UNIT - 1

→ Introduction :- Microprocessor based system, Origin of μprocessors, Harvard and Von Neumann architectures with examples, Microprocessors Unit versus μcontroller Unit. 8086 architecture :- Internal architecture of 8086 μprocessor, register organization, physical memory organization, general bus operation.

Microprocessors :- It is a programmable circuit that supports the execution of a set of instructions called an instruction set.

- Each instruction in the instruction set is represented by a predefined unique sequence of 1's & 0's called "OPCODE" (operation code)
  - μprocessor can execute one instruction at a time.
  - Sequence of instructions is called a "program"
  - These programs expressed in terms of hexadecimal bytes are called 'machine language programs'

→ Thus, the type of circuits, systems and machines which can remember the sequence of instructions, operands related to each instruction, can execute them and store the result are called "programmable circuits or machines".

Origin of microprocessors :-

Intel corporation is a pioneer in processor industry right from infant stage till date.

(1) The first 4-bit processor 4004 from Intel corporation in 1971. It has only 1000 transistors with huge component density.

(2) Immediatly in 1972, Intel introduced 8 bit microprocessor 8008 which was not successful because of many limitations.

(3) In 1974, Intel released the first general-purpose 8-bit microprocessor 8080.

(4) Finally, in 1977, first 8-bit fonctionally complete CPU 8085 was introduced.

* The most popular among 8-bit processor has on chip clock generator, optimum utilization of registers, & powerful ALU.

* The major limitations are limited memory addressing capacity, slow execution, limited general purpose registers, complex instructions & addreseingmodes not available.

It does not support pipelining or parallelism for execution of instruction

(5) The first 16 bit CPU - 8086 was introduced in the year 1978 by Intel.

* 8086 has 16-bit general purpose registers which supports 16 bit ALU, rich instruction set, segmented memory addressing scheme.

* The major limitation was it doesnot have memory management and protection capabilities

* 8086 is the first member of Intel's advanced processors family.

(6) IBM PC started in July 1980, which was equivalent to 8086 but has 8-bit external data bus.

* It has one or two floppy disk drives, keyboard, monitor and early version of MS-DOS as operating system from microsoft.

(7) In March 1983 IBM-PC launched new version called PC-XT with 10 megabyte hard disk, floppy disk, keyboard, monitor and an asynchronous communication adapter.

(8) The Intel's 8086 limitations were overcome by 80286. But it has a major limitation that once it is switched into protection mode, it was difficult to get it back to real mode, the only process is to reset the system.

(9) Due to the demand of faster CPU's in mid-80's 80386 was introduced with 32 bit CPU from Intel.

* It supports huge virtual memory, paging and four

levels of protection. It has math coprocessor, high speed graphical applications.

(10) Later 80486 CPU was introduced by Intel due to the OS windows 95 graphical interface.

In addition to all the features, it has 8KB four way set associative code and data cache & five stage instruction pipelining.

(11) In the course of evolution to support more & more complex instructions at assembly language level the designers introduced 64-bit processor: Pentium I.

Thus a family of operations like single instruction Multiple data (SIMD) & Multiple instruction Multiple data (MIMD) emerged for multimedia applications wii which resulted in launching of pentium variants - PI, P-II, P-III, P-IV, pentium PRO, & pentium MMX etc.,

Later the technology leads to single processor core to multi-core processor architecture with massive parallelism and processing capabilities

Generation of microprocessors:-

(i) **I generation (1971 - 1973):-**

a) These I generation processors works serially- they fetched the instructions, decoded it and executed it

b) Intel's 4004 ,- 4 bit processor in 1971
8008 - 8 bit processor in 1972
8080 - 8 bit " comes in this generation 1974.

(ii) **II generation (1974 - 1978):-**

a) Very large scale integrated (VLSI) circuits led to chips that speed up to hundreds of millions of switching / second.

b) It marks the beginning of 8- bit processor like Motorola's 6800 & 6809, Intel's 8085 & Zilog's Z80.

