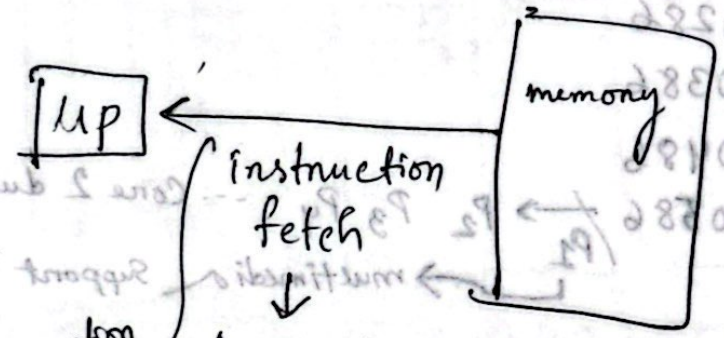
primary
↳ RAM/ROM
↓
Hard coded
Fast

Secondary
↳ Hard disk, SSD, CD, DVD
S/w

— SRAM — (cache memory)

— DRAM

— SDRAM



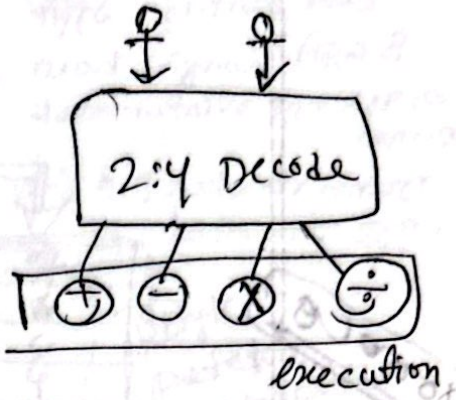
instruction cycle

instruction fetch
↓
instruction Decode : understand the instruction
↓
Execution

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

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

ADD 00
SUB 01
MUL 10
DIV 11
ALL LLL



Data	Address
00	Addition

BUS System :

Lines that transfers 0's and 1's

- A. Bus
- D Bus
- C Bus

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

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

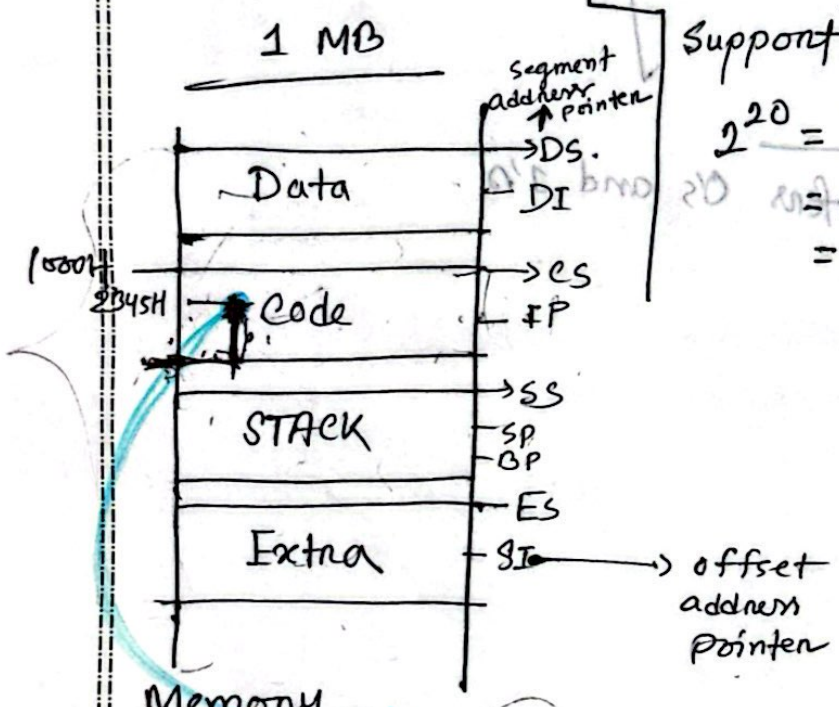
Architecture 8086

- 16 bit microprocessor
- It's mean
- at a time 16 bit data এর উপর process করতে পারে।
- pipeline microprocessor
  → যখন EU তে execution চলবে তখন BIU memory থেকে instruction fetch করা শুরু করে দিবে (parallel ই চলে execution - Fetch)

FEEF

- 20 bit address bus support করে।

$2^{20} = 2^{10} \times 2^{10}$
$= 1KB \times 1KB$
$= 1MB$



PA = (Segment address X 10H) + offset address

$= (10000H \times 10H) + 2345H$

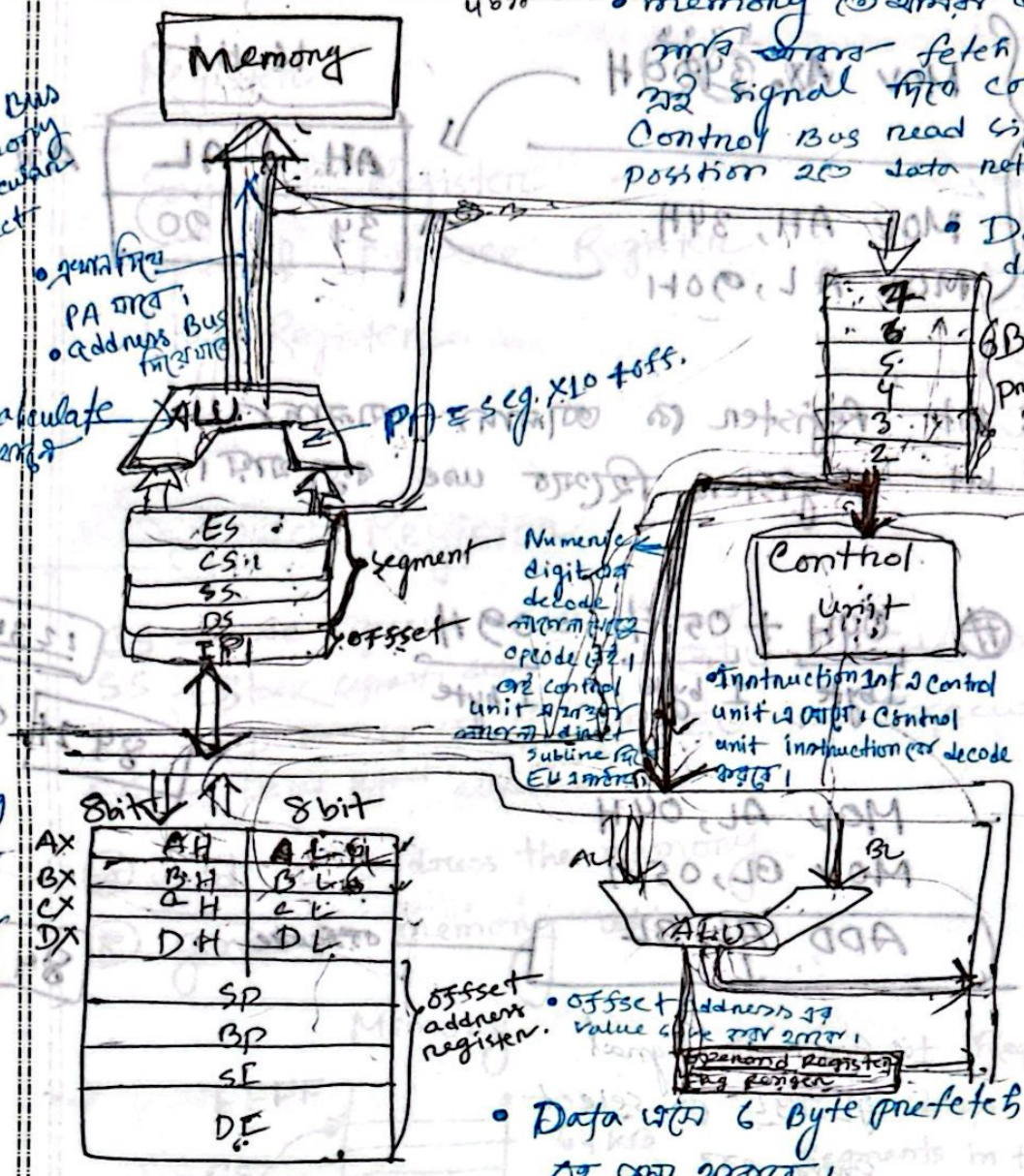
$= 12345H$

MOV AX, BX

Add AL, BL

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

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



• memory তে থাকবে data পাঠাতে পারি আবার fetch করতে পারি। এই signal দিয়ে control Bus. Control Bus read signal দিলে position এ data retrive হয় up তে আসবে।

• Data bus আসবে data বাস হবে।

• address Bus যেটা memory থেকে particular section select করছে।

• প্রথমবার PA যাবে। • address Bus দিয়ে যাবে

• PA calculate হবে এখানে

PA = Seg. X10 + off.

6 Byte prefetch instruction Queue

Numeric digit এর decode করলে পাবে opcode এই। এই control unit এ আসবে Control unit এ আবার Control unit instruction কে decode করবে।

• 8086 microprocessor maximum 6 byte এর instruction Suffont করে।

• prefetch এ যখন দুটো byte ফাঁকা হবে তখন memory থেকে আবার fetch করা শুরু করবে। Single cycle এ দুটো fetch হবে।

• offsset address এর value এই রকম আসবে।

• Data এসে 6 Byte prefetch Queue (FIFO) তে জমা থাকবে।

1 – EU

2, 3, 4, 5, 6, 7 } prefetch

8

9

• Jump instruction এলে unnecessary instruction Flush করে দিবে।

100010

opcode

MOV AL, BL

opcode part

MOV AL 25H

1 byte

• opcode হলো 0 & 1 এর byte pattern প্রত্যেকটা রেজিস্টারের জন্য আলাদা আলাদা opcode থাকে। Numeric value এর opcode নাই না

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

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

MOV AX, 3420H

MOV AH, 34H
MOV AL, 90H

16 bit		
AH	AL	AX
34	90	

• 16 bit Register কে আলাদা আলাদা 8 bit Register হিসেবে use করা যায়।

04H + 05H = 09H
1 byte 1 byte 1 byte

MOV AL, 04H
MOV BL, 05H
ADD AL, BL

* Control signal AL, BL কে select করবে এবং ALU কে add এর নির্দেশ দিবে।

1234H

84 21

12H

84H

Register

Segment Register
Special purpose Register
Flag Register

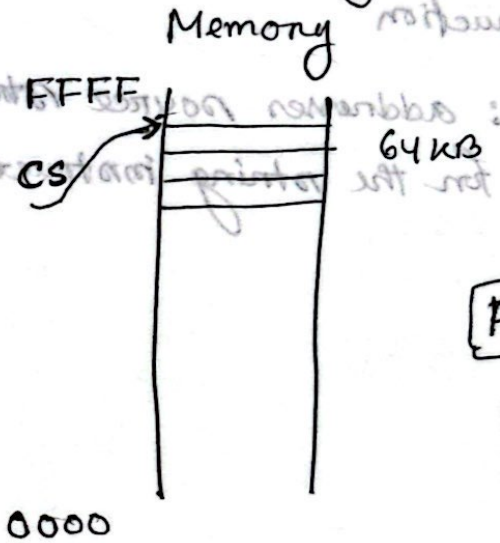
~~13:15~~
~~17:09~~
~~16:27~~
~~14:08~~
~~10:40~~
~~17:34~~
~~19:00~~

Segment Register

CS — code segment : hold the code
SS — Stack segment : define area of memory
DS — Data segment : contains most data used by a prog.
ES — Extra segment : additional "

BIU — Fetch, Decode
EU — Executing

* used to address the memory
* generates memory addresses



* Segment Register are used to address particular segments in this memory.

$$\boxed{PA = seg \times 10H + Offset}$$

• Book (Memory)
↓
Chapters (segment
Topics [page no]

Sat / Sun / Mon / Tue / Wed / Th / Fri

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

Multi purpose Register :

- RAX (accumulator) : is used for instruction such mul, Div and...
- RBX (base index) : holds offset address of a location in memory
- RCX (count) : holds ~~various~~ count for various instruction
- RDX (Data) : holds a part of result from a multiplication & dividend before devision

RBP (Base pointer) : points to a memory location for memory data transfer

RDI (destination index) : destination string data

String instruction

RSI (Source index) : addresses source string data for the string instruction.

Special Purpose Registers

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

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

RIP (Instruction pointer):

IP points to the next instruction in a program, is used by up to find the next sequential instruction in a program.

RSP (stack pointer):

Addresses an array of memory called stack.

Stack memory stores data using RSP.

RFlags:

condition of up and control its operation.

Flag Register : of 8086 :

Shows the status of Result.

→ contains the status of current result.

STATUS Flag:

changed by the ALU

Carry Flag:

```
  1111
  0001 :
 -----
 1 0000
```

→ If carry ~~flag~~ coming out of the MSB. That mean carry Flag = 1.

Parity Flag:

1000 0011

here 3 one's

so, odd parity.

So; Parity Flag = 0

Even parity = 1
odd parity = 0

Auxiliary Carry:

higher nibble Lower Nibble (4 bit)

```
 1111 | 1111
 0000 | 0001
 -----------
 0000 | 0000
```

- Lower Nibble থেকে higher nibble এ Carry গেলে Auxiliary carry = 1.

Scanned with CamScanner

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

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

④

Zero Flag:

Zero Flag = 1 → Result is 0

Zero Flag = 0 → Result not 0.

Sign Flag:

OF	PF	AF	ZF	SF	CF
0	1	0	0	0	0

$\boxed{1}000\ 0001$

If MSB is 1, number is negative = 1

If MSB is 0, " positive = 0

$\boxed{\text{MSB Result} = \text{Sign Flag}}$

Overflow Flag:

If number goes out of range then overflow = 1

$$\begin{array}{r} 0111\,1111 \\ 1111\,1111 \\ \hline 1\,0000\,0000 \end{array}$$

OF = 1

Sign Flag check না করে overflow flag check করা লাগবে।

OF = 1 → Sign Flag give wrong sign

OF = 0 → Sign Flag give right sign

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

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

-80H , -01, 00, 7FH

যদি overflow হবে।

-128 . . . 0 . . . +127

7F
+1
80

(#)

42H → 0100 0010

23H → 0010 0011

65H 0110 0101

OF	SF	ZF	AF	PF	CF
0	0	0	0	1	0

(#)

37H → 0011 0111

29H → 0010 1001

60H 0110 0000

Same হলে Same

সেলে - 0

ভিন্ন সেলে - 1

OF	SF	ZF	AF	PF	CF
0	0	0	1	1	0

8	4	2	1
1	0	0	0
0	1	0	1

(#)

42H → 0100 0010

43H → 0100 0011

85H 1000 0101

OF	SF	ZF	AF	PF	CF
1	1	0	0	0	0

added two positive numbers (42,43) never be negative. Because overflow 1.

Scanned with CamScanner

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

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

Control Flags: → Changed by us.

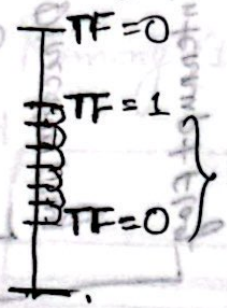
TRAP Flag:

Single stepping

↳ Debug every line by line

1 = perform single stepping

0 = do not ,,



execute line by line

Interrupt Flag:

process gets interrupt from external sources.

↳ Mobile screen on ~~thakle~~ থাকলে touch করলে interrupt হয়।
screen off ~~thakle~~ থাকলে interrupt হয় না, because we can disable interrupt.

1 = Enable interrupt

0 = disable ,,

Direction Flag:

works for → String instructions

Choose auto incrementing addresses or decrementing addresses.

1 = auto ~~increment~~ decrement

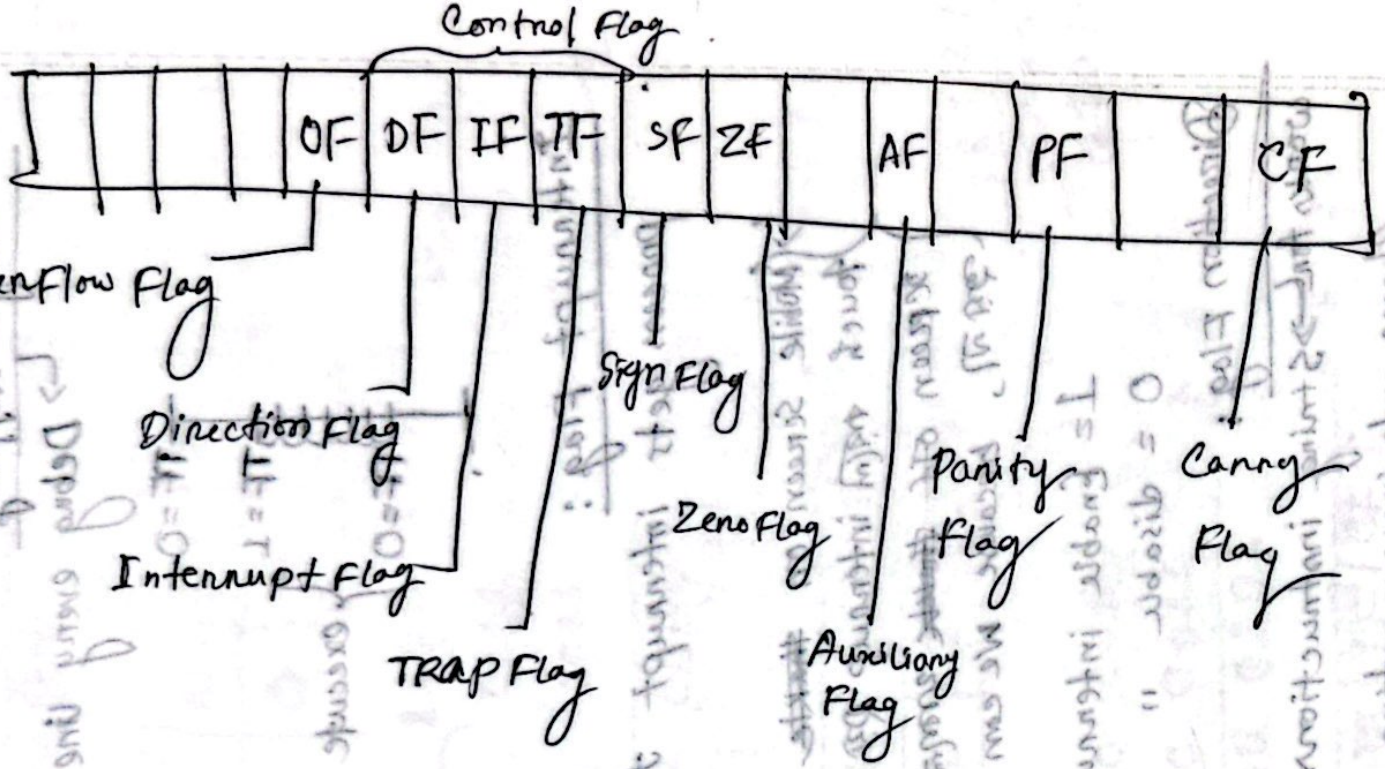
0 = auto increment

2-20

Scanned with CamScanner

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

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



Scanned with CamScanner

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

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

Real mode memory addressing: access only first 1 MB

operation → to address only 1MB of memory.

up addresses memory using logical address

2FFFF

20000

00000H

CS

DS

2^{20} = 1MB

2^{20}

16 bit

2^{16}

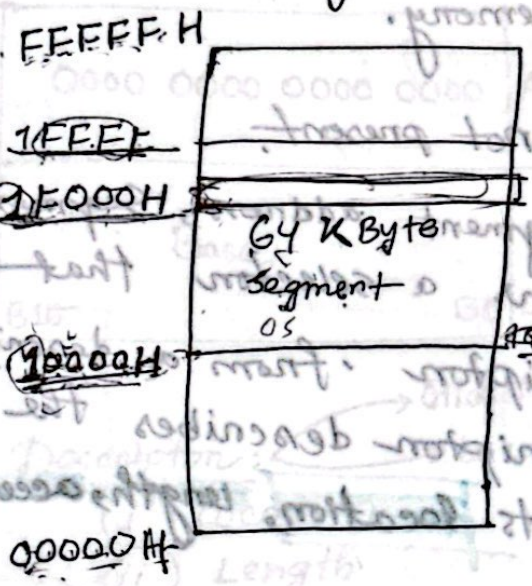
16

2^{10}

64KB

64KB

Memory (1MB) conventional/ DOS memory system.



offset = F000 → windows Real mode use मा हो।

→ Real mode allows application software — 8086/8088

(*) 64 bit cannot run in real mode

Segment Register
2000H

Starting address
20000 H

Ending address
2FFFFH

Offset

2^{16}

2^{10} 2^{6}

64KB

▸ Segment address ↳ memory

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

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

Protected mode memory addressing: above 80286

→ Allows up to access all the data & ~~as w~~ programs located above 1 Mbyte of memory. as well as within 1st 1 Mbyte of memory.

