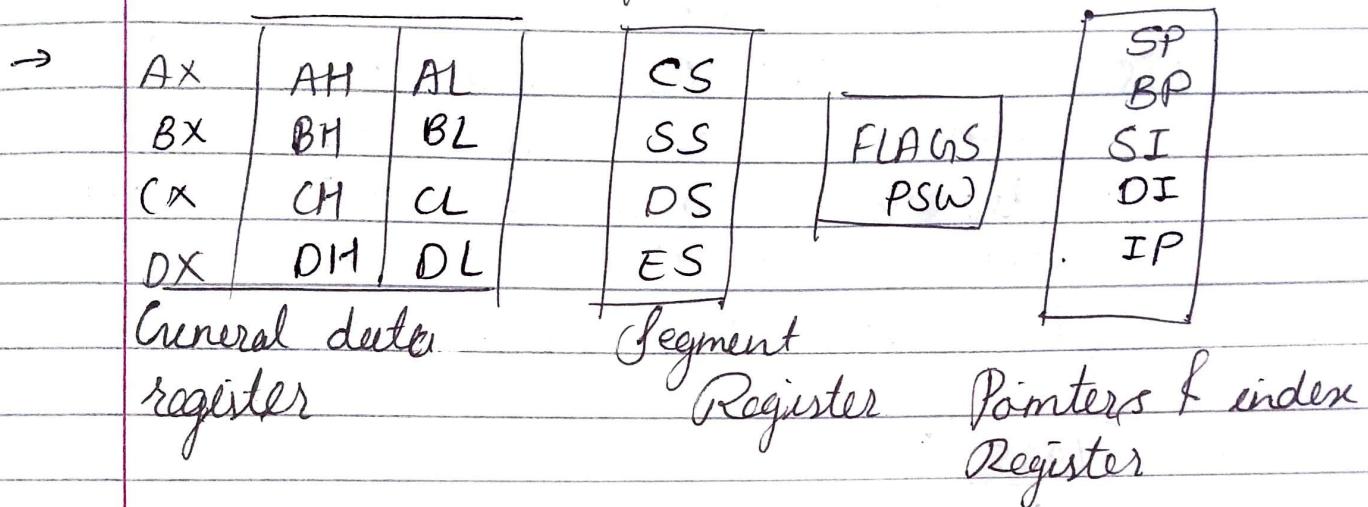


18DCE115

1. Explain register organization of 8086 with the help of Block diagram.



→ 8086 has a powerful set of registers. All of them are 16-bit registers.

→ They are classified into: 1) General Data
2) Segment Reg. 3) Pointers & index 4) Flag.

→ 1) General Data Register:

→ AX, BX, CX, DX are general ~~data~~^{16-bit} registers.

→ All data reg. can be used as 16 / 8 bit

→ AX is used as 8 bit accumulator. AL is designed 8-bit while AH is higher 8 bit + used for 8-bit opⁿ. BX is used for offset storage for forming physical add. in case of certain addressing mode. CX is used default counter in case of loop and string instruction.

2) Segment Register: 18dec115

- A) CS Register
- B) DS Register
- C) ES Register
- D) SS Register

→ CS: Used for addressing memory location; the code segment of the memory.

DS: Points to the data segment of the memory where the data is stored.

ES: Also refers to a segment in the memory which is another data segment in memory.

SS: Used for addressing stack segment.

3) Flags / PSW:

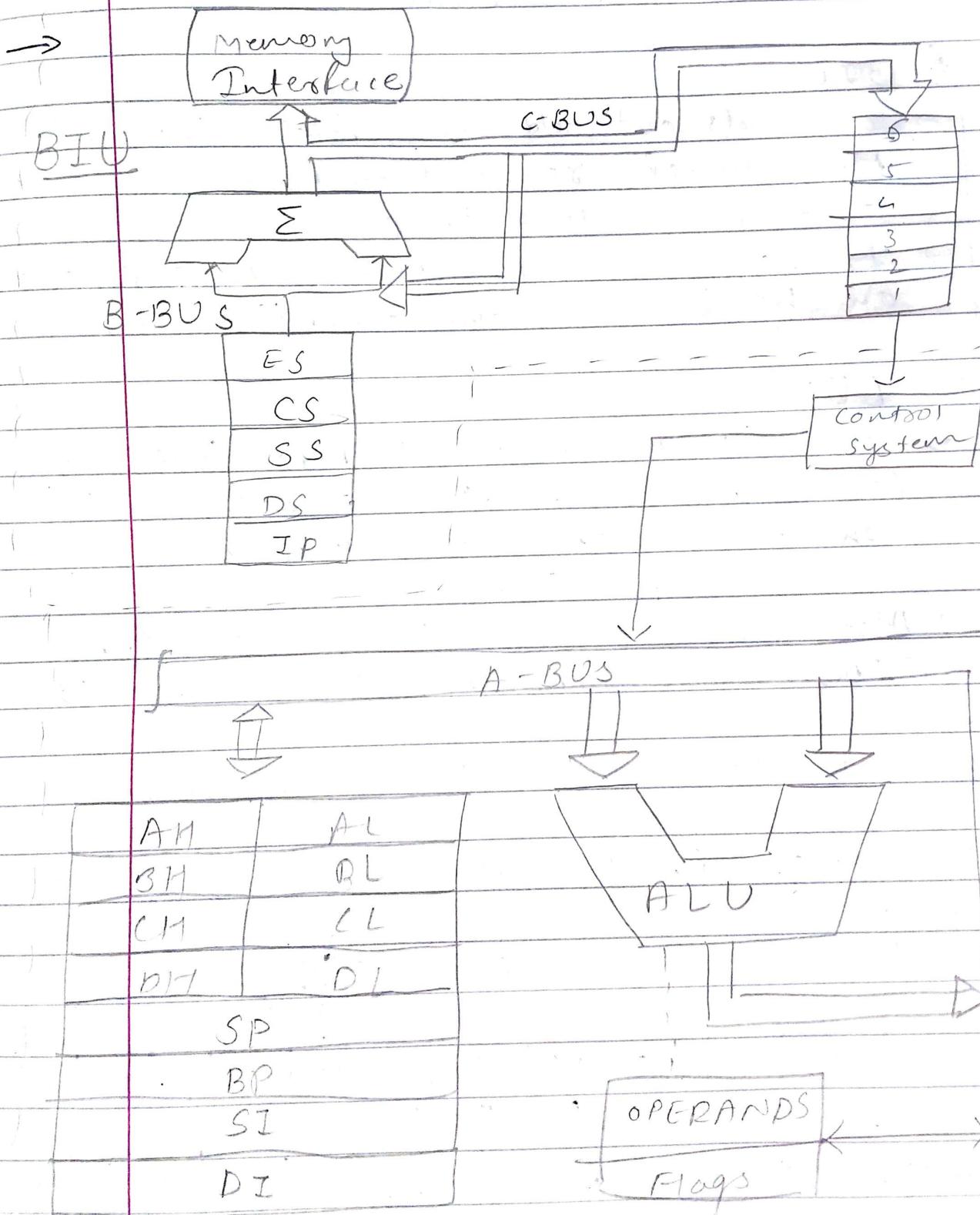
→ Contents indicate the results of computation in the ALU. It also contains some flag bits to control CPU opn.

4) Pointer & Index:

- IP: Store memory loc. of next instruction
- BP: Contains offset within data segment
- SP: Contains offset within stack segment
- SI: To store the offset of source data in data segment.
- DI: Used to store offset of dest in data of extra segment.

18dec115

Q. Draw & Explain Architecture of 8086 Microprocessor.



18dce115

- Divided into two independent functional parts: BVS Interface Unit Execution Unit

(A) BIU:

- It sends out addresses, fetches instructions from memory, reads data from ports & memory & conveys data to ports & memory.
- It handles all transfers of data & address on the buses for the execution unit.

(B) EU:

- Tells the BIU where to fetch instⁿ or from decodes instⁿ & executes instⁿ.
- Contains control circuitry, which directs internal operations.
- Also has a 16-bit ALU which can ADD, SUBSTRACT, AND, OR, XOR, INCREMENT

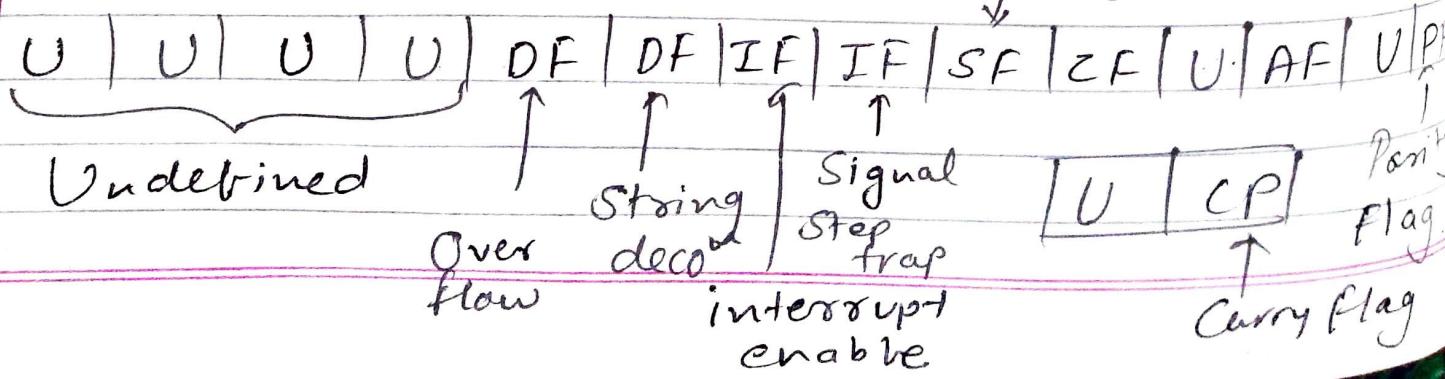
[3]