(iii) **III generation (1978 - 1980):-**

a) A 16- bit processor was in existence.

b) Intel's 8086 in 1978 & Zilog's Z8000 was introduced.

c) This generation has IC transistor counts of about 2,50,000.

(iv) **IV generation (1981 - 1995):-**

a) It marks the beginning of 32- bit processor with millions of transistors on single package.

b) Intel's 80386 and motorola 68020/68030 introduced in this generation.

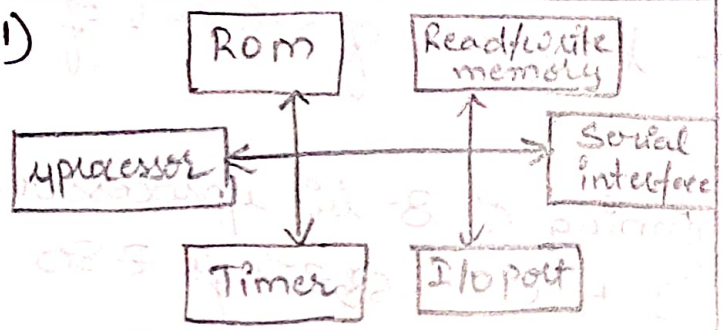
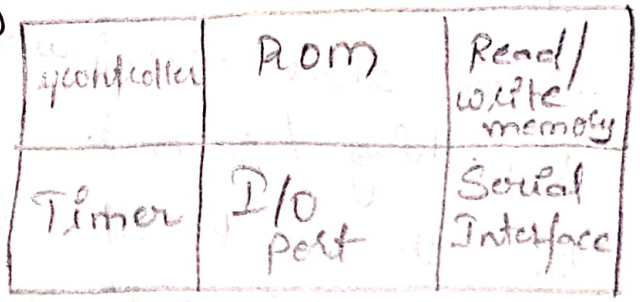
(v) Fifth generation (1995 - till date)

a) It has super scalar processing and designed with more than 10 million transistors.

b) It introduced 64-bit μprocessors.

c) This generation has Intel's Pentium, Celeron and recently dual and quad core processors with upto 3.5 GHz speed.

⟹ Microprocessor VS Microcontrollers :-

Microprocessors	Microcontrollers
1) 	1) 
2) Microprocessor is a heart of computer system	2) Micro controller is a heart of embedded system.
3) It is just a processor, memory and I/O components have to be connected externally	3) It has external processor along with internal memory and I/O components.
4) Due to I/O and memory devices connected externally circuits becomes large.	4) Memory and I/O present internally, the circuit is small.
5) Cost of entire system increases.	5) Cost of entire system is low.
6) External components leads to high power	6) Since external components are low, power consumption's

consumption. They can't run on batteries.	is low. They can be run on batteries.
7) Since memory & I/O components are external, they need external operation hence it is slower.	7) Since components are internal, most of the instructions are internal, hence the speed is fast.
8) It has less no of registers hence more operations are memory based.	8) It has more no of registers hence programs are easier to write.
9) Microprocessors are based on Von-Neumann architecture where program and data are stored in same memory module.	9) Micro controllers are based on Harvard architecture where program memory & data memory are separate.
10) Mainly used in personal computers.	10) Used mainly in washing machines, MP3 player etc;

⇒ VON NEUMANN ARCHITECTURE :-

* Von Neumann architecture is based on the concept of stored program computers where program data and instruction data are stored in the same memory.

It was designed by famous mathematician and physicist John Von Neumann in 1945.

Advantages :-

① Simplicity :- Since all data and instructions are

stored in same memory there is no need to create complicated system routing that may coincide.

2) Cost-Effective :- Smaller components are needed compared to other architectural designs hence economical.