→ Segment address not present.

↳ In place of segment address, segment register contains a selector that selects a description from a description table. The description describes the memory segments location, length, access rights.

Selector —selects— Description (2X8192).

→ Offset address can be 32 bit no. instead of a 16 bit no in the protected mode.

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

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

$2^{10} = 1KB$

$2^{20} = 1MB$

$2^{30} = 1GB$

$2^{40} = 1TB$

Selectors and descriptors :

→ selects one of (2X8192) descripton

Segment Register

$\boxed{\text{Selector}}$

80286

31		0
0000 0000 0000 0000	Access Rights	Base (B23 … B16)
Base (B15 … B0)	Limit (L15 … L0)	

4

0

Offset

2^{24}

$2^4 \cdot 2^{20}$

16MB

2^{32}

$2^2 \cdot 2^{30}$

4 GB

Descripton : → Global descripton → System → applies all program

→ Local descripton → Applications

(i) Location

(ii) Length

(iii) Access Rights

Location { Base address → starting location of memory segment.

80286 → 24 bits (Segment at any location 16 MB memory)

80386 → 32 bit (4GByte of memory)

$24 = 2^{24} = 2^4 2^{20}$

$= 16 MB$

$32 = 2^2 \cdot 2^{30} = 4GB$

Scanned with CamScanner

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

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

Length → size of segment.

Limit → contains the last offset address in a segment

80286 → 16 bit [Base + Limit] (1-64 KB)

80386 → 20 bit (1 MB or 4 KB or 4 GB in length)

G — bit (Granularity bit):

G=0, limit has length of 20 bit. FFFFF H

G=1, ~~limit~~ appends/multiply it by FFFH

(multiply by 4 K bytes)

FFFFFFFFF H

1MB × 4KB

= 4 GB

G=1

D-bit:

D=0, 16 bit (operand)

D=1, 32 bit

AV bit:

AV=0, segment not available

AV=1, " available

1MB * 4KB

2^{20} 4×2^{10}

4×2^{30}

4GB

$2^{22} \cdot 2^{10}$

4GB

2^{12} 2^{20}

2^{32}

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

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

Access Rights Byte:

→ contains access to protected mode segment

→ If segment goes beyond this limit, program is interrupted by microprocessor indicating a general protection fault.

→ Can specify wheather a data segment can be written or is write protected.

7	6	5	4	3	2	1	0
P	D	PL	S	E	ED/C	R/W	A

Access Rights Byte

15	3	2	1	0
selectors		TI	RPL	

(13-bit)

Segment Register

(80286 – core2)

TI=0 (Global description)

TI=1 (Local description)

RPL – Request Privilege level of a memory segment

PL (00) – highest priv

PL (11) – lowest priv

00 – high

→ 10

01

→ 11 – low

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

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

Access Rights

Segment register

- PL < RPL

access is granted.

A = 0 → segment not accessed

A = 1 → segment can be accessed

R = 0 → not read

R = 1 → read

W = 0 → not write

W = 1 → write

ED = 0 → segment expand upward (data segment)

ED = 1 → segment expand downward (stack segment)

C = 0 → Ignore DPL

C = 1 → abide by PL

DPL → Descripton privilege level.

S = 0 → System descripton

S = 1 → code / data segment descripton

P = 0 → descripton undefined

P = 1 → Segment valid base + limit

E = 0 → descripton describe data segment

E = 1 → " " code

Chapter-03
Addressing Modes

Sat / Sun / Mon / Tue / Wed / Thu / Fri
Date :...../...../.....

MOV AX, BX

Addressing modes : (Bharat)

mov destination, source

→ In which an operand is given in an instruction

25

1 Immediate:

- data in the instruction

Eg: mov CL, 34H; CL ← 34H

mov CX, 2000H; CX ← 2000H

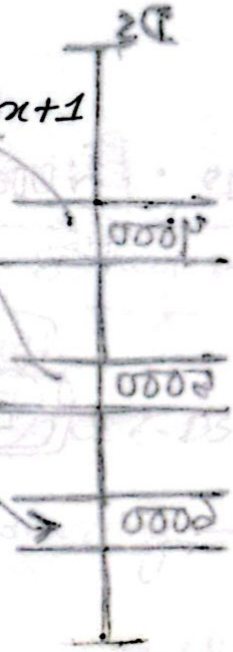
2 Register:

- data inside the Register

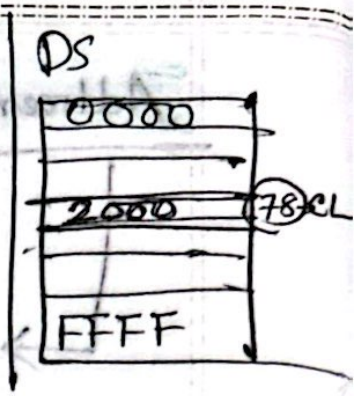
Eg: mov CL, BL ; CL ← BL

mov CX, BX ; CX ← BX

INC BX ; BX ← BX+1



③ Direct:



- address in the instruction
  ↳ Telling the processor go to the address from there you will get the data

mov CL, [2000H] ; CL ← DS : [2000H]

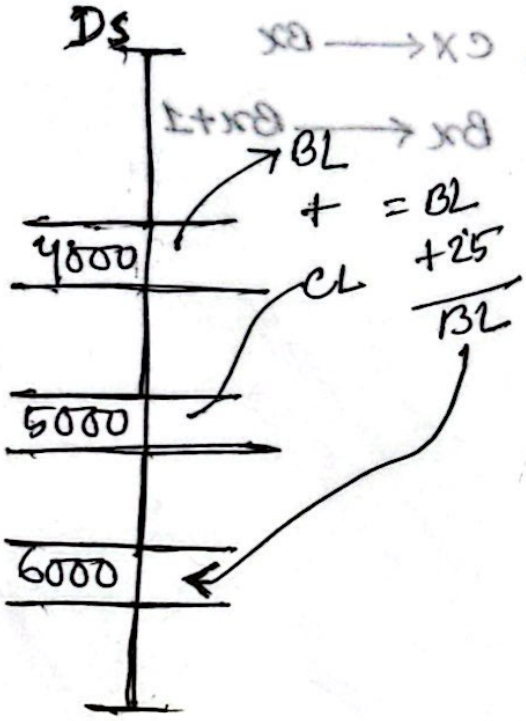
Offset address — Data Segment — offset address

[ ] square indicating address

mov CX, [2000H] ; CL ← DS: [2000H]
CH ← DS: [2001H]

One location always carry 1 Byte of Data

mov [2001H], CL ; DS: [2001H] ← CL



MOV BL, [4000H] → Direct
MOV CL, [5000H] → Direct
Add BL, CL → Registers
Add BL, 25H → Immediate
MOV [6000H], BL → Direct

Scanned with CamScanner

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

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

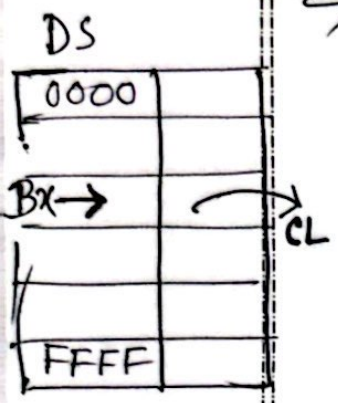
④ Indirect : → 32 type

[more than 95% of operation using indirect addressing]

Well developed addressing mode

- Address in Register

↳ In the instruction there will be a register, processor will go to that register. That register will given address. processor will go to the address. From there processor get the data.



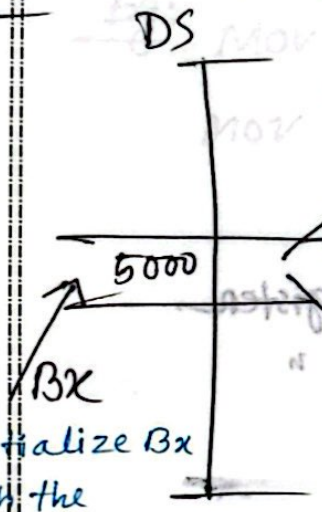
(I) Register Indirect: (address ← Register)

Eg: mov CL, [BX] ; CL ← DS: [BX]

mov cx, [BX] ; CL ← DS: [BX]

CH ← DS: [BX+1]

- arr[9] - direct
  i=9
  arr[i]
  ↳ indirect



MOV CL, [5000H] ; CL ← DS: [5000H]

Direct addressing

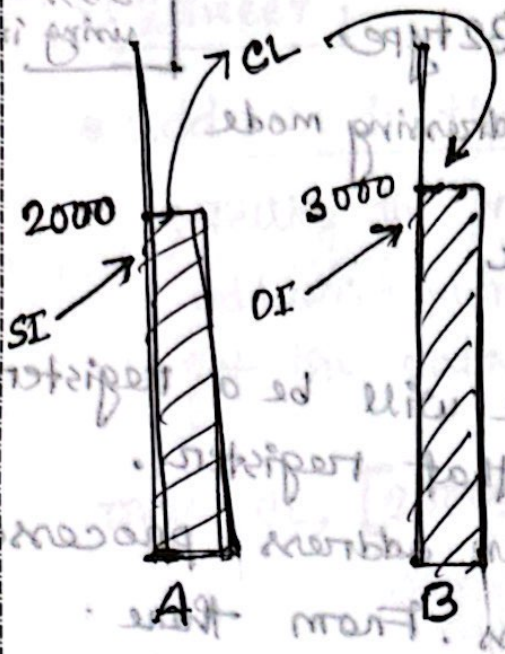
MOV BX, 5000H

MOV CL, [BX] ; CL ← DS: [500H]

- Initialize Bx with the value [5000H]

Indirect addressing → By increment Bx we can get multiple address value

→ use for series of location




```
MOV SI, 2000H
MOV DI, 3000H

MOV CL, [SI]
MOV [DI], CL
INC SI
INC DI
```

Block trasfer A to B

AX
BX → only 1 register can use for memory address.
CX
DX

BP, SI, DI → Offset Register can use

Bx, SI, DI	→ DS
BP	→ SS

Bx, BP → Base Register
SI, DI → Index "

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

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

(II) Register Relative:

(address ← Register + displacement)

Bx = 2000	78
2001	81
2002	95
Bx+3 = 2003	27

MOV CL, [Bx+03H] ; CL ← DS:[Bx+03H]

MOV CL, [BP+05H] ; CL ← SS:[BP+05H]

(III) Base Indexed:

(address ← base register + index Register)

Eg: MOV CL, [Bx+SI] ; CL ← DS:[Bx+SI]

MOV CL, [BP+SI] ; CL ← SS:[BP+SI]

$\boxed{Bx + index}$

By incrementing index we can get series of data.

Scanned with CamScanner

(IV) Base Relative Plus indexed:

(address ← base Register + index Register + displacement)

Ex:

MOV.CL, [Bx+SI+03H]; CL ← DS: [Bx+SI+03H]

Implied:

STC; CF ← 1

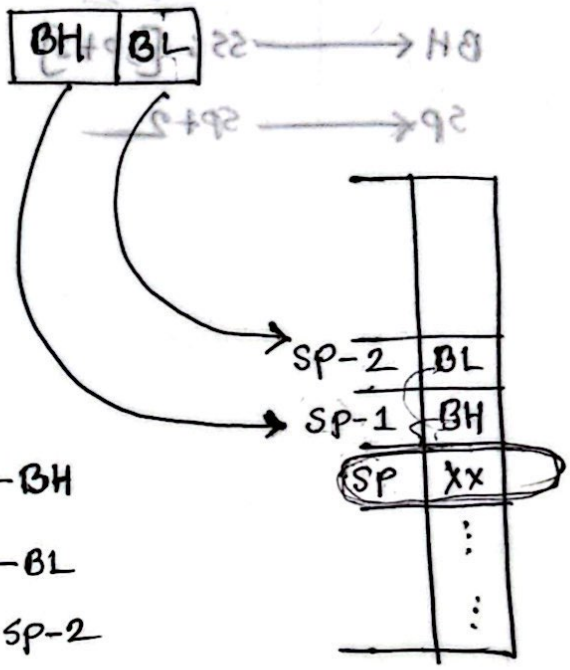
CLC; CF ← 0

DAA; Decimal Adjust after addition

push & pop Instructions:

- push and pop don't support Immediate addressing mode.
- must be 16 bits Register (Word)

(I) push Bx:

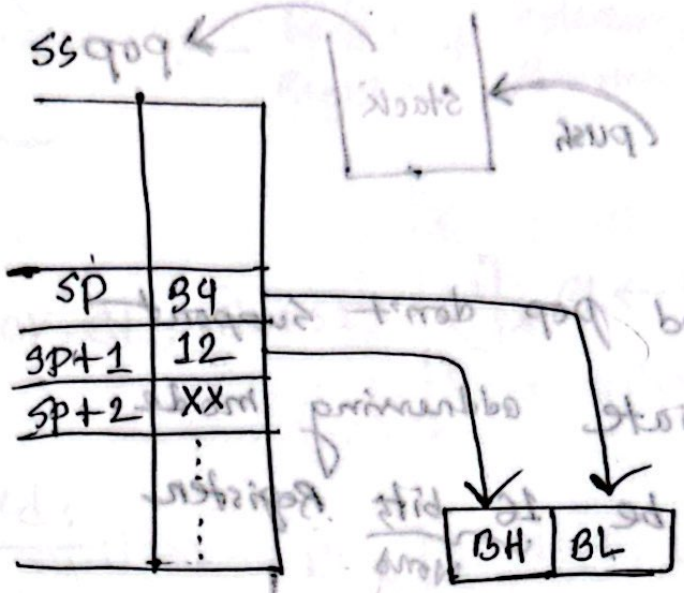


SS: [SP-1] ← BH

SS: [SP-2] ← BL

SP ← SP-2

(II) POP BX :



BL ← SS:[SP]

BH ← SS:[SP+1]

SP ← SP+2

Chapter-04

Data Movement Instructions

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

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

70 taka

65 taka

35 taka

4-1 MOV : Revisited

Machine Language

The opcode

MOD Field

Register Assignments

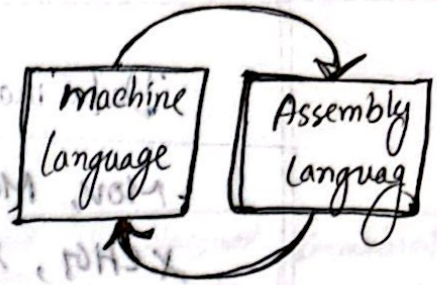
R/M Memory addressing

special addressing Mode

32 bit — Addressing modes

An immediate instruction

Segment mov instructions



*** Conversion ***

- Table

4-2 push/pop

Data movement instructions:

MOV, MOVSX, MOVZX, PUSH, POP, BSWAP
XCHG, XLAT, IN, OUT, LEA, LDS, LES, LFS
LAHF, SAHF

String Instructions:

MOVS, LODS, STOS, INS, OUTS

MOV AX, 25H

Instruction Format:

MOV AX, BX + [1000H]
↳ 010 . . .

Displacement

8BEC H
↳ Binary
↓
Binary form থেকে instruction analysis করতে হবে।

1) 16 bit:

1000 1011	11 10 1 100	8BEC H
opcode	MOD-REG-R/M	

MOD, REG, R/M

8 – 1000
B – 1011
E – 1110
C – 1100

opcode	MOD-REG-R/M	Displacement	Immediate
1-2 bytes	0-1 bytes	0-1 bytes	0-2 bytes

MOD
00
01
10
11

Scanned with CamScanner

MOV AX, BX

66 8BECH

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

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

2) 32 bit:

67H 66H

Address Size 0-1 bytes	Register Size 0-1 bytes	Opcode 1-2 bytes	MOD-REG-R/M 0-1 bytes	Scaled index 0-1 bytes	Displacement 0-4 bytes	Immediate 0-4 bytes

• 32 bit start with 66/67H.