Explain flags of 8086 microprocessor.

- 16-Bit flag divided into two parts:

- Condition Code

- Machine control flag. Sign Flag



18dcells

SF: It is set when the result of any comput" is negative.

ZF: It is set, if the result of the comput" or comparison performed by the previous instruction is zero.

PF: It is set to 1, if the lower byte of the result contains even number of 1's.

CF: It is set, when there is a carry out of in case of cond" or borrow of subtraction.

AF: Auxiliary carry is set if there is a carry from the lowest nibble.

OF: It is set when overflow occurs.

TF: It is set, the processor enters the single step execution mode.

JF: It is set, the maskable interrupts are recognised by the CPU.

DF: Used by string manipulation instr'.

(4)

Explain all the signals of 8086 Micro-processor.



- These are three signals:
 - 1) Control Signal
 - Interrupt Signal
 - 3) DMA Interface Sig

18dce115

1) Control Signal:

→ They control functions such as when bus is to carry a valid address in direction data are to be transferred on the bus, when valid write data are on bus & when to put read data on the system bus.

2) Ready Signals:

→ It is used to insert wait states into bus cycle such that it is extended by a of clock periods.

3) Interrupt Signal:

→ The key interrupt signals are interrupt request (INTR) & INTA. INTR represents an active interrupt request. When interrupt request has been recognized by 8086, it indicates this fact to external circuit with logic 0 on the INTA output.

3) DMA Interface Signal:

→ The DMA interface of the 8086 minimum mode consist of the HOLD and HLDIA signals.

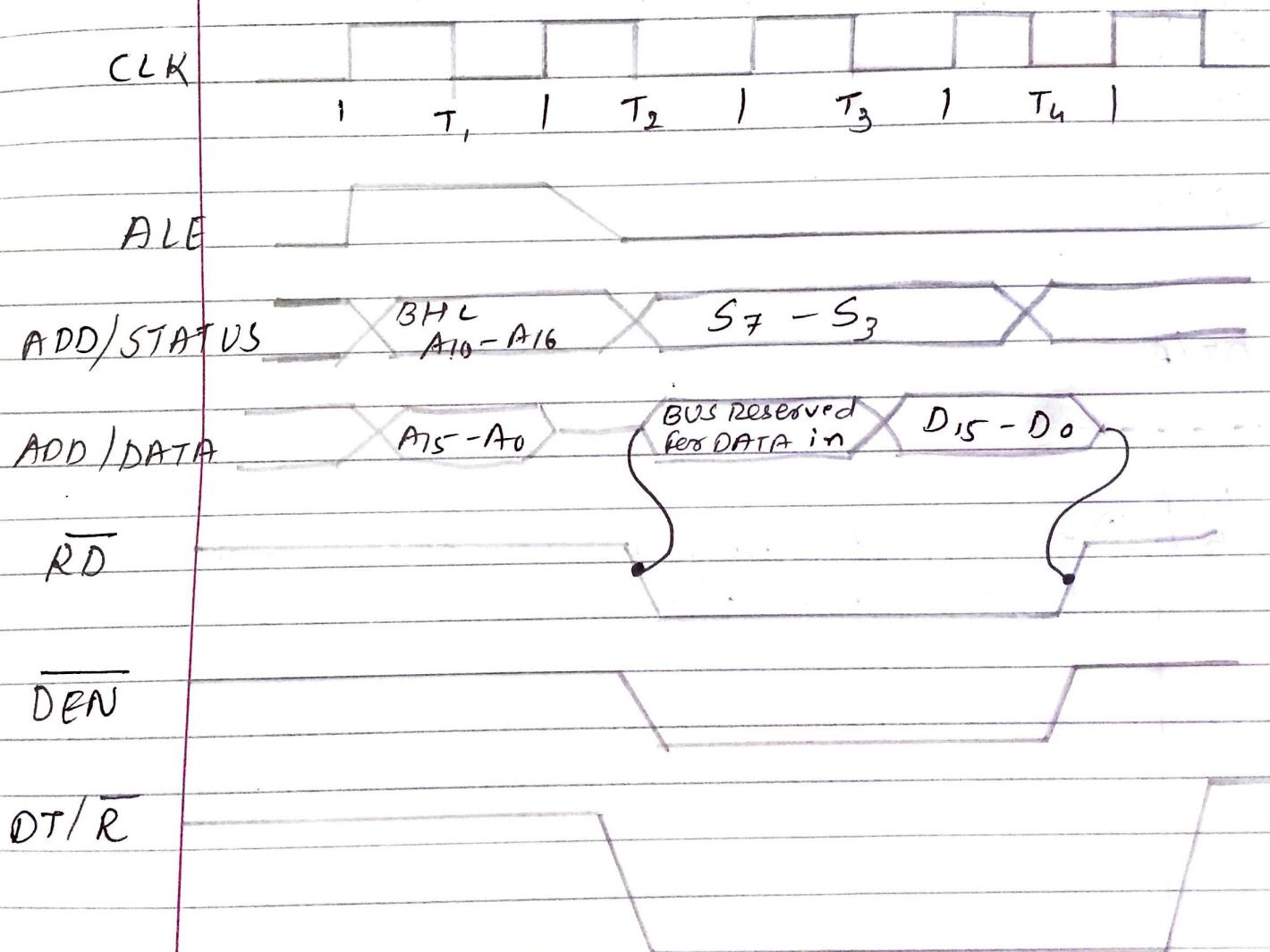
→ When an external device wants to take control of the syst. bus it signals to the 8086 by switching HOLD to logic 1.