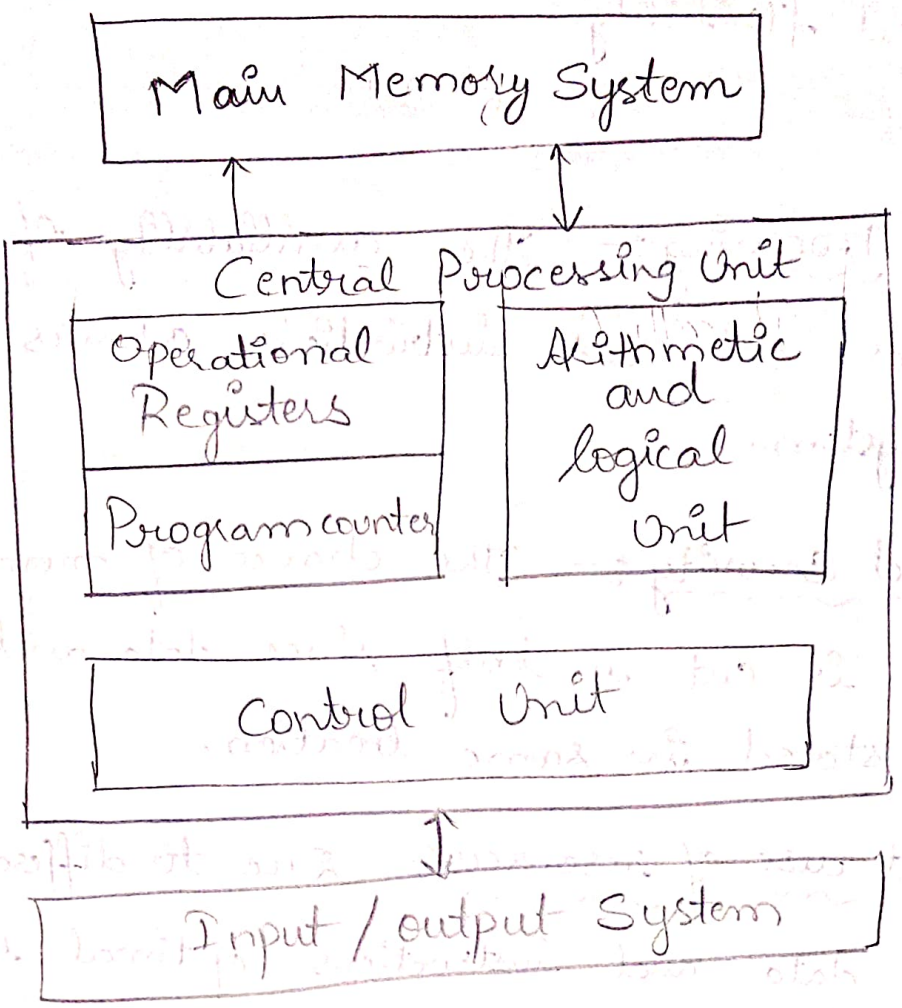
3) Flexibility :- A program can always be changed or altered without any physical changes in circuitry.

Disadvantages :-

1) Bottleneck issues :- Sharing bus for both data and control instructions can make the operations slow.

2) Memory corruption :- Since data and instructi reside in same memory, there could be possibility of one erasing one data on the other.

Architecture :-



HARVARD ARCHITECTURE :-

It is a digital computer architecture whose design is based on the concept where seperate storage and seperate buses are used for instruction and data.

Features :->

1) Seperate memory spaces
2) Fixed instruction length.
3) Parallel instruction and data access.
4) More efficient memory usage.

5) Suitable for embedded systems.

6) Limited flexibility.

Advantages:-

1) Faster Processing:- The availability of two separat buses for data and instructions enhances the velocity of the system.

2) Improved Security:- The chance of memory corruption is cut in half since data and instructio are not stored in same location.

3) Efficient use of resources:- Due to different memory sizes for data and instructions optimal use of buses are done.