Opcode:

i) Assembly → Binary

ii) Binary → Assembly

MOV · AX, BX ▶ 8BECH

opcode

16 bit

1000 1011 | 1110 1100

Byte 1 | Byte 2

Assembly:

Byte-01 D W MOD Byte-02

1000 10 1 1 | 11 110 100

REG R/M

opcode : 100010 → intel manual এ পাওয়া যাবে এটার জন্য MOV হবে।

operation code

MOD 11 → R/M is a Register

Direction

D → 0 : REG → R/M

→ 1 : R/M → REG

MOV (REG), (R/M)

MOV AX, SP

W → 0 (move one byte of data) MOV AL, BL

→ 1 → word (16) AX, BX, CX

→ double word (32) EAX, EBX

Scanned with CamScanner

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

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

MOD, REG, R/M

- Convert binary to → Assembly

Immediate 0-4 bytes	Displacement 0-4 bytes	Scaled index 0-1 byte	MOD-REG-R/M 0-1 byte	Opcode 1-2 byte	Segment Override 0-1 byte	Size 0-1 byte
	8BECH					

Opcode						D	W
1	0	0	0	1	0	1	1

MOD		REG			R/M		
1	1	1	0	1	1	0	0

D=1

REG → Destination

R/M → Source

move AX, BX

MOV REG, R/M

MOD	Function
0 0	NO displacement MOV AL, [DI]
0 1	8-bit sign extended displacement MOV AL, [DI+2]
1 0	16 bit signed displacement MOV AL, [DI+1000H]
1 1	R/M is a register

MOD → 11

W

So, R/M is Register

MOV BP, SP

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

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

REG (when R/M = 11) R/m is a Register :

Code	W=0 (Byte)	W=1 (Word)	W=1 (Double Word)
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

REG
101

REG=BP

R/M
100

~~R/M=DS:[SI]~~

R/M=SP

When R/M ≠ 11

R/M code	Addressing mode 16 bit	32 bit
000	DS: [BX+SI]	DS: [EAX]
001	DS: [BX+DI]	DS: [ECX]
010	SS: [BP+SI]	DS: [EDX]
011	SS: [BP+DI]	DS: [EBX]
100	DS: [SI]	Scaled I bit
101	DS: [DI]	SS: [EBP]*
110	SS: [BP]*	DS: [ESI]
111	DS: [BX]	DS: [EDI]

EAX

[SI+4000]

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

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

Load effective address: LEA

→ Compute the effective address of 2nd operand (Source) & stores it in 1st operand (Destination)

→ Unlike mov instruction LEA stores computed address → target register

→ Stores contents (data) of target register

→ LEA BX, [DI] loads the offset address specified by [DI] → BX

→ MOV BX, [DI] loads data stored at memory location [DI] → BX

→ MOV BX, OFFSET LIST perform same function as LEA BX, LIST

* → OFFSET works with simple operands (LIST) & not with [DI], LIST [SI].....

→ OFFSET is more efficient (1 clk cycle) than LEA (2 clk cycle)

→ MOV BX, OFFSET LIST is faster because assembler calculate offset address of LIST. Microprocessor calculates address of LEA instruction.

Scanned with CamScanner

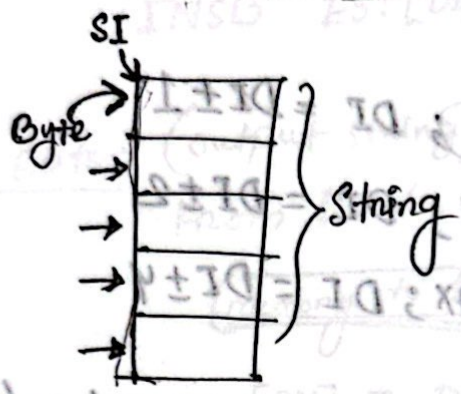
Sat / Sun / Mon / Tue / Wed / Thu / Fri

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

~~LODS~~ String Data Transfer Instruction :

Move AX, [DI] Byte, Word, Doubleword, Quad word

↳ deals with single data



DI & SI : Destination Index, Source Index } String

LODS, STOS, MOVS, INS, OUTS } string data

LODS:

loads accumulator (AL, AX, EAX, RAX) with the data stored at the data segment, offset address indexed by SI

LODSB AL = DS: [SI] ; SI = SI ± 1

LODSW AX = DS: [SI] ; SI = SI ± 2

LODSD EAX = DS: [SI] ; SI = SI ± 4

Scanned with CamScanner

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

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

STOS: (store)

→ Stores accumulator (AL, AX, EAX) at the extra segment memory location (addressed DI register)

STOSB ES:[DI] = AL ; DI = DI ± 1

STOSW ES:[DI] = AX ; DI = DI ± 2

STOSD ES:[DI] = EAX ; DI = DI ± 4

MOVS: (transfers string data from one memory location to other memory location.)

~~MOVSB~~ → data segment location addressed by SI

→ extra " " " DI

MOVSB ES:[DI] = DS:[SI] ; DI = DI ± 1 ; SI = SI ± 1

DS:[SI] = ES:[DI]

MOVSW

MOVSD

INS : (Input string)

transfer byte, word, double word from I/O Device to Extra segment.

INSB ES:[DI] = [DX]
I/O address

I/O to memory
hard disk to memory

OUTS : (output string)

From DS to I/O Device

memory to hard disk

OUTS [DX] = DS:[SI] ; SI = SI ± 1

Scanned with CamScanner

Chapter-06

Program Control Instructions

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

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

→ Program control – directs flow a program, change the flow.

Main Memory (RAM)
↓
Segment

- **Jmp**
  - unconditional Jump
    - Short Jump (2 Bytes)
    - Near Jump (3 bytes)
    - Far Jump (5 bytes)
  - conditional Jump

- **Relational Assembly Language**

  .IF, .ELSE, .ELSEIF, .WHILE, .ENDW
  .REPEAT, .UNTIL

Scanned with CamScanner

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

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

~~DMA~~

Microcontroller & Embedded System :

⇒ **Microcontroller** Vs **Microprocessor**

Microcontroller	Microprocessor
Ready made	large scale processing power
+	+
Small scale processing power (ছোট কাজ করা হয়)	Can customization
+	+
(memory ভিতরেই থাকে)	(memory আলাদা থাকে)
+	+
সুনির্দিষ্ট processing এর জন্য। + দাম কম	দাম বেশি

→ advantage, application

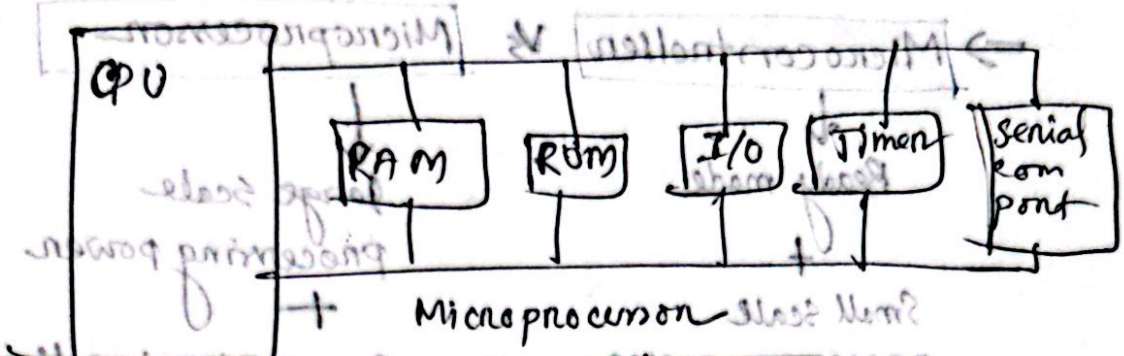
Embedded System:

System এর ভিতর কিছু set করে থাকা।

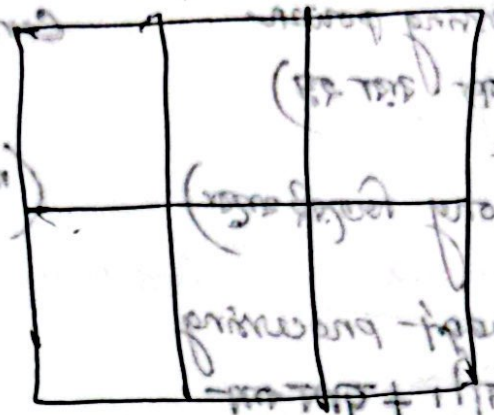
· specific কাজের জন্য specific system

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

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



Microprocessor



সব একসাথে থাকে।

Microcontroller

8051 Architecture তৈরি করবে

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

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

2018-19
2017-18
2016-17
2015-16
22
21
20
19
18
16
15
14
13

Chapter-02

Microprocessor and It's Architecture

8B9E004C

1000 10 11 (D W)

10 01 1110 0100 1100

MOD REG R/M

1684

W=1 (32 bit)

D=1
Source = M = ~~ESP~~ DS:ESI
Destination = REG = ~~DS:ESI~~ ~~ESP~~ ESP

MOV ESP, [ ]

8B

8B9E

W=1 (16 bit)

1000 10 1 1 (D W)

10 01 1 110

MOD REG R/M

MOV REG, M

D = 1
Source ⟹ Memory → DS: [SI]
Destination ⟹ REG → BX

MOV BX, [BP+4C00H]

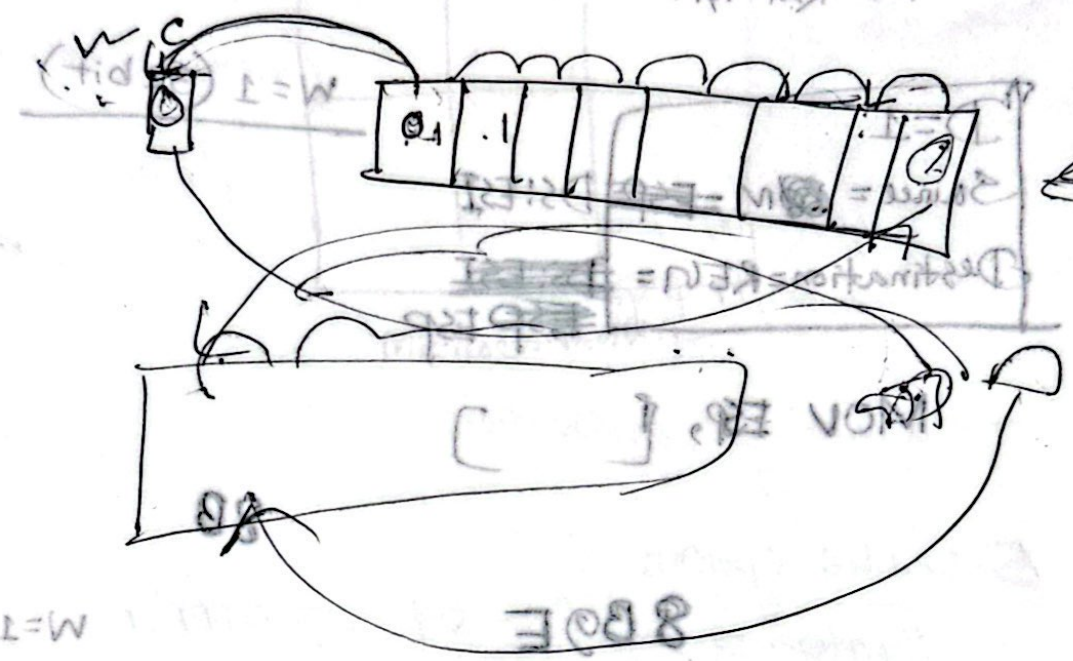
Sat / Sun / Mon / Tue / Wed / Thu / Fri

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

MOV SI, [BX+2]

10001011 01110111 00000010

8B77 02



DMA

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

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

- CPU do everything

→ processor কে কিছু সময় work free রাখার জন্য, যেখানে processing হয় না এমন controller থাকবে যা processor কে support করবে এমন controller হলো DMA controller.

Transfering Completely a bulk memory

- তাহলে processor অন্য purpose এর জন্য free থাকবে।
- DMA থাকলে processor এর improvement হবে।

{ internal memory → internal memory
{ external " → " "

- external part এ wait states থাকবে।

⇒ DSP56300 contains six DMA channel

→ DMA এর মধ্যে সব দেওয়া থাকবে যা processor deal করতো আগে। processor এর replacement এ নিজেকে ধারণ করিয়েছে।

Scanned with CamScanner

DMA count Register
└ Counting
higher dimension
দিয়ে define হবে।

গুলো cycle এ একটা word transfer করছে

Chinmoy Sir

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

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

- Evaluation of microprocessor
- 8086 and 8088 pin diagram
- 82C55 mode operations, mode 0,1,2

$\overline{BHE}$	A_0	
0	0	16 bit
0	1	(U) 8 bit
1	0	(L) 8 bit
1	1	Idle

data bus — 16 bit

address bus — 20 bit

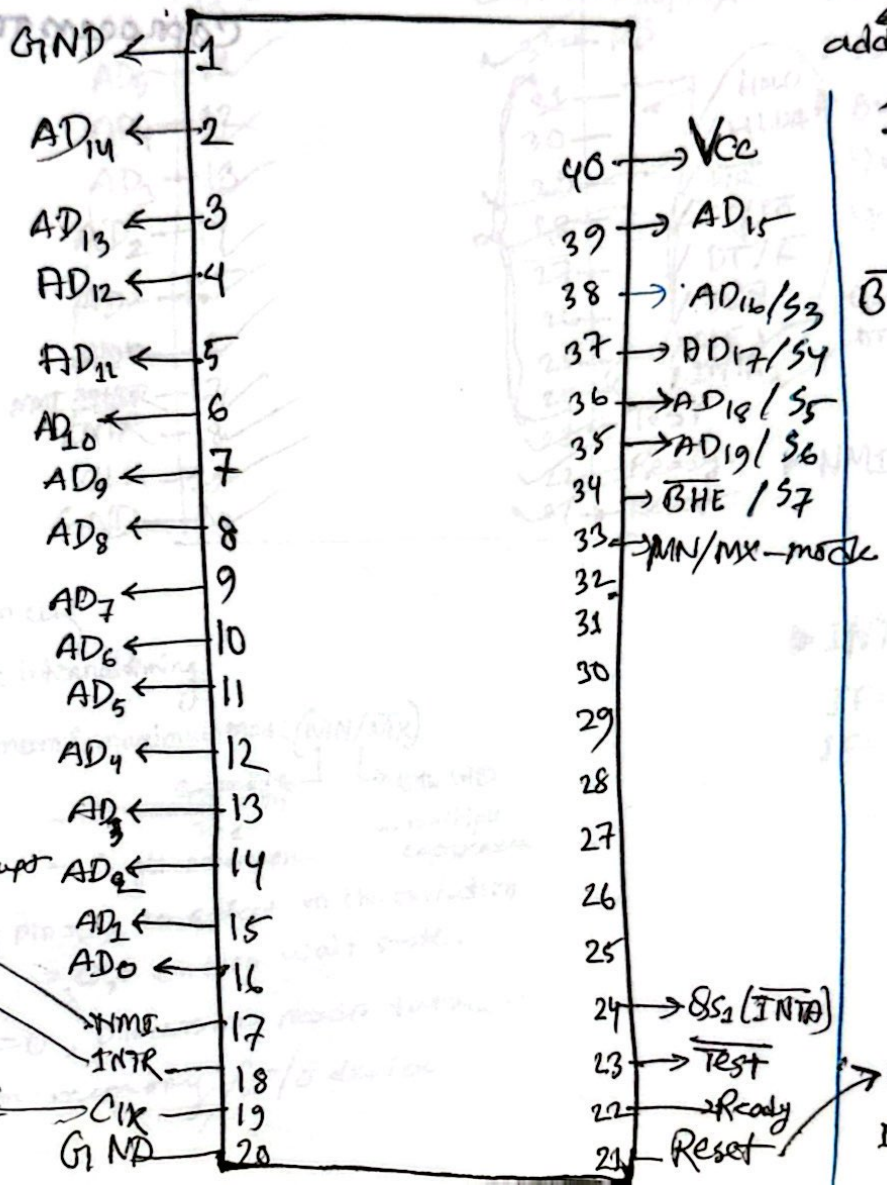
Pin diagram of 8086:

40 pin (DIP - Dual in Pin)

1'n up

(2-16) → multiplex ~~add~~ + data bus

address



(high Priority) non maskable interrupt

Interrupt

heart beat

16 bit ~~bus~~

So ($AD_0 - AD_{15}$)

$\overline{BHE}$ → Bus High Enable

Distinguish between low byte, high byte

CS: FFFF DS, ES, SS 0000

IP: 0000

queue empty

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

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

24 — 31 ⟹ Dual meaning

Pin 33 ⇒ select MIN/MX

Minimum mode

maximum mode

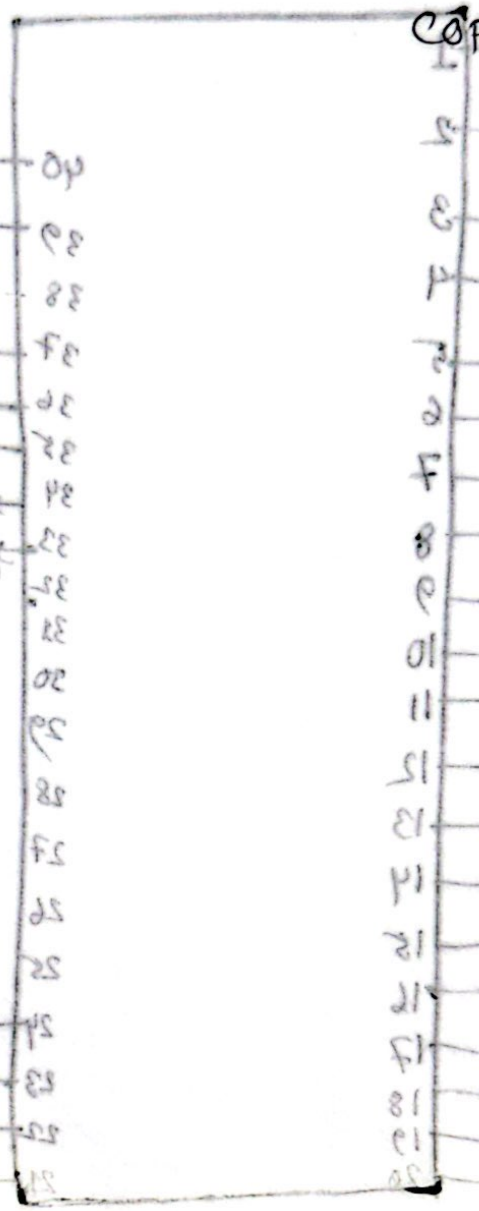
33 pin connected with Vcc

33 pin connected with ground.