The 8086 signals external device that it is in this state by switching its HLDA output to logic 1 level.

[5]

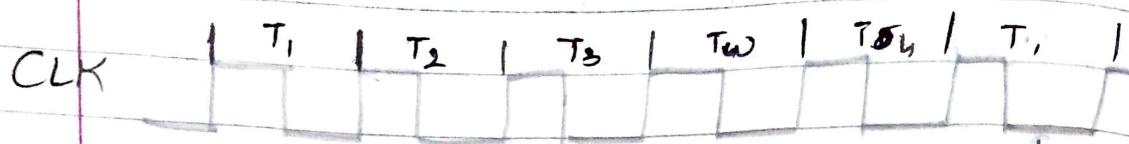
Draw Timing Diagram of memory read & memory write operation.

① READ:

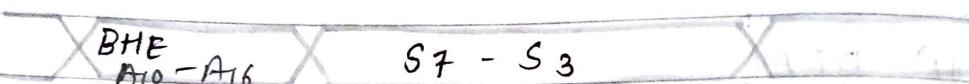
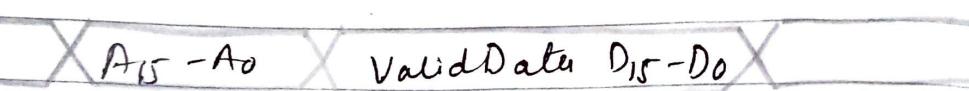


2) WRITE:

18dce115



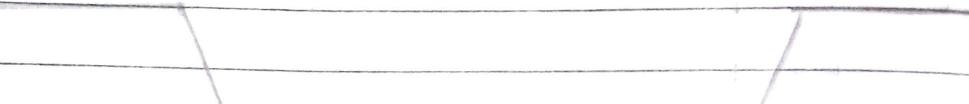
ALE

ADD/
STAGESADD/
DATA
⑥

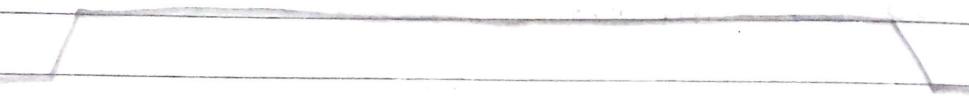
WR



DEN



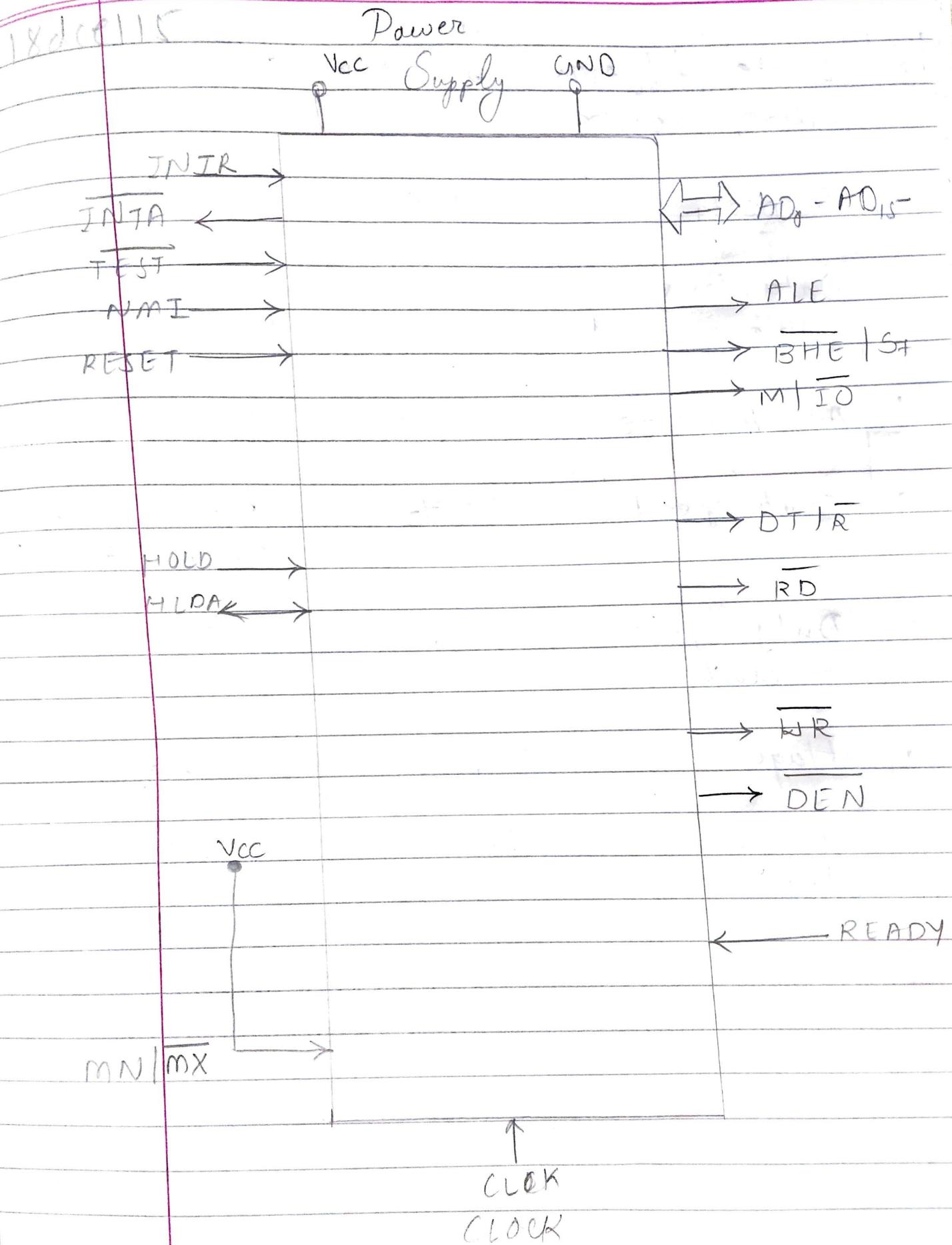
DT/R



[6]

Draw block diagram of 8086 microprocessor of minimum mode.