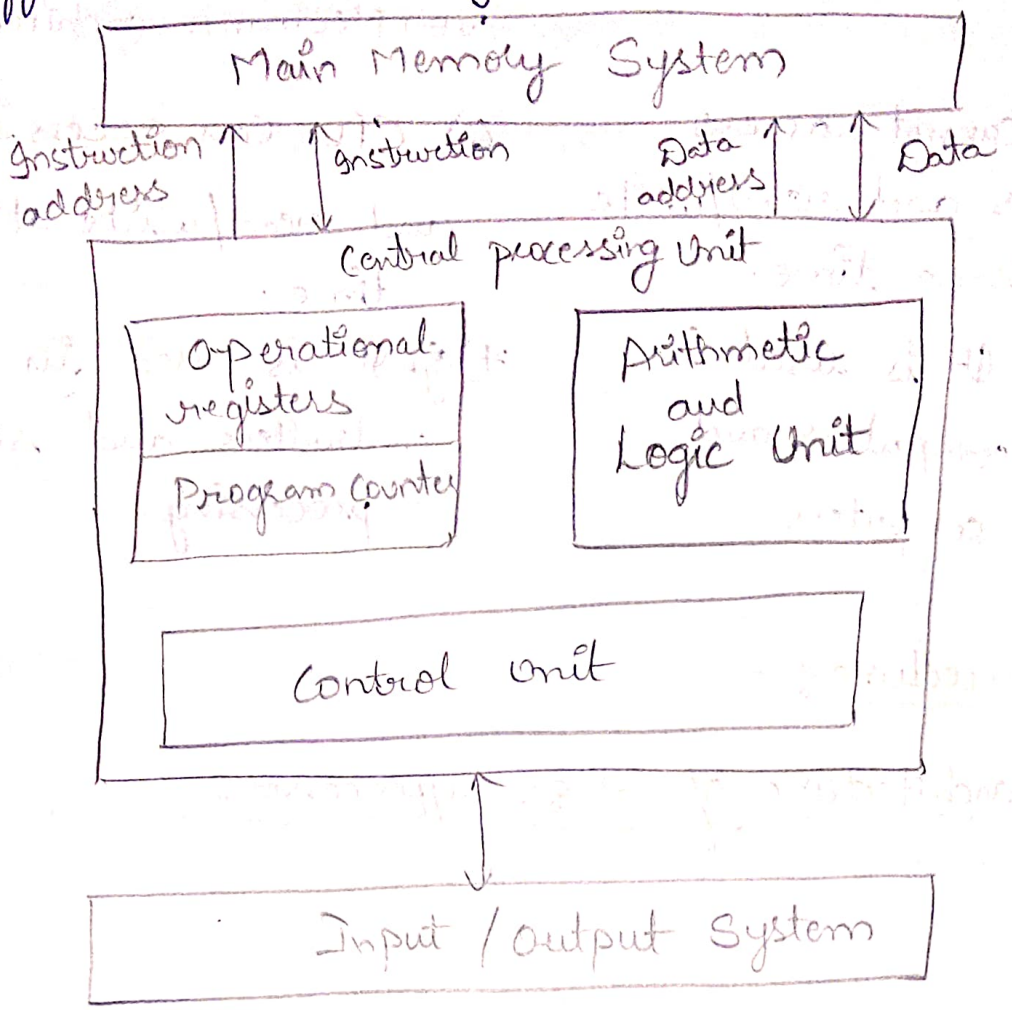
Disadvantages:-

1) Complexity:- The design and implementation of this architecture requires other hardware facilities.

2) Higher Cost:- Due to two sets of memory and seperate buses, the implementation costs high tha Von-Neumann architecture.

3) Less Flexibility competitors:- Changing or improving the system can be little tricky because of

different memory regions.



Differences between Von Neumann & Harvard architecture :-

Von Neumann Architecture	Harvard Architecture.
1) Ancient computer architecture based on program computer concept.	1) modern computer architecture based on relay based model.
2) Same physical memory address is used for instructions and data.	2) Separate physical memory address is used for instructions and data.
3) Common bus is used for both data and instruction transfer.	3) Seperate buses are used for data and Instruction transfer.
4) Two clock cycles are used to execute single instruction.	4) Single clock is used for one Sigle instruction execution.

5) It is cheaper in cost.

6) CPU cannot access instructions and read/write at the same time.

7) Eg:- It is used in personal computers and small computers.