8086 connected with coprocessor

ALE → Address Latch Enable

bar means active low



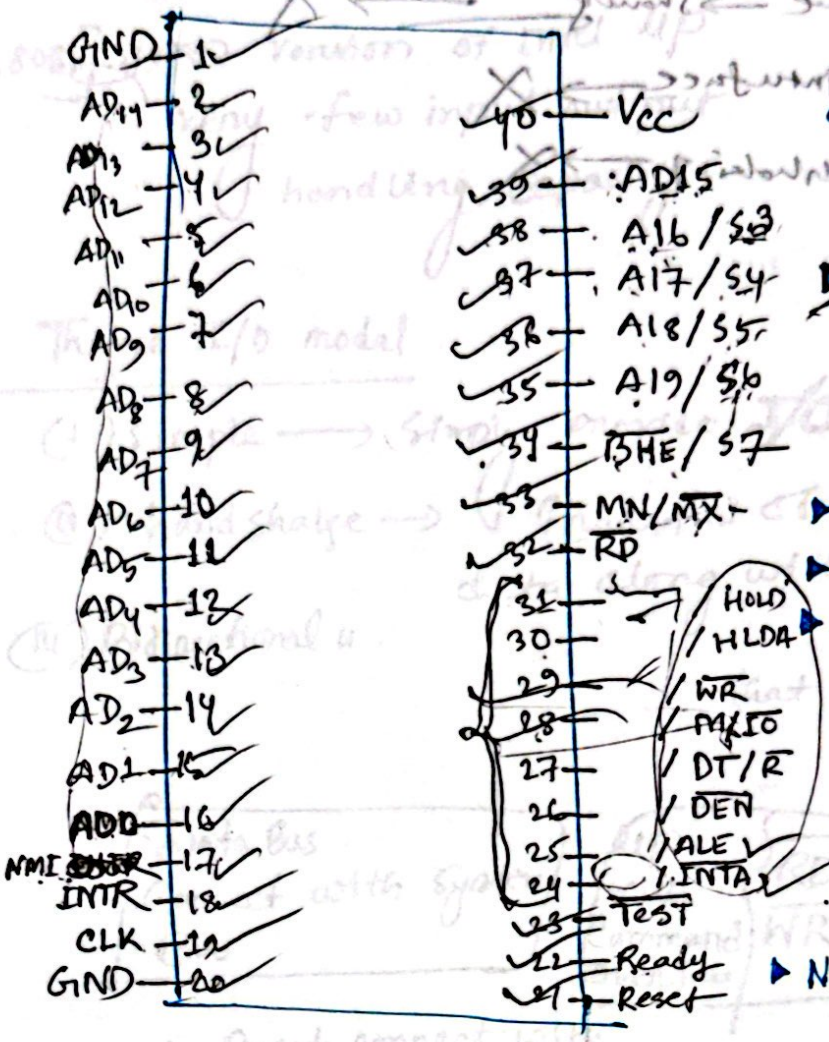
Sat / Sun / Mon / Tue / Wed / Thu / Fri

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

ALE
↳ Address Latch enable

active low

AD0-AD15



▶ Address DataBus [AD0-AD15]

- ALE = 1
  ↳ address bus [A0-A15]
- ALE = 0
  ↳ Data bus [D0-D15]

▶ Address Status Bus [A19/S6 - A16/S3]

- ALE = 1 → address
- ALE = 0 → status

▶ S3, S4 indicate segment

▶ S5 → Interrupt flag status

▶ Bus High enable (BHE)
↳ used by up for memory banking
↳ used to enable D15-D8 from D15-D0
Can transfer 16 bit only on one cycle (go memory banking)

▶ NMI (Non maskable interrupt) — high priority

▶ INTR (Interrupt Request)
IF = 1, INTR = 1, interrupt
IF = 0, INTR = 1, NO "

▶ System CLK
↳ give internal timing

▶ Minimum & maximum mode (MN/MX)
- Connected with Vcc — Single processor
- ↳ with GND — multiple coprocessor

▶ Ready pin = 1, no effect on the operation
" = 0, enters wait state.

▶ RD = 0, processor reads data from memory/I/O device

Scanned with CamScanner

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

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

3 → Assign → X

Hardware → Shared → X

I/O interface → X

Handshaking → X

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

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

PPI (8255)

(~~8085~~, 8086) পুরাতন version of Intel μP
very few input-output
handling capacity.

Three I/O model

(i) Simple → Simply provide I/O

(ii) hand shake → Transfer of data along with handshake initialization that will be done.

(iii) Bidirectional u.

Data Bus Connect with System Bus	μP will Command over here $\overline{RD}$ $\overline{WR}$	A0, A1 select Port A, B, C

→ Reset connect with μP Reset out pin.

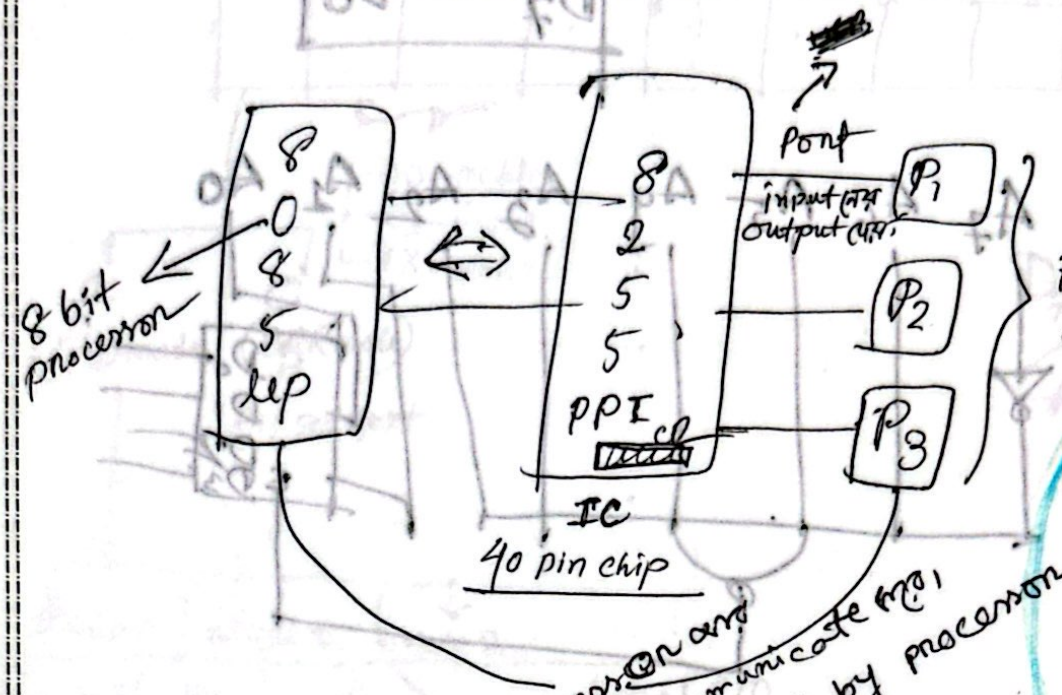
PPI

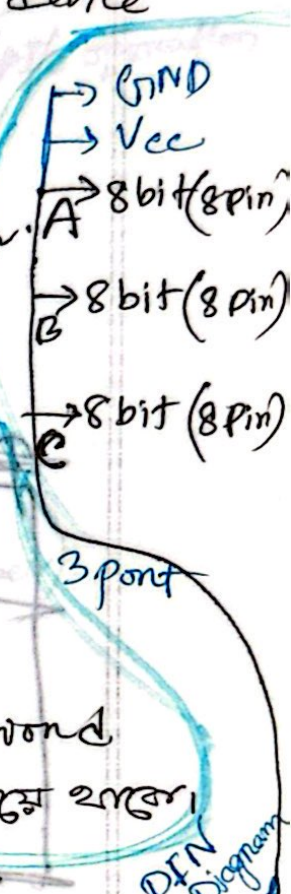
Sat / Sun / Mon / Tue / Wed / Thu / Fri

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

Programmable peripheral Interface

8255 PPI





→ Communicate করার জন্য 8085 Control word পাঠায় 8255 কে।

↓

8255 এর control register এ control word Store হয়। → এই অনুযায়ী I/O function হয়ে থাকে।

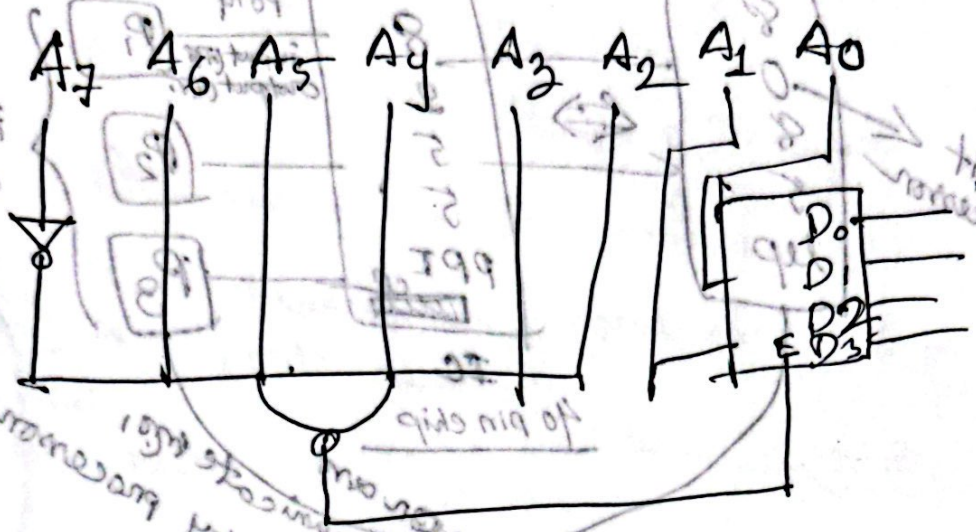
Control Info ($D_0 - D_7$) ← $\overline{RD}$ $\overline{WR}$ Reset $\overline{CS}$ A_1 A_0 decoder

$\overline{CS}$ → Chief select দিয়ে 8255 Select হয়।

→ A0, A1, Particular port Selection এর জন্য কাজ করে।

→ Decoder enable করতে Chief select কাজ করে

8085 → (A_{15} — A_8 | A_7 — A_0 / D_7 — D_0)



~~port A~~

A_1	A_0	port
0	0	Port A
0	1	port B
1	0	port C
1	1	Control Register / Command Resigter

Handshaking → Synchronization Communication
Synchronize करना है।

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

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

Control word:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

- 00 – mode 0
- 01 – mode 1
- 1X – mode 2

$D_7 = 1/0 \rightarrow$
→ BSR
→ I/O operation

1 (I/O) mode

0 (BSR mode) → Bit set Reset

Mode 0 | Mode 1 | Mode 2

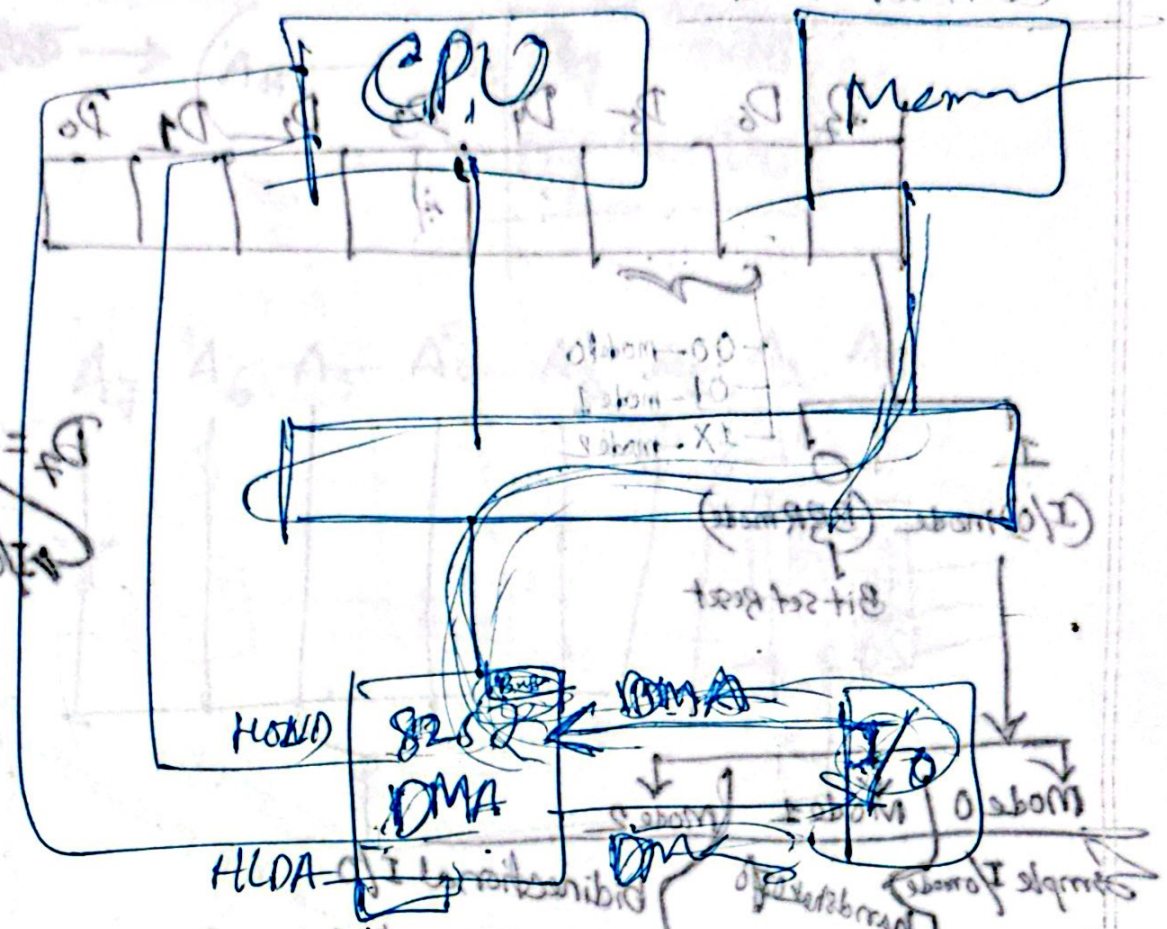
Simple I/O mode
Port A, B, C

handshake I/O
A, B
port C used for handshake
Upper → A
Lower → B

bidirectional I/O
- port B inactive
- port C used as handshake (Port Upper)

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

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



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

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

cmp AX, BX

0 - AX = BX

5 - AX > BX

(-5) (AX < BX)

1

cmp AX, BX → AX − BX

8 4 2 1

(5) AX > BX

10 → 1010

8 → 1000

0000

ADD AX, DX

1000 (CF) = Carry

1010

1 1110

INC SP

1001 →

ADC DX, BX

0001 0000 0000 0001

0010 0000 1111 1111

0011 0001 0000 0000

INC BYTE PTR [BX]