18dC9115



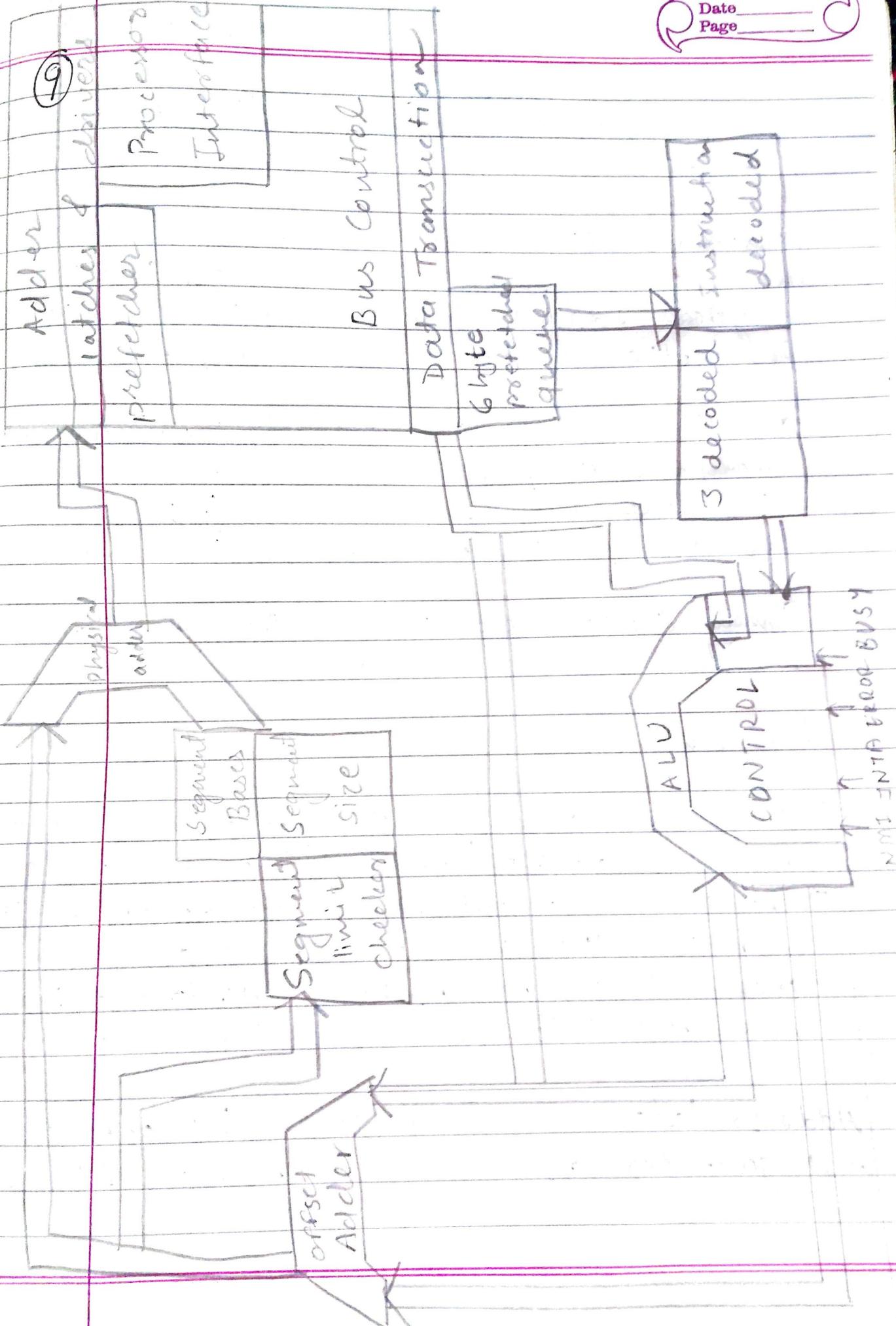
8

18dce115
Write difference between 8085 & 8086 microprocessors.

	8085	8086
→ Property		
→ Data, Bus size	8-bit	16 bit
→ Add. Bus size	16-bit	20-bit
→ Clock speed	3 MHz	Varies in range 5.8 - 10 MHz
→ Duty cycle for clock	50%	33%
→ Flags	5 flags	9 flags
→ Memory Size	64 KB	1 MB.

(9)

change



1. Address Unit

- 80286 Internal Block diagram [4 units Address, Instruction, Execution].

Address Unit: for calculating physical address we need segment address & offset address. So, AU calculates physical address of the instruction to be executed. Instruction & data is present at phy. add.

- CPU fetches, decodes & executes the instruction. AU computes the physical add. of instⁿ & data. CPU wants access. In real mode, AU computes using segment base & offset (20 bit add.).

Segment Checker: checks whether the segment exceeds limit of the add. or not.

Bus Unit:

- It performs same operations as BIU of 8086. It performs all memory & I/O read & write operations.
- It acts as an interface to outside world. It prefetches the instruction bytes, stores them in queue. It controls transfer of data to and from processor extension devices like 80287 math coprocessor.

Address Latches- Contains 20/24 bit address.

- It generates control signals such as BHE Bus.

180CE115

Instruction Unit:

It decodes 3 prefetched instructions & stores it in a queue which are given to execution unit.

Execution Unit:

It is same as that of 8086 microprocessor.
It executes instructions coming from IU.
In Real mode, it has same set of instructions as 8086.

(10)

→ Read Address Mode:

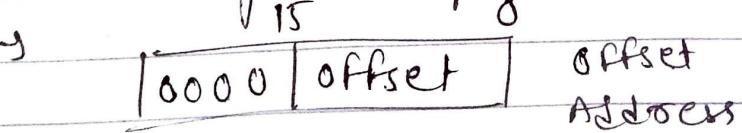
→ 80286 supports read add. mode. 80286 works similar to 8086 in this mode.