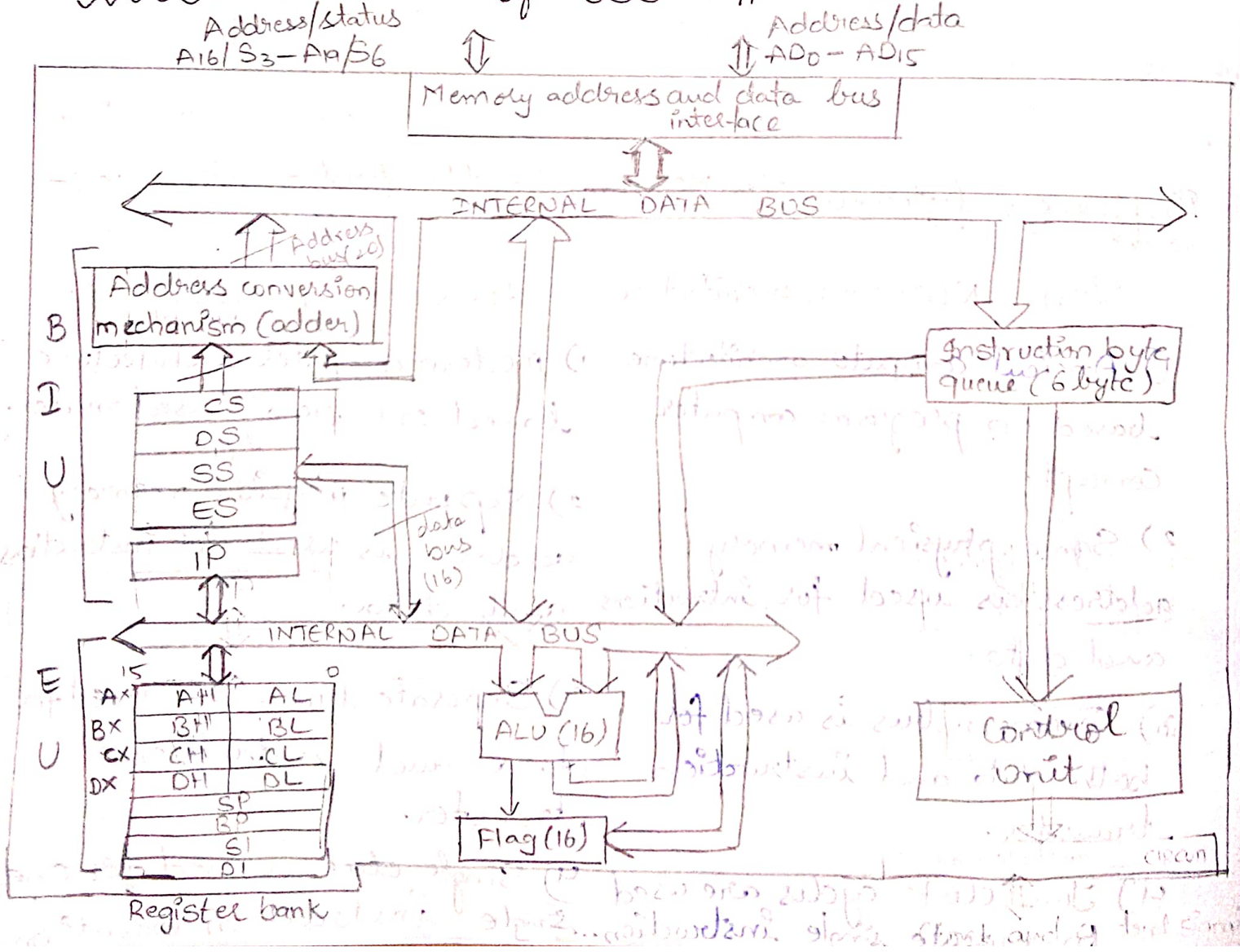
5) It is costlier than Von Neumann architecture.

6) CPU can access instructions and read/write at the same time.

7) Eg:- It is used in micro controllers and signal processing.

⇒ 8086 Architecture:-

(i) Internal architecture of 8086 processor:-



The internal block diagram describes the overall organization of different units inside the chip.

1) The complete architecture is divided into two parts
   a) BIU (bus interface unit)
   b) EU (Execution unit)

Execution Unit :- It includes the ALU, eight 16-bit general purpose registers, 16-bit flag register and control unit.