C = 0

A = 1

S = 0

O = 0

Z = 0

P = 0

ADD DH, DL

ADD DH, BL

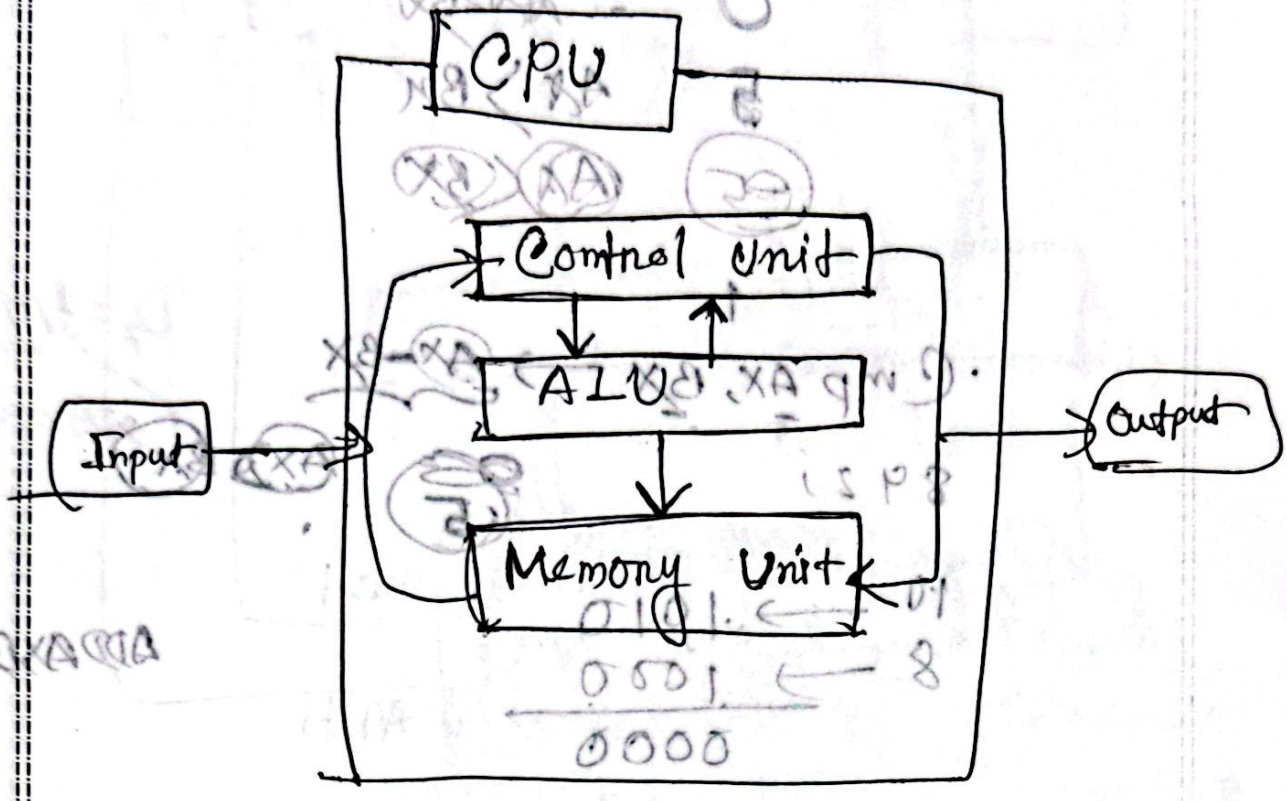
ADD DH, CL

ADD DH, DL

~~ADD~~ DH, DH

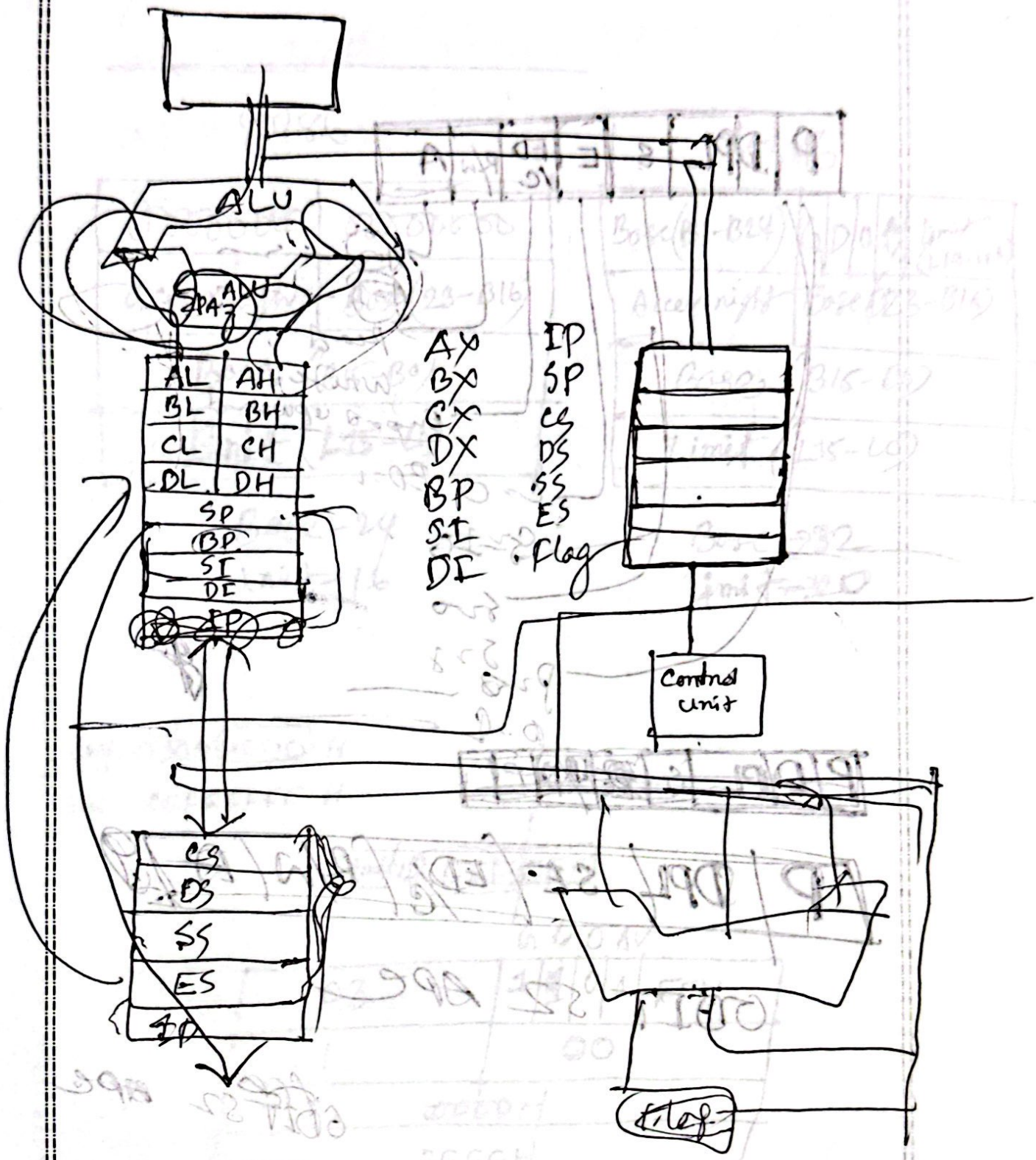
Sat / Sun / Mon / Tue / Wed / Thu / Fri

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



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

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



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

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

P	DPL	S	E	ED/C	R/W	A

A=0 —

A=1 —

Read

write

Expand upward

E=0 ED=1

S=1 C

S=0

S=1

P=0 —

P=1 —

P	DPL	S	E	ED/C	R/W	A

P	DPL	SE	ED/C	R/W	A

GDIT S2 APC

GDIT S2 APC

~~P DPL SE ED/C R/W A~~

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

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

80286

00000000	00000000
Access rights	Base (23-B16)
Base (B15-B0)	
Limit (L15-L0)	

Base-24
Limit-16

80386

Base (B31-B24)	G	D	0	AV	Limit (L19-L16)
Access rights	Base (B23-B16)				
Base (B15-B0)					
Limit (L15-L0)					

Base → 32
Limit → 20

Start: 03000000 H

End: 05FFFFFF H

Limit: 2FFFFFF

	G	D	0	AV	
03	1	1	0	1	FFF
1	00				
0000H					
2FFFH					

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

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

8086 & 8088	———	8087
80286	———	80287, 80287XL
80386 DX	———	80287, 80387 DX
80386 SX	———	80387 SX
80486 DX	———	~~80487~~ In built
80486 SX	———	80487 SX

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

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

Ame the address of the next instruction

1111 1111
0000 0001
10000 0000

0000 0000 0010 0000

RPL
0011

0001

4KB

03000000

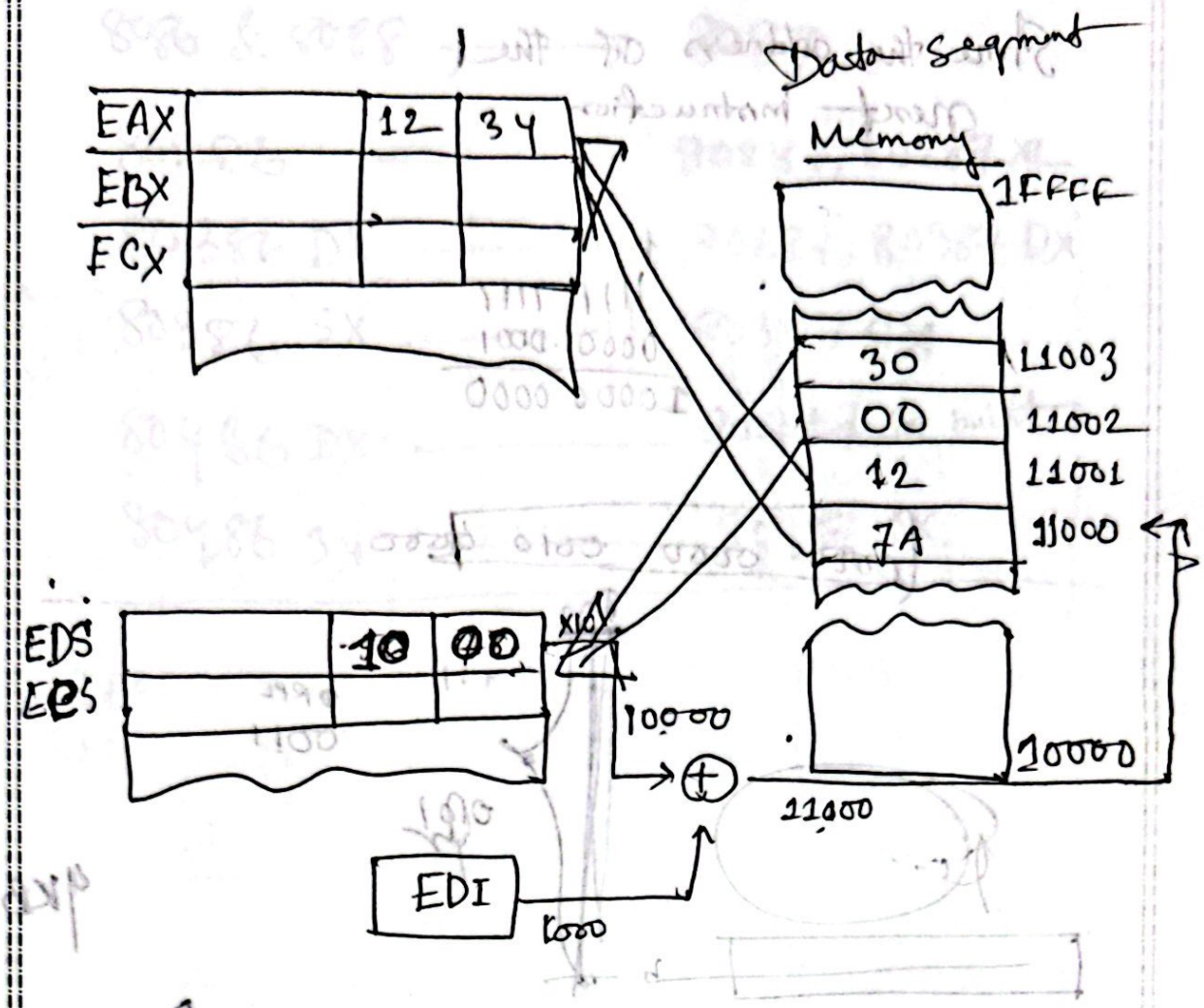
G=0

Limit

00000

FFFFF

Base


Data segment
Memory
EAX
12
34
EBX
ECX
1FFFF
30
00
12
7A
11003
11002
11001
11000
EDS
10
00
x10
10000
10000
11000
EDI
1000

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

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

Sorna Maam (6th class)

TCP/IP — Forouzan

Chapter-23 ⟶

Electronic Mail

Architecture, agent

ASCII ⟶

Architecture

Four scenarios :

1st :

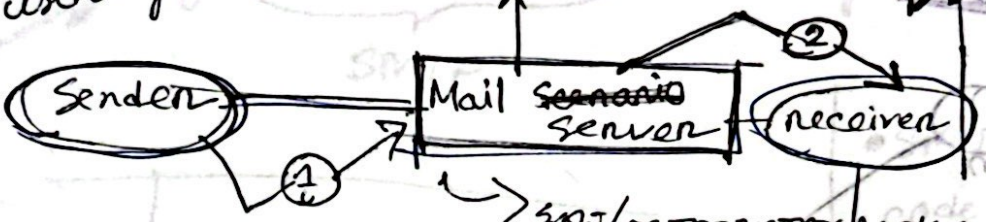
Application layer এ এই application গুলো রয়েছে। 2 ধরনের architectural paradigm গুলোর কাজ করে।

→ client server based.

→ peer to peer

• only user agent

• Sender, receiver এর আলাদা আলাদা folder থাকে।



কুলে/কলেজ আলাদা locker থেকার system

→ directly connected

• 1 টা mail server

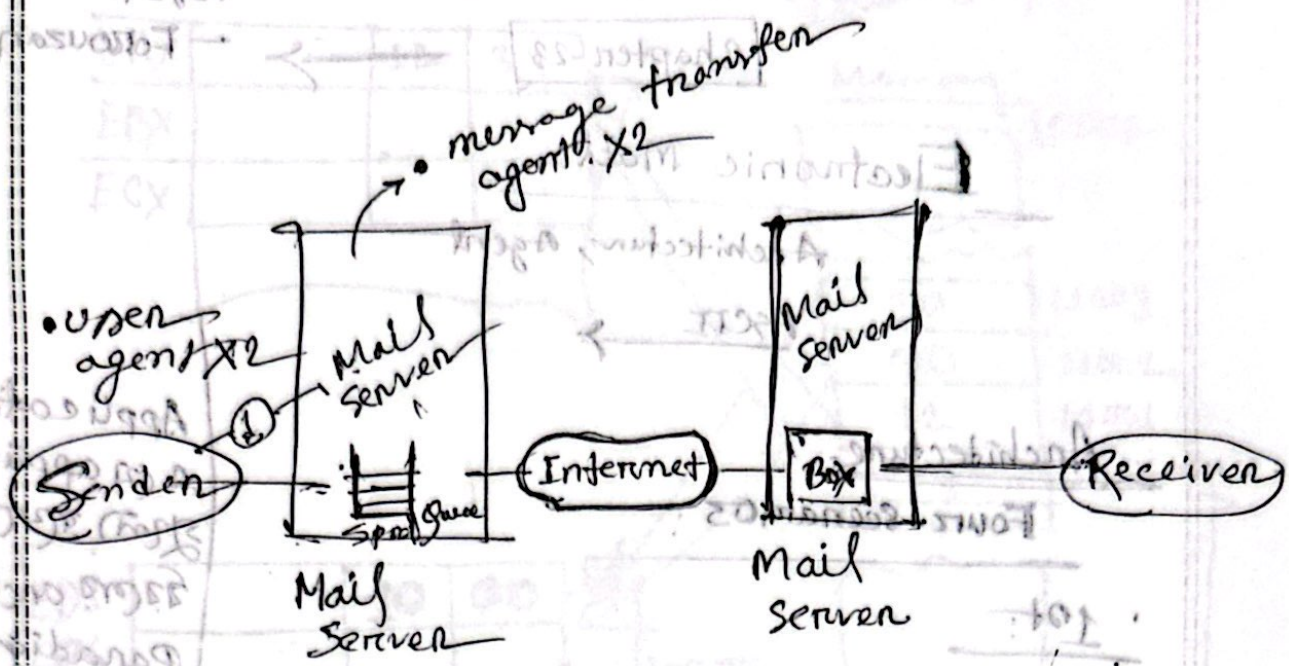
• receiver এর read করতে হলে এই software package থাকতে হবে।

• When the sender and the receiver of an e-mail on the same email server, we need only two user agents.

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

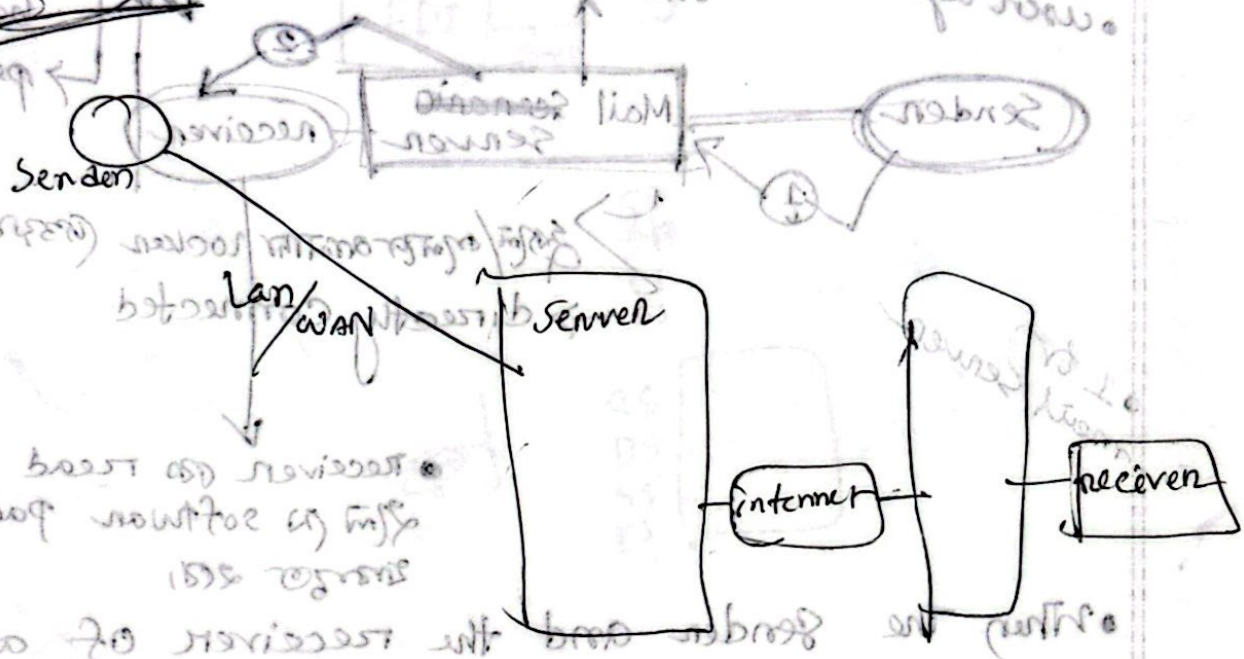
Date :...../...../.....

2nd : str mail server



13 min পরে

3rd :

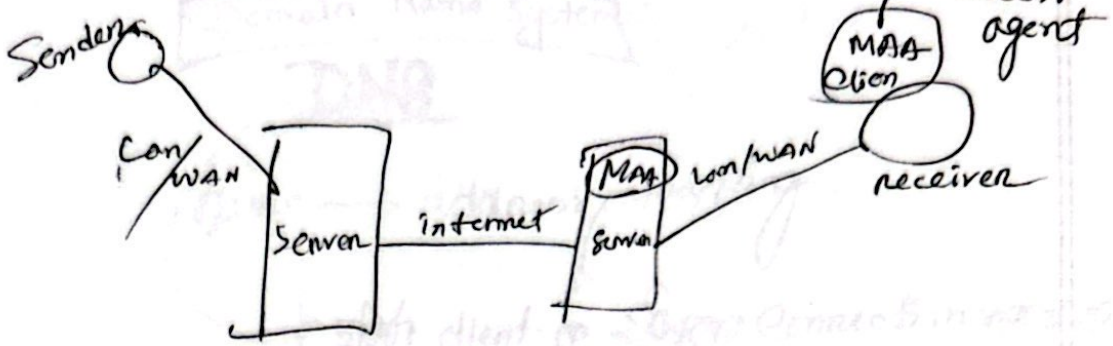


Sat / Sun / Mon / Tue / Wed / Thu / Fri

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

4th → today's common situation.

HTTP (80)



23 min পর্যন্ত

Agents,

Outlook দিয়ে email send করা যায়।
↳ এটাই user agent

access
↓
pulling protocol
POP3
↓
Post office protocol → Email format

IMAP4 → Message transfer agent → mainly কাজ করে।
SMTP

SMTP
↓
Simple mail transfer protocol

→ Connection establishment
↳ 3টির step
→ message transfer
↳ 6টির step.
→ Connection termination
↳ 2টির step.