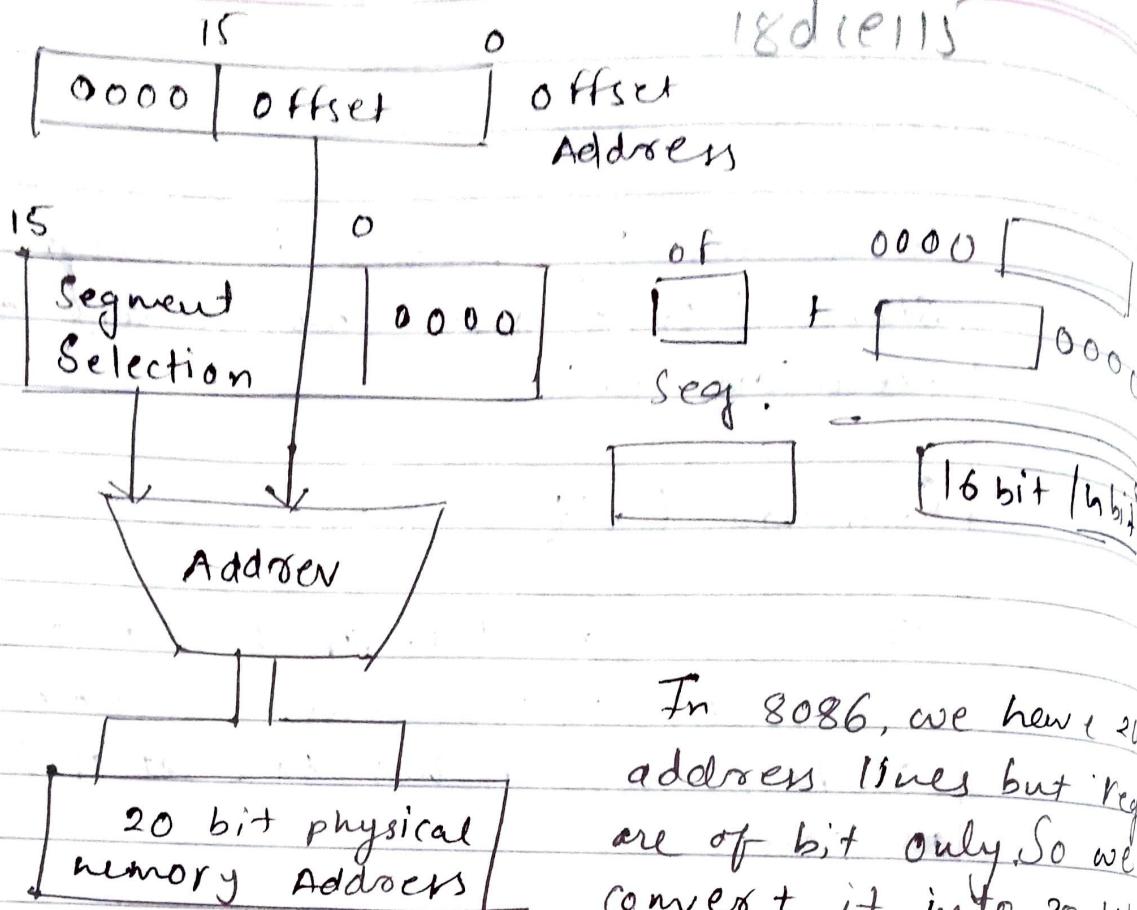
→ As in 8086, physical memory is organized in terms of segments of 64 KB.

→ In real mode, first 1 KB of memory starting from 00000H to 003FFH is reserved for interrupt Vector tables.

→ Address from FFFF0H to FFFFFH are reserved for system initialization.

→ In real mode, it initialize the IP and other registers of 80286.





In 8086, we have 20 address lines but they are of 16 bit only. So we convert it into 20 bit by appending zeros.

* Protected Virtual Address Mode:

→ 80286 is the first processor to support concept of virtual memory & memory management.

→ It is able to address 1 GB of virtual memory.

→ Terminology:

1) Selector

2) Descriptor

3) Descriptor Table

4) Global & Local descriptor table

18dce115

Q1] What are the Salient features of 80286?