• SMTP code number 25

POP3 (110)

Black → command
Colon → Response msg

IMAP ⇒ spam check অধিক করে

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

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

Chapter-19

Domain Name System

DNS

Name → address } mapping

→ প্রতিটা client কে ২ ধরনের Connection করতে হয়।

Flat name Space

restriction নাই

Central control করতে হবে, (duplicity থাকে না হয়)

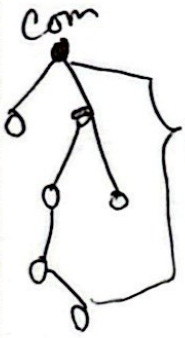
Hieronchical Name space

এটা ব্যবহার হয়।

Level → 128 পর্যন্ত হতে পারে। (127 (.)Dot দিতে পারি) PSTU.ac.bd

Label → maximum 63 character পর্যন্ত থাকতে পারে.

(PSTU) → Lable



Com এর domain → কতটুকু area এরজন্য Com responsible.

FQDN PQDN

Zone → কোন কোন server এর জন্য authority হিসেবে কাজ করবে,

Sat / Sun / Mon / Tue / Wed / ...

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

pstu. ac. bd. (root) → FQDN

→ প্রত্যেকটা node এর জন্য আলাদা আলাদা server থাকবে।

→ node delete, add secondary server করতে পারবে না।

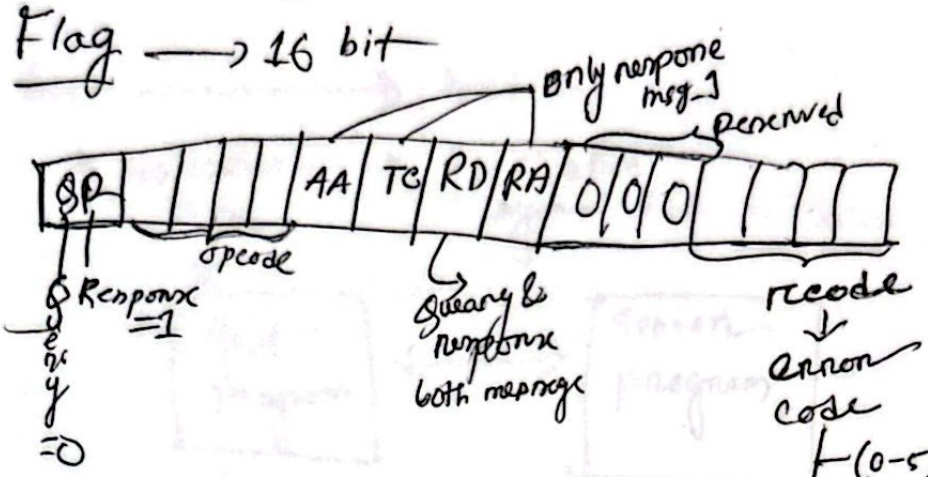
Primary → update করতে পারবে।

inverse domain → address কে name এ convert

- → variable length
- header → 12 byte

identification
↳ identify the response against query,

Flag → 16 bit



Standard → generic/country domain
inverse → query ⊙ IP দিয়ে response এ domain name বের করি,

Answer
Authoritative
Additional
} Resource Record

Question section } question record.

34 min

Sat / Sun / Mon / Tue / Wed / Thu

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

- question record → query name
- ~~Dom~~ Resource Record → Domain name

} Same पिक्चर

Chapter – 02

Application layer

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

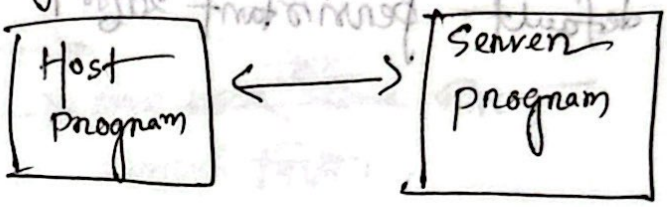
Date :...../...../.....

Physical layer
Data link layer
Network layer
Application ⟶ Top most layer
Transport "

খরচ বেশি) Client-server
Security নাই) Peer to peer

host ⟶ host
▸ application layer ⟶ ▸ application layer

Google – 100 data centre



Socket → interface
(দরজা)
↳ যা দিয়ে send হবে।
↳ যা " receive "।

Throughput ?

- data integrity
- timing
- throughput ⟶ 32 KbPS নিশ্চিত হবে।
  - বেশি দিলে Buffer এ waiting এ থাকবে।
- Security

transport service layer for application layer

Senden →

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

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

TLS

Transfer Layer Security
+
enhacement of TCP

} security provide করতে পারে।

HTTP protocol :

→ used in web application।

⇒ by default persistant হয়।

HTTP request message

Request line → GET /somedir/page.html HTTP/1.1 ——→ ↵

(method: GET; URL: /somedir/page.html; HTTP version: HTTP/1.1)

Header line:
- HOST: www.someschool.edu
- Connection: close
- User-agent: Mozilla/5.0
- Accept language: fr

Carriage return
Line feed

Get method →

post method → কোন ~~book name~~ কিছু লিখে search দিলে।

Head " → কোন object পাঠাবে না, output পাঠাবে।

put " → কোন কিছু upload করলে

Delete " →

এই দুই ক্ষেত্রে entity body থাকে।

Scanned with CamScanner

(TCP/IP)

Sobuz Sir

Internet layer

OSI

- presentation
- Session

Application → user

Transport

Network

Data link

Physical → Transmission medium

software

hardware

● OSI model এর কাজ।

→ History of Internet

logical address*

IP address.

83
27.75
109.75

Question কমন আছে

Application Layer → Application specific address

Transport " → port address

Network layer → Logical address (IP)

Data link " → Physical address. (MAC)

cmd
↓
ipconfig/all

Scanned with CamScanner

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

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

CRC → Cyclic Redundancy Check

(repeater)

⇒ HUB, switch,

↓ সবার কাছে পাঠায়।

↓ specific পাঠায় (mac address)

bridge, Router

Ch-04

- Network layer

313-314

29:41

36:16

Sat / Sun / Mon / Tue / Wed / Thu / Fri
Date : ...../...../.....

Sobuz Sir

1. Chapter 1 (Kurose Exercise) → 1 set
2. chapter-5 (book + exercise) — 1.5 set

0.5 set → (1-4), ~~Bol Algo~~

Chap-01

pc, workstation, web server, mail server

host: is a device that connects to other devices in a network.

end system: is a device that is directly connected to the internet.

Scanned with CamScanner

Access technology

Sat / Sun / Mon / Tue / Wed / Thu / Fri
Date :...../...../.....

Protocol:

A protocol is a set of rules that commands the data communication.

DSL (Digital Subscriber line)

dial up modem
DSL
cable modem
fibre-to the home (FTTH) → use optic fibre provide internet access to residential & commercial
Cable to HFC (Hybrid Fiber coax)

802.11

Ethernet
WIFI
3G, 4G } wireless internet access

Sat / Sun / Mon / Tue / Wed / Th / Fri

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

2/19, 23 13, 14, 15

30+15+10+12

30

30
10
15
15

IPV4 — 32 bit

1111

2^{32} → IPV6

192.168.0.1

192

121

0 A 31

10 B 30

110 C 29

1110 D 28

1111 E 28

256

0–127 class A

128–191

2^{8-1}

2^7

2^7

-128

2^7

$= 128$

26

Net id 127

Net id

0.0.0.0

0.255.255.255

127.0.0.0

127.255.255.255

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

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

0

Net id · 192.0.0

192.0.0.0

192.0.0.255

110

Net id → 192.255.255.

223.255.255.

223.255.255.0

223.255.255.255

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

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

127 - 192

224.255.255.255

A → 0
B → 10
C → 110
D → 1110
E → 1111

Class - A

0 | 0 0 0 0 0 0 0 | → (0 - 127)

Net id 0 → Net id 1 → Net id 127

0.0.0.0 — 0.255.255.255

1.0.0.0 — 1.255.255.255

127.0.0.0 — 127.255.255.255

Class - B

10 | 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0

Net id: 128.0 → Net id: 128.1 → ~~Net id 128~~ Net id: 191.255

128.0.0.0 — 128.0.255.255

128.1.0.0 — 128.1.255.255

191.255.0.0 — 191.255.255.255

Class - C

192.0.0 → 192.0.1 → 223.255.255.0

192.0.0.0 — 192.0.0.255

192.0.1.0 — 192.0.1.255

223.255.255.255

110

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

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

186

2^{24}

73.22.17.25

73.0.0.0

.1 - .254

73.255.255.255

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

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

141.14.0.0/16

B

$n_{sub} = n + \log_2 S$

$= 16 + \log_2 2^2$

$= 16 + 2$

$= 18$

1 | 2^31

30 2^2

2^0 32

32

1 - 2

32 . 2^0

141.14.0.1 → 141.14.63.254

141.14.64.1 → 141.14.127.254

141.14.0.0/18

141.14.64.0/18

128 64 32 16 8 4 2 1

1 1 1 1 1 1 1 1

160

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

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

130.34.12.64/26

$\frac{26}{4}$

$n_{sub} = n + 4$

$= 26 + \log_2 4$

$= 26 + 2$

$= 28$

From slide - 61

P2P application

- File sharing
- Torrent → popular (bit torrent)

peer to peer (P2P)

যেখানে client নির্দিষ্ট device/server এর উপর dependend না।

→ যেখানে প্রত্যেক device server হিসেবে আবার client হিসেবে কাজ করে।

→ আমার কাছে কোন File থাকলে অন্য সবাই জানবে। এবং তারা নিতে পারবে File টা। অন্য কারেকটা device File টা নিলে তখন ২ টা device সেই File এর জন্য server হিসেবে কাজ করবে। এভাবে server এর সংখ্যা বাড়ে।

Self scalability:

new peers bring new service capacity and new service demands.

→ Server এর কাজ ভাগ হয়ে যায়।

Scanned with CamScanner

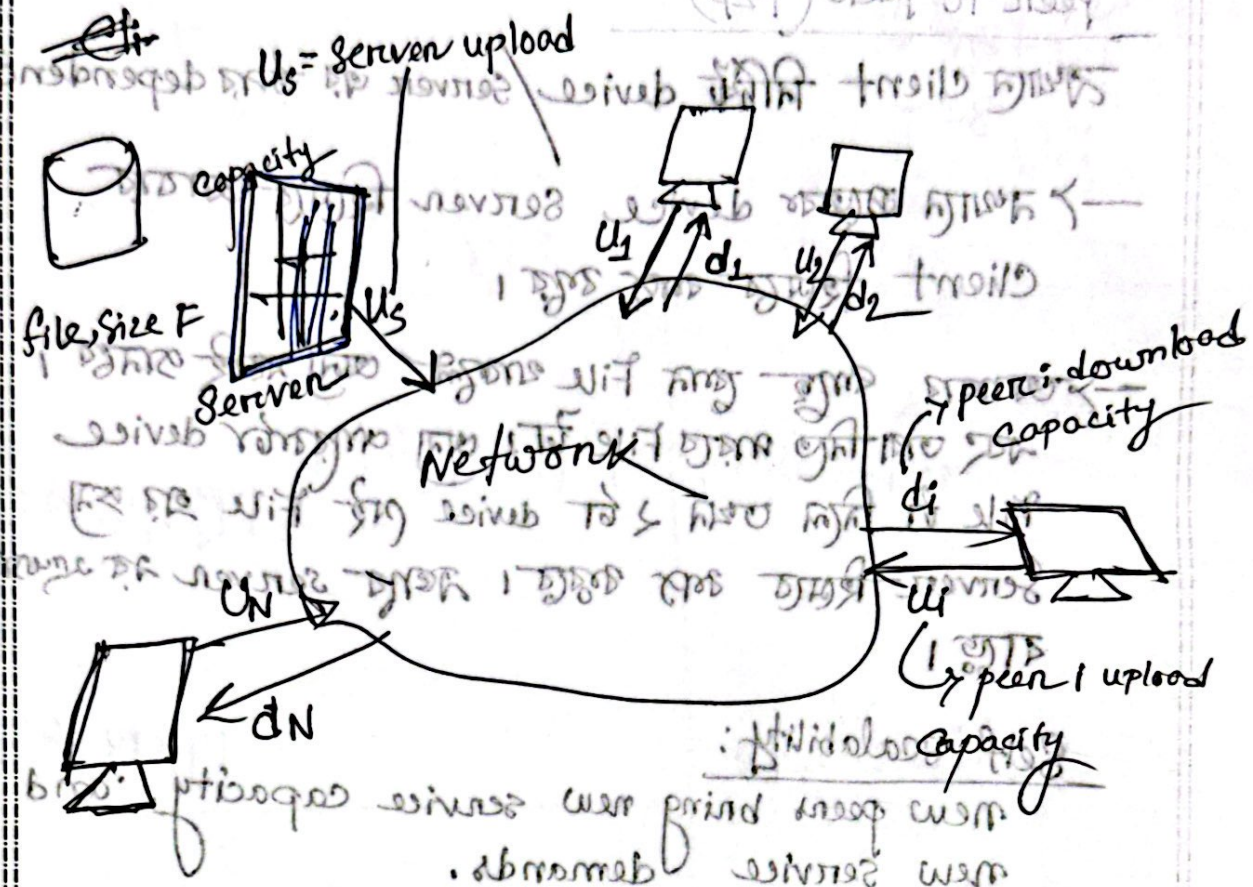
2.5

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

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

abundant → প্রচুর

▸ how much time distribute File (size F) from one server to N peers?



Client-server Distribution time, (D_{cs}):

- $D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}$

$NF \rightarrow$ N সংখ্যক peer এর কাছে আছে F bit File

$u_s \rightarrow$ Server upload rate

▸ Needed time $\left(\frac{NF}{u_s}\right) \rightarrow$ Theoritically

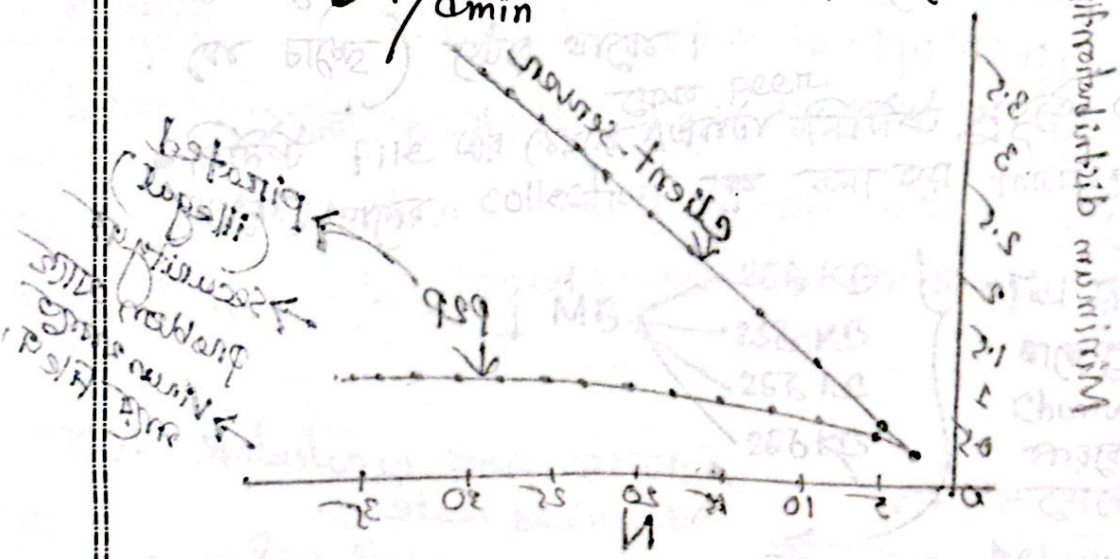
▸ বাস্তবক্ষেত্রে সময় এর চেয়ে অনেক বেশি লেগে থাকে

$d_{min} \rightarrow$ minimum download rate $(d_1 - d_n)$

▸ Server ঠিকই upload করছে কিন্তু download হতে আরো বেশি সময় লাগে। যতক্ষণ না download হবে ততক্ষণ server অন্যের কাছে File পৌঁছাতে পারবে না।

তাই দুটার মধ্যে maximum টাইমনিবো।

- $F/d_{min} \rightarrow$ কম সময় যার লাগছে তার প্রয়োজনীয় সময়।



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

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

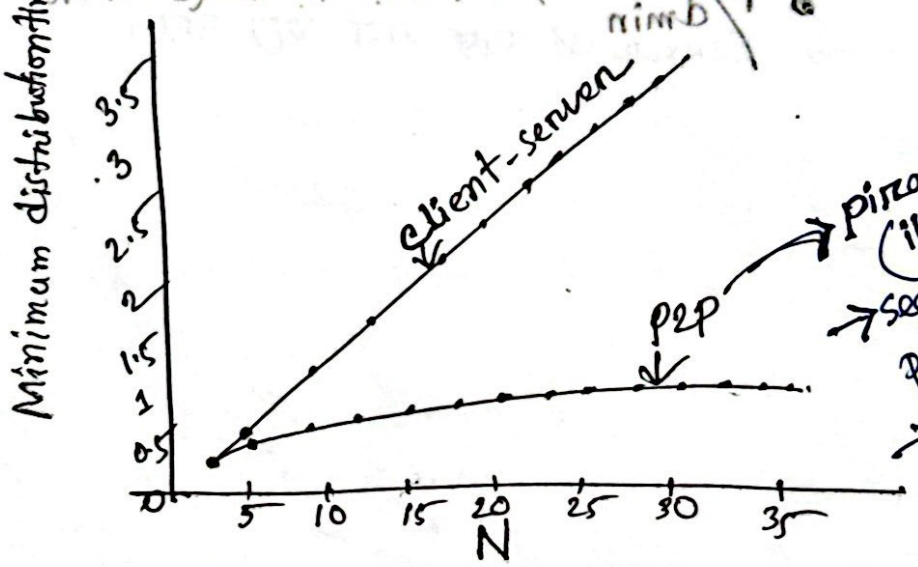
P2P Distribution Time (D_{P2P}):

$$D_{P2P} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

→ একটা peer main server থেকে file নিতে সময়।

▸ পুরুষাত্র প্রথমবারে একটা server থেকে কোন একটা peer কোন file download করলে একটা সময় লাগবে (client-server এর মতো). download এর পর যখন সবাইকে লোড করে নিবে তখন peer গুলোর মধ্যে File distribute time কম লাগবে।

$$D_{P2P} = \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$



→ Pirated (illegal)
→ security এর problem আছে
→ Virus থাকতে পারে File এ।

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

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

How Bit Torrent works?

Bit Torrent Protocol → P2P File Distribution

→ File divided 256 KB chunks ← (1)

→ peers in torrent send/receive file chunks. ← (2)

↓

main server থেকে আগে কোন এক peer কে download করতে হয়।

↓ • peer কে torrent ব্যবহার করতে হবে।

↓

Torrent:

group of peers exchanging chunks of a file.

tracker:

tracks peers participating in torrent. (information add হয়)

▶ নতুন user file খোঁজার জন্য যখন search করবে তখন tracker এর database এ সে add হবে। যা search করবে সে related একটা table/File থাকে database এ। যেখানে সে File সম্পর্কিত তথ্য (কার কাছে আছে, কে কে চাচ্ছে) স্টোর থাকে।

• নির্দিষ্ট File এর ক্ষেত্রে, যেসব peer একটা নির্দিষ্ট মুহূর্তে connected থাকে, তাদের collection কে বলা হয় torrent.

1 MB → 256 KB, 256 KB, 256 KB, 256 KB

• Full File এক জনের কাছে থেকে না নিয়ে ও chunk wise নিতে পারবে বিভিন্ন জনের থেকে।

▶ Chunk wise দিতে পারবে।

→ একই সাথে server এবং client হিসেবে কাজ করবে।

Scanned with CamScanner

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

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

▸ peer এর দুটো Decision নিতে হয়:

1 → কার সাথে connect হবো। কার TCP connection accept করবো। (কাউকে দেওয়ার জন্য)

2 → কার কাছ থেকে download করবে। কোন chunk আগে download করবে।

(rarest chunk সবার আগে download করবে)

↓

কম সংখ্যক chunk যাতে বৃদ্ধি পায়। → Rarest First

~~(almost events~~

tit-for-tat:

যদি কোন কিছু download করো তোমাকে upload ও করতে হবে।

• সেরাগুলো ৪ জনকে upload করবে।

• যার download rate বেশি তাকে upload দেয় দিবে।

• যাদেরকে upload করছে এমন নির্দিষ্ট মুহূর্তে তাদেরকে বলে unchoked দিয়ার। → বাকিদের বলে choked

• কিছু second পর পর download rate check করে unchoke। peer change করে।

10 second

• কম bit rate বাদ দেওয়া হবে। → • 10 sec পরপর recalculate করবে download rate.

30 sec পর update করবে।

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

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

Content Distribution Network

Video streaming & CDN:

→ Bit torrent যেখানে আমরা File download করি।

→ Video streaming এ download complete হওয়ার আগে আমরা সেটা play করতে পারবো। (online এ movie দেখা) — Netflix, Youtube, Amazon

একইসাথে play হয় এবং পরবর্তী chunk download হয়

Challenge → heterogeneity (আমাদের device এর bandwidth, specification, bit rate ভিন্ন ভিন্ন) etc.

Solution → distributed, application-level infrastructure

কেউ Samsung, কেউ Apple

Video: → sequence of images

~~24/s~~ 24 images/sec

↓

frame

Image: array of pixels

↳ represent by bits → per pixel এ 24 bit লাগে। (ভিন্ন ভিন্ন ও হয়)

- 8 bit Red
- 8 " Blue
- 8 " Green

• ৩টা color দিয়ে সমস্ত color কে represent করতে পারি

▶ ছবিতে pixel একই color hold করলে, ধরি, পরপর 100 টা pixel same color

100×24 = 2400 bit লাগবে।

সেক্ষেত্রে, purple color হলে purple এর 24 bit combination এর জন্য bit লিখে (,) দিয়ে 100 লিখবো। 100 টা লিখতে 8 bit ব্যবহার করি।

Spatial (within image)

↓

common নেওয়া

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

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

Temporal (from one image to next) → common দেওয়া

Sending only difference from frame i to frame i+1.

- Constant bit rate (CBR) :— video encoding rate fixed
- Variable bit rat (VBR) — encoding rate changes as spatial, encoding changes

EX: MPEG — 1.5 Mbps
MPEG 2 — 3-6 Mbps

Streaming Videos :

100 Kbps ⟶ 12 Mbps

→ Mbps বেশি হলে storage capacity ও বেশি হতে হয়।

67 min → encoding 2 Mbps

1 GB storage লাগবে

Challenge :

→ server-client bandwidth vary over time.
→ same ও হতে পারে

→ packet loss, delay due to congestion will delay payout, or result in poor video quality.

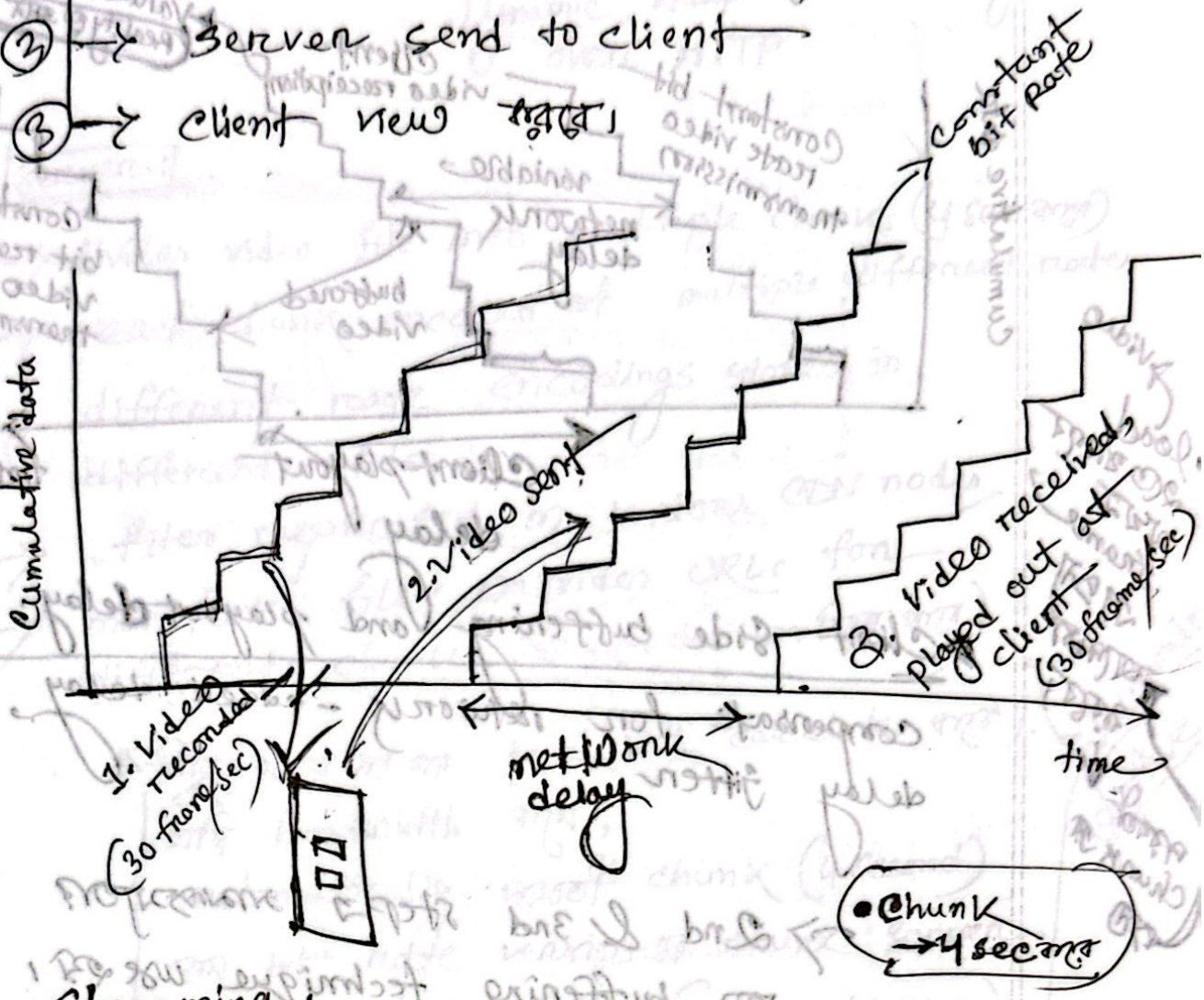
Sat / Sun / Mon / Tue / Wed / Thu / Fri

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

① → First record video & stored video on server

② → Server send to client

③ → client view করবে।



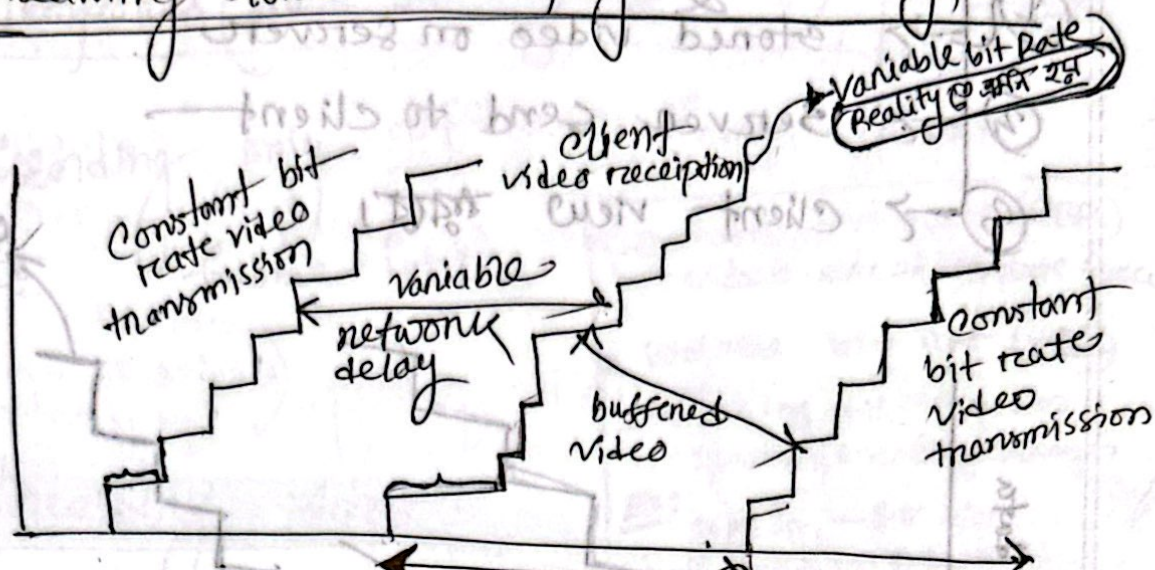
Streaming:

at this time, client playing out early part of video, while server still sending later part of video.

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

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

Streaming stored video: playout buffering:



• load হতে থাকবে যখন frame এর জন্য অপেক্ষা করবে পরবর্তী chunk এর জন্য

Client side buffering and playout delay:

compensate for network-added delay
delay jitter

→ 2nd & 3rd step এ সমস্যাগুলোর জন্য buffering technique use হয়।

• যে সময়ে আসার কথা সে সময় না এসে অন্য সময় আসাকে বলে Jitter.

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

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

Streaming Multimedia: DASH: protocol

↓

Dynamic, Adaptive Streaming over HTTP.

Server:

→ divides video file into multiple chunks (4 sec করে)

→ each chunk encoded at multiple different rates

→ different rate encodings stored in different files

→ files replicated in various CDN nodes

→ manifest file: provides URLs for different chunks. (এর সাথে decision নিতে পারে)

360
480
720
1080
• per chunk

- কিছু সময় পর পর bandwidth calculate করে।
  যদি bandwidth কমে,
  automatically পরবর্তী chunk (4 second)
  কম bit rate version এর chunk server থেকে client এ আনে। (vice-versa)

Dynamic Adaptive

Client:

→ periodically estimates server to client bandwidth.

→ Consulting manifest, request one chunk at a time
- → chooses maximum coding rate sustainable given current bandwidth
- → can choose different coding rates at different points in time (depending on available bandwidth at time), and from different servers.

"Intelligence" at client:

Client determines
- → when to request chunk (so that buffer starvation, or overflow does not occur)
- → what encoding rate to request (higher quality when more bandwidth available)

Scanned with CamScanner

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

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

→ where to request chunk (: can request from URL server that is close to client or has high available bandwidth)

Streaming video = encoding + DASH + playout buffering

Content Distribution Networks (CDNs):

Challenge:

How to stream content (selected from millions of videos) to hundreds of thousands of simultaneous users?

→ একটা নির্দিষ্ট server হলে possible হয় না।

⇒ ~~একটা নির্দি~~ simultaneously একটা নির্দিষ্ট threshold value এর বেশি হলে server crash করে।

⇒ **option 1:**

Single, large "mega-server"

- single point of failure.
- point of network congestion.
- long (and possibly congested) path to distant clients.

~~quite simply: this solution doesn't scale~~.

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

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

Option-02:

Store / serve multiple copies of videos at multiple geographical distributed sites (CDN)

enter deep:

push CDN servers deep into many access networks.

→ close to users

→ **Akami:** 2,40,000 servers deployed in 120 countries (2015)

bring home:

Smaller number (10's) of larger clusters in POPs near access nets.

→ used by limelight.

Sat / Sun / Mon / Tue / Wed / Thu / Fri
Date :...../...../.....

How does Netflix work?

→ Stores copies of content (MADMEN) (web series) at its (worldwide) openConnect CDN nodes.

→ Subscriber requests content, service provider returns manifest,

→ use manifest client retrieves content at highest supportable rate.

▶ 3 component এবং

- → Netflix website (Amazon Cloude) → DASH mechanism এর
- → private: CDN Infrastructure (Paid)

CDN private

▶ studio হতে movie কিনে নেয় netflix.

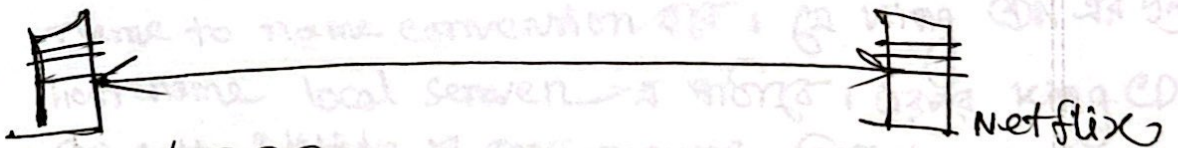
↳ Then Amazon cloude এ (website) upload করে দেয়।

↳ সেখানে নির্দিষ্ট Video এর Several version (DASH) তৈরি হয়। যাতে different device (Desktop, Smartphone) হতে চালাতে পারে।

↳ এরপর বিভিন্ন version CDN এ upload করে দেয়। → Server, Server rack

↳ website থেকে redirect করে দিবে। কিন্তু user যাবে website এ।

• use PUSH Service. যখন মানুষ Netflix কম use করে তখন CDN গুলাই নিজে upload করে দেয়।



• MADMEN এর বিভিন্ন version বিভিন্ন CDN এ store করে রাখে।

Scanned with CamScanner

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

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

CDN:

OTT → Over the Top

~~এ internet~~

Stream এর জন্য internet এর top level ব্যবহার করে।

CDN content access: a closer look:

From slide + BOOK

- name to name convention হবে DNS এ। netcinema.com এ

▶ বর্তমানে netflix private CDN use করে তাই Redirect এর প্রয়োজন হয় না।

➡ Normally Third party CDN use করলে প্রয়োজন হয়।

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

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

Case Study: Netflix :

Slide থেকে পড়বে।

→ private CDN use হয়। তাই Redirect এর প্রয়োজন হয় না netflix এ। কারণ সে জানে তার authoritative DNS কোথায়

- Youtube Dash ব্যবহার করে না।
  - ↳ Youtube এ download করার সময় quality নিজেরাই select করতে হয়।
  - ↳ Netflix এ automatic হয়।

⇒ Youtube pull cash in ব্যবহার করে। — private CDN আছে।

⇒ Netflix " " " না।

▼▼▼ ——— ←

→ Netcinema website movie এর reference থাকে

→ movie থাকে CDN এ

Bob প্রথমে Netcinema website visit করবে। ~~কারণ~~ Local DNS এ যাবে। Local DNS server hand over করবে netcinemar authoritative DNS server এ. কারণ ওর কাছে (local DNS) এ info নাই। Net cinemar authoritative IP দিবে না কারণ netcinemar কাছে video Store নাই। name to name conversion হবে। সে King CDN এর এক host name local server এ পাঠাবে। তারপর King CDN এর Authoritative এর কাছে request দিবে the KING CDN এর IP Bob কে পাঠাবে। সে IP অনুযায়ী visit করে video দেখতে পারে।

Scanned with CamScanner

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

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

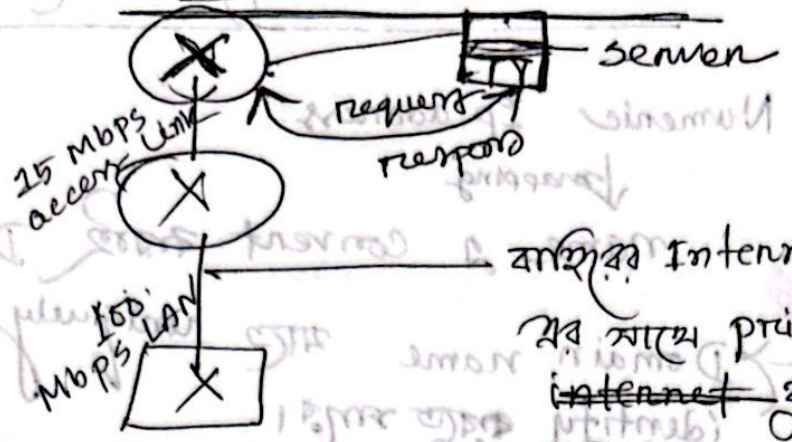
Domain Name System (DNS)

Numeric Ip address
↓ mapping
name এ convert করার DNS

→ Domain name যাতে uniquely identify করতে পারে।

Microprocessor
Artificial Intelligence
COA
Numerical
Networking

Install Web Cache



বাহিরের Internet এর সাথে private network ~~internet~~ যুক্ত হয়।
এটাকে বলে access link

ধরি,

* per second এ নির্দিষ্ট document এর জন্য 15 টা request যায়।
size 1 Mb

∴ per second এ 15 Mb এর request যায়, Institution থেকে।

* public Internet এর Router থেকে ~~Ser এ রে~~ Origin Server এ request গিয়ে response আসতে প্রয়োজনীয় সময় 2 second ধরে নিব (RTT) → Internet delay