- It is an advanced version of the 8086 microprocessor.
- 80286 is designed for multiple & multitasking system.
- It is high performance 16 bit microprocessor, six times faster than 8086.
- It provides H/W & S/W logic to implement multitasking OS like Windows.
- It is upward compatible with 8086, 8088 & 80186/80188 instr set.
- It was used in IBM PC-A/T. It provides 24-bit address bus, hence it can access 16 MB physical or semiconductor memory.
- The clock frequencies are 4 MHz, 6 MHz, 8 MHz, 10 & 12.5 MHz.
- It is housed in 68 pin flat package. It is a stage pipeline microprocessor.
- It performs in 4 MIPS.
- The MMU also provides protection mechanism.

Q2] Signal Description of 80286.

- 80286 is available in 68 pin PLCC.

CLK - This is the system clock input pin. The clock frequency applied at this pin is divided by two internally and is used for deriving fundamental timings for basic operations of circuit.

18 d(c) 115

$A_0 - A_{23}$ (24 bit Add. lines)

→ Unidirectional so we can carry address from microprocessor to memory to peripheral device.

$$2^{24} = 2^4 \times 2^{20} = 16 \text{ MB}$$

$D_0 - D_{15}$ (16 bit data lines)

They are used to carry data.

BHE: $(D_0 - D_7) \& (D_8 - D_{15})$ We can transfer data on higher orders $(D_8 - D_{15})$

COD/INTA, TO, SISO: works together for bus cycle transfer of I/O or interrupts.

Ready: If peripheral device is ready to transfer data it informs through ready signal.

Lock: Peripheral device raises lock signal so that no one can raise exception when peripheral device is working.

HOLD & HLD ACK: It comes from peripheral device that it wants to hold data transfer & there's hold acknowledgement.

PREQ: It requests 80286 to perform a data operation transfer for processor extension.

PEACK: It indicates the processor extension that data transfer request or the requested operation is being performed.

INT! When peripheral devices wants to connects
intoupts is.

RESET! It resets the microprocessor.

* Do as directed.

① org 100h

MOV AX, 2000h

MOV DS, AX

MOV SI, 400h

MOV CX, 0fh

MOV AX, 000h

MOV AL, [SI]

begin : inc SI

CMP AL, [SI]

inc abc

MOV AL, [SI]

abc : dec CL

inc begin

ret.

② org 100h

MOV AX, 3000

MOV DS, AX

MOV DL, 1uh

MOV SI, 100h

MOV DI, 500h

MOV AX, 0000h

l2: MOV AL, [SI]

MOV CL, DL

l1: inc SI

MOV BL, [SI]

Mov CMP AL, BL

l2 next

ic next

xchg AL, BL

Mov [SI], BL

l3: loop l

Mov BH, AL

Mov [DI], BH

inc DI

18dcells

MOV SI, DI

dec DL

inz 12

ret.

③ org 100h

MOV AX, 3000h

MOV DS, AX

MOV SI, 200h

MOV CX, 0Ah

MOV DH, 00h

MOV DL, 00h

a: MOV BH, [SI]

SHR BH, 1

jc odd.

inc even

add: inc DH

inc SI

dec CX

inz a

hit

Even: inc DL

inc SI

dec CX

inz a

④ org 100h

jmp start

m1:

S DB 'abcba'

S size = \$ - m1

db 0Dh, 0Ab, '\$'

start:

max ab, 9

mov dx, offset s

int 21h

lea di, S

mov si, di

add si, S size

dec si

mov CX, S size

cmp CX, 1

je is-palindrome

next char:

mov al, [di]

mov bl, [si]

cmp al, bl

jnp not-palindrome

inc di

dec si

⑤ Count = delay * f
n

Let n = 18

$$\text{Count} = \frac{200 \times 10^{-3} \times 5 \times 10^6}{18}$$

$$= 55555.55 \approx (0903)$$

→ MOV CX, 0D903H

LABEL:

DEC CX

JNZ LABEL

⑥ org 100h

MOV AX, 2000h

MOV DS, AX

MOV AX, 0001H

L1: CALL RED

CALL YELLOW

CALL GREEN

CALL YELLOW

JMP L1

H2T

YELLOW: OUT 02, AX

MOV CX, 094CH

L3: DEC CX

NOP

JNZ L3

RET

GREEN: OUT 03, AX

MOV CX, 94CFH

L4: DEC CX

NOP

JNZ L4

RET

RED: OUT 01, AX

MOV CX, 2533H

L2: DEC CX

NOP

JNZ L2

RET

RET

(7)

org 100h 180115
MOV AX, 0207H
PUSH AX
POPF

(8)

MOV AX, 5000H
MOV DS, AX
MOV AL, [0700H]
MOV CX, [0700H]
SUB CX, 1

LABEL1 :

MUL CX
LOOP LABEL1

RET