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

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

1.5×10^{-3}

- তিনটা delay consider করতে হবে।

→ Internet delay → 2 sec
→ access delay
→ LAN delay

B.book

traffic intensity of LAN

$$\frac{(15 \text{ request/sec}) \cdot (1 \text{ Mbits/request})}{100 \text{ Mbps}}$$

$= 0.15$

intensity on the access link

$$\frac{(15 \text{ request/sec}) \cdot (1 \text{ Mbits/request})}{15 \text{ Mbps}}$$

$= 1$

1MB = 1000 KB
1GB = 1000MB

1 msec = 0.001 sec

Slide

Access link

$$\frac{(15 \text{ req/sec}) \cdot (100/1000 \text{ Mbits/req})}{1.54 \text{ Mbps}}$$

$= 0.974$
↳ minutes

LAN

$$\frac{(15) \cdot (100/1000)}{1000 \text{ Mbps}}$$

$= 0.0015$
↳ μsecs

Distance Vector Algorithm

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

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

→ 7 layer with protocol

Chap-1 (Exercise

Sobuz sir

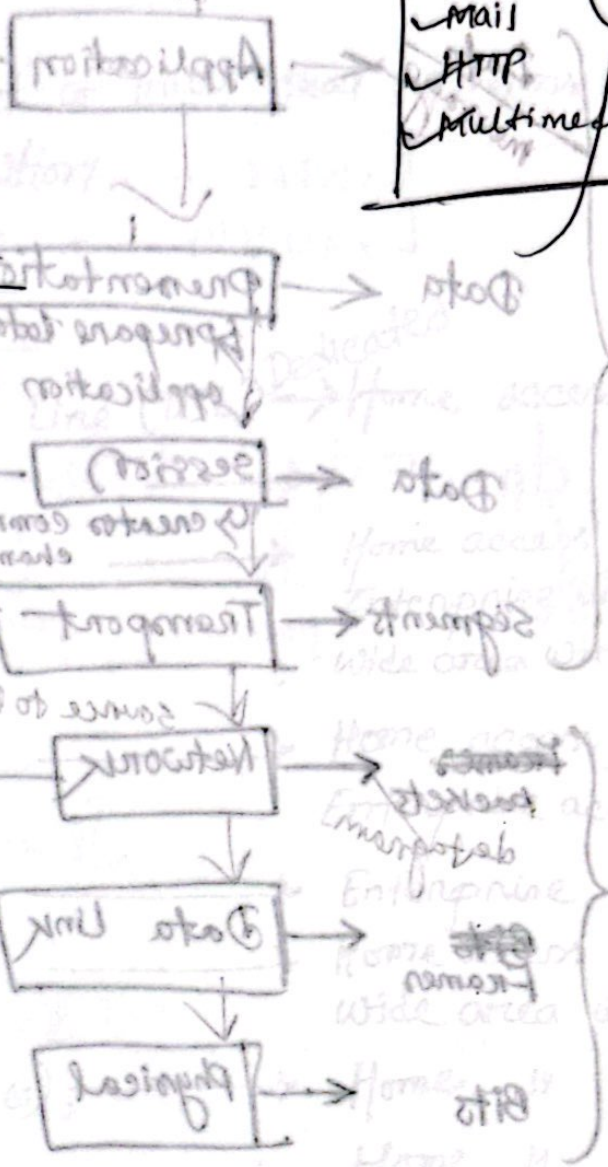
Chapter-01

Saroar maam

2-set

DNS
Mail
HTTP
Multimedia

Revision রাত ৮টা

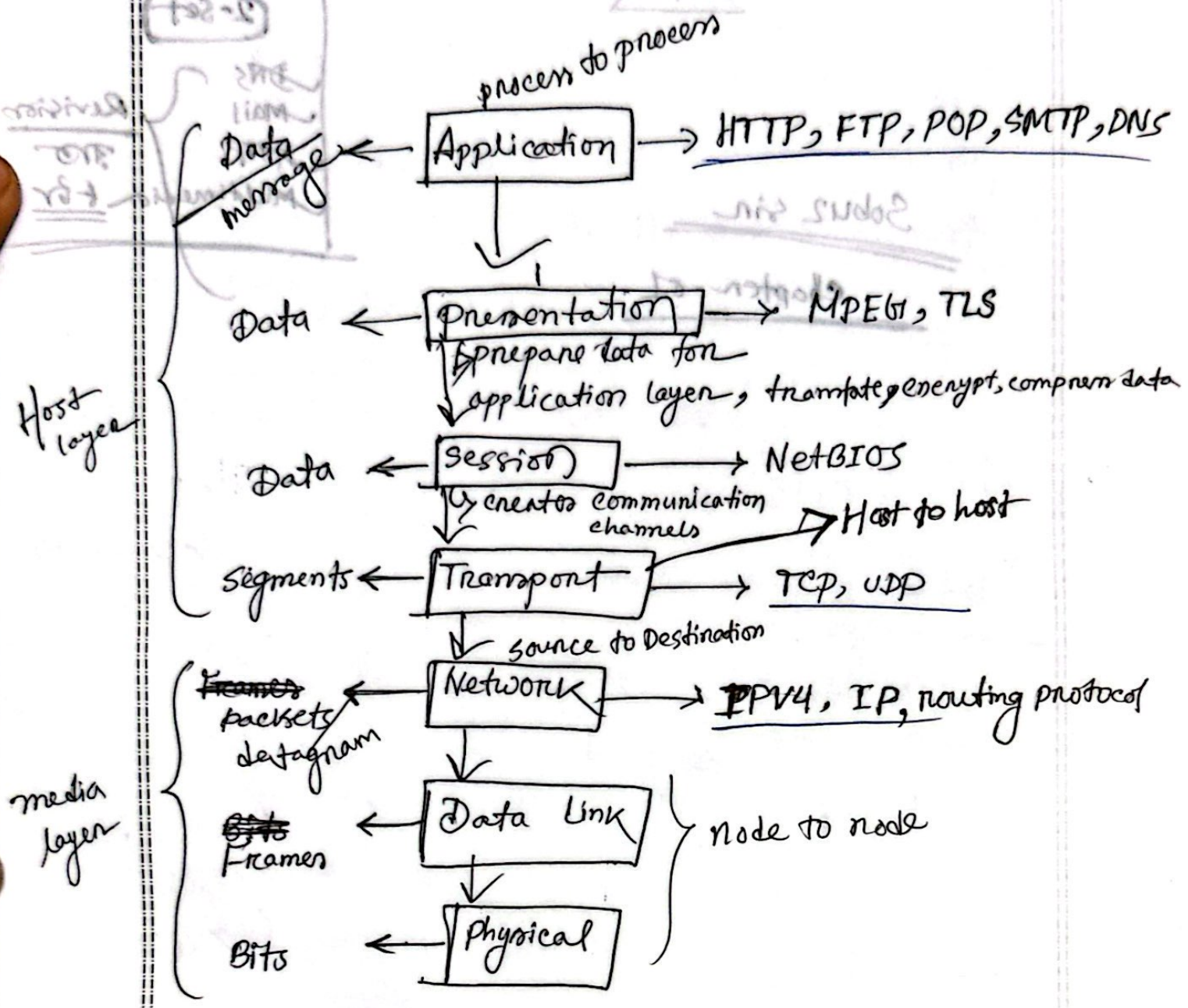


Chap-01 (1-set)

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

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

Write the duties/services and protocols of each specific OSI Layer.



Dial-up-modem
(Dedicated)

Sat / Sun / Mon / Tue / Wed / Thu / Fri
Date :...../...../.....

host = client and servers
↓
often in data centre

Downstream speed
upstream speed

protocol:

protocol is a set of rules that governs the data communication.

Access technologies:

DS → (24-52) Mbps
US → (3.5-16) Mbps
(8K-2.5K)

Digital Subscriber Line (DSL) —Dedicated→ Home access

Cable Internet → " "

DS → 100 Mbps
US → 10 Mbps
(1K-4K)

Fiber optic → Home access.
Enterprise access
Wide area wireless acce

WiFi (802.11) → Home access, Wireless access
Enterprise access

10 Mbps, 100 Mbps, 1 Gbps, 10 Gbps
• Run over twisted pair → Fiber optic

Ethernet → Enterprise access

Satellite Internet → Home access
Wide area wireless a

Cellular (3G, 4G, 5G) → Home, "

MiMAX → Home, "

HFC (Hybrid Fiber Coax) —Shared→ cable based access

Coaxial cable Internet → "

upload (2-10) Mbps
download (10-20) Mbps

Fiber to the Home (FTTH) —dedicated→ Home access

Scanned with CamScanner

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

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

HFC (Hybrid Fiber Coaxial) → bandwidth is Shared among users.

- (40 Mbps — 1.2 Gbps) downstream transmission rate. → no collision
- (30-100) Mbps upstream transmission rate

popular wireless internet access technology

- → WIFI → 10 meter
- → 3G, 4G → 10 Km

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

- d_{nodal} → total delay
- d_{proc} → processing <ms sec
- d_{queue} → queuing
- d_{trans} → transmission $= L/R$
- d_{prop} → Propagation $= d/s$

L = packet length
R = transmission rate

d = length of physical link
s = propagation speed

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

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

Packet switching:

Transfer of small piece of data across various networks.

Circuit switching

A dedicated channel or circuit is set up for a single connection between the sender and receiver during the communication session.

Packet switching	Circuit switching
(i) Shared resource dynamically among multiple user and session.	(i) Dedicated connection for entire session.
(ii) Resource Reservation.	(ii) no resource reservation
(iii) data into packets which are individually routed and sent	(iii) established a dedicated physical path for the entire session.
(iv) Variable delay due to packet processing and routing	(iv) Lower delay. once connection is establishment
(v) Ex: IP network	(v) Ex: public telephone network.
	* Can gurantee certain amount of end to end

Signal into frequency

requires analog hardware

FDM

TDM (rate conv)

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

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

IXP → Internet Exchange point

↳ এটি দ্বারা ISP গুলো connected থাকে।

(meeting point where multiple ISPs can connect and peer together).

(19) # File size = 4 million bytes = 4×8 million bits = 32 million bits

Throughput = 500 kbps

= 500000 bits per second

Time to transfer = $\frac{32 \text{ million}}{500000}$

= 64 sec

$\boxed{1 \text{ sec} = 1000 \text{ msec}}$

(18) # Distance = 2500,000 meter

Speed = 2.5×10^8 m/s

$T_p = \frac{d}{s} = \frac{2500,000}{2.5 \times 10^8}$

= 0.01 sec

= 10 msec

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

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

Layer can perform

- Error control → provided more than one layers.
- Flow control
- Segmentation
- multiplexing
- connection setup

(27)

Five layer

- Application
- Transport
- network
- Data link
- physical

Layer → কী থাকে → protocol → duties → principle Responsibilities every layer এর

* কিছু problem
* Layer details

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

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

Physical layer :

* physical connection between the devices.
* contains info in the form of bits.
* transmitting individual bits from one node to the next.
* get signal & convert 0 & 1 and sent to Data link layer.

- Bit synchronization
- Bit rate control
- physical topologies
- Transmission mode — Simplex, half duplex, Full

device: Hub, Repeater, Modem and cable

Data link layer: (node to node)

* make sure data transfer is error free from one node to another

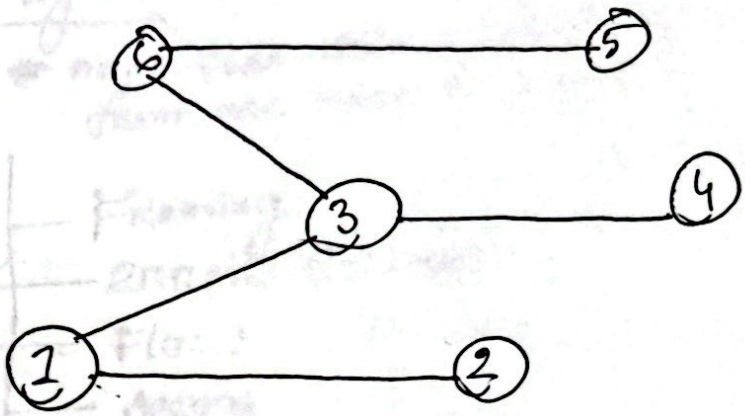
- Framing
- error control
- Flow "
- Access "
- physical addressing

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

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

$1 \rightarrow (1\text{-}4) \rightarrow 5 \rightarrow$ IPV4

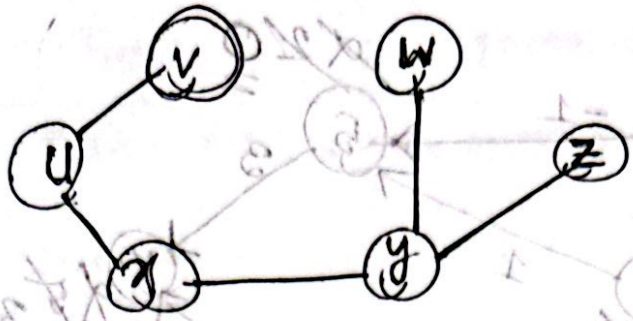
1	2	3	4	5	6
1	7,1	9,1	∞	∞	14,1
1,2		9,1	22,2	∞	14,1
1,2,3			20,3	∞	11,3
1,2,3,6			20,3	23,6	
1,2,3,6,4				23,6	
1,2,3,6,4,5					



4

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

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



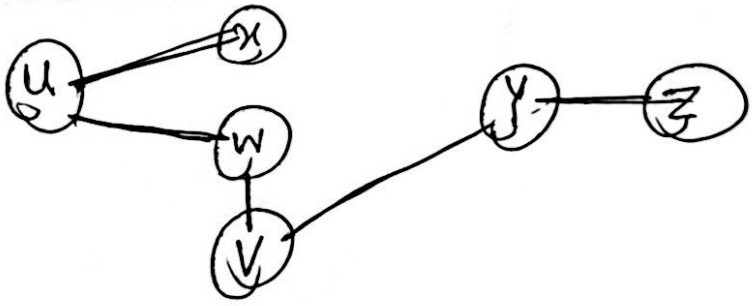
Dijkstra's

↓

Link state routing algorithm

	u	v	w	x	y	z
u		7, u	3, u	5, u	∞	∞
uw		6, w		5, u	11, w	∞
uwx		6, w			10, v	14, x
uwxv					10, v	14, x
uwxvy						12, y
uwxvyz						

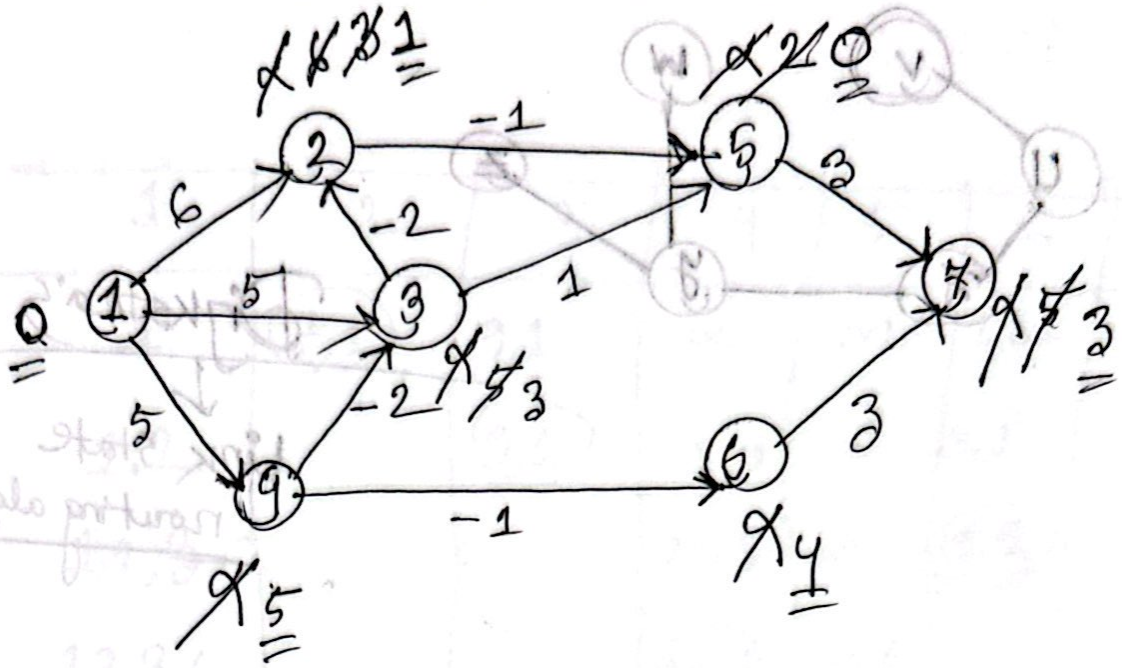
D(v) P(v)



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

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

Relaxation = (7-1) = 6



if $d[u]+c(u,v) < d[v]$

$d[v] = d[u] + c(u,v)$

(1,2) (1,3) (1,4) (4,3) (3,2) (2,5) (3,5) (4,6) (5,7) (6,7)

Step-1 : complete

Step-2 : compute

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

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

	Layer	protocols	
Message	Application →	HTTP, FTP, Telnet, SMTP →	Application specific
Segment	Transport →	TCP, UDP →	Port
Packet	Network →	IP (IPV4, IPV6), ICMP →	Logical
Frame	Data link →	Ethernet, WiFi →	Physical address (MAC)
Bit	Physical →	DSL, 10 Base-T	

OSI - open system Interconnections

- Addressing
  - physical → manufacture company এর। → Change hop to hope
  - Logical → IP
  - port
  - Application-specific

  IP, port → always same

FTP → 21
HTTP → 80
HTTPS → 443
TELNET → 23
SSH → 22 ← Secure shell

port number

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

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

Data link layer
- LLC – Logical Link control
- MAC → Media access control

- All bits 1 → ~~multicast~~ broadcast
- LSB " → 0 → Unicast
  → 1 → multicast

- 56K Modems → Telephone এর মাধ্যমে
- DSL Technology
- Cable modem
- T Lines
- SONET
- PPP
- PPPOE

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

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

Circuit switching

→ Circuit Change করতে হয় —— পুরাতন (early telephone system)

Packset switching — আধুনিক

→ connectionless

→ connection-oriented service

beginning needs to be divisible by the number of adder