1) The 8, 16-bit general purpose registers can be used to store 8-bit or 16-bit data during program execution. Each register has some special functions also.

(i) AX/AL :- AX or AL is used as an accumulator. It is used in ALU or operation or I/O operations or some decimal & ASCII operations.

(ii) BX :- It holds the offset address of a location in the memory.
It is also used to refer a data in the memory using look up table technique with the help of XLAT instruction.

iii) CX/CL :- * It holds the count value while executing the repeated string instructions and the loop instruction.
* CL is used to hold the count of shift value during shift/rotate operation.

(iv) DX :- * It is used to hold a part of result dur a multiplication operation and a part of dividend during division operation.

* It also hold I/O device address while executing IN & OUT instructions.

(v) SP :- * Stack pointer hold the offset address of the data stored at the top of the stack segment(S

* SP along with stack segment in BIU unit is used to decide the address at which the data is to be pushed or popped during PUSH or POP instruction is executed.

(vi) BP :- Base pointer. It is used to hold the offset address of the data to be read from or written into the stack segment.

(vii) SI :- (Source index) :- holds the offset address of the source data in the data segment, while executing string instructions.

(viii) DI :- (Destination index) :- holds the offset address of destination data in the extra segment, while executing string instructions.

Flag registers:- 8086 has 16-bit flag register divided into two parts

(i) Condition flag or Status flags (lower byte of 16-bit flag)

(ii) Control flags. (higher byte of 16-bit flag).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	O	D	I	T	S	Z	X	AC	X	P	X	C

Control flags — Status flags

(i) S- Sign flag:- Set = 1 = the result is negative.
Reset = 0 = the result is positive.
Used for signed computation equals to MSB of the result.

(ii) Z- zero flag:- Set = 1 = the result of AL operation is zero
0 = the result of AL operation is non zero.

(iii) P- Parity flag:- 1 = Set = when the lower byte of the result has odd parity.
0 = Reset = when the lower byte of the result has even parity.

(iv) AC- Auxilliary carry flag:- 1 = Set = If there is a carry in the lower nibble during addition or borrow for subtraction.
0 = Reset = If there is no carry or borrow during addition or subtraction operation.
Mostly used during BCD operations.

(v) Carry flag:- Set = 1 = If carry is generated after 8-bit or 16 bit addition or subtraction operation.

Reset = 0 :- If no carry is generated after 8-bit or 16-bit addition or subtraction operation

O- Overflow flag :- If two signed numbers are added or subtracted, an overflow may occur.

∴ 1 = Set = If the machine exceeded the storage capacity.

0 = Reset = If the machine is not overloaded.

D- Direction flag :- It is used by string manipulation instructions.

1 = Set = The string is processed from higher address to lower address. (auto-decrement mode)

0 = Reset = The string is processed from lower address to higher address (auto-increment mode)

I = Interrupt flag :-

Set = 1 = Any maskable interrupts are recognised by the CPU; ie., interrupt pin is enabled. otherwise it is zero ie., pin (INTR) is disabled

T- Trap flag :- If it is Set = TF = 1 : then 8086 is interrupted after each instruction execution (single Step execution). If it is reset = TF = 0 ; then trapping is disabled.

Bus Interface unit:- 1) It contains the circuit for physical address calculations and a pre decoding instruction byte queue (6 byte long).

2) This unit establishes communication b/w external devices & peripherals including memory via bus.

3) The complete physical address 20 bits long is generated by using segment and offset address each 16-bits long.

For example:- Segment address = 1005H

Offset address = 5555H

∴ Physical address = segment address (10H) + offset addr

= (1005)(10H) + 5555H

= 10050 + 5555 = 155A5H

4) The complete 1 MEGABYTE of memory are addressed by 16 logical segments in 8086. Each segment is about 64KB of memory.

There are 4 segments basically

1) CS (code segment register):- Used to address memory location in the code segment where executable programs are stored.

2) DS (Data segment register):- Used to address memory locations of dagta segment where data resides.

3) Extra segment register (ES):- It also contains address locations of data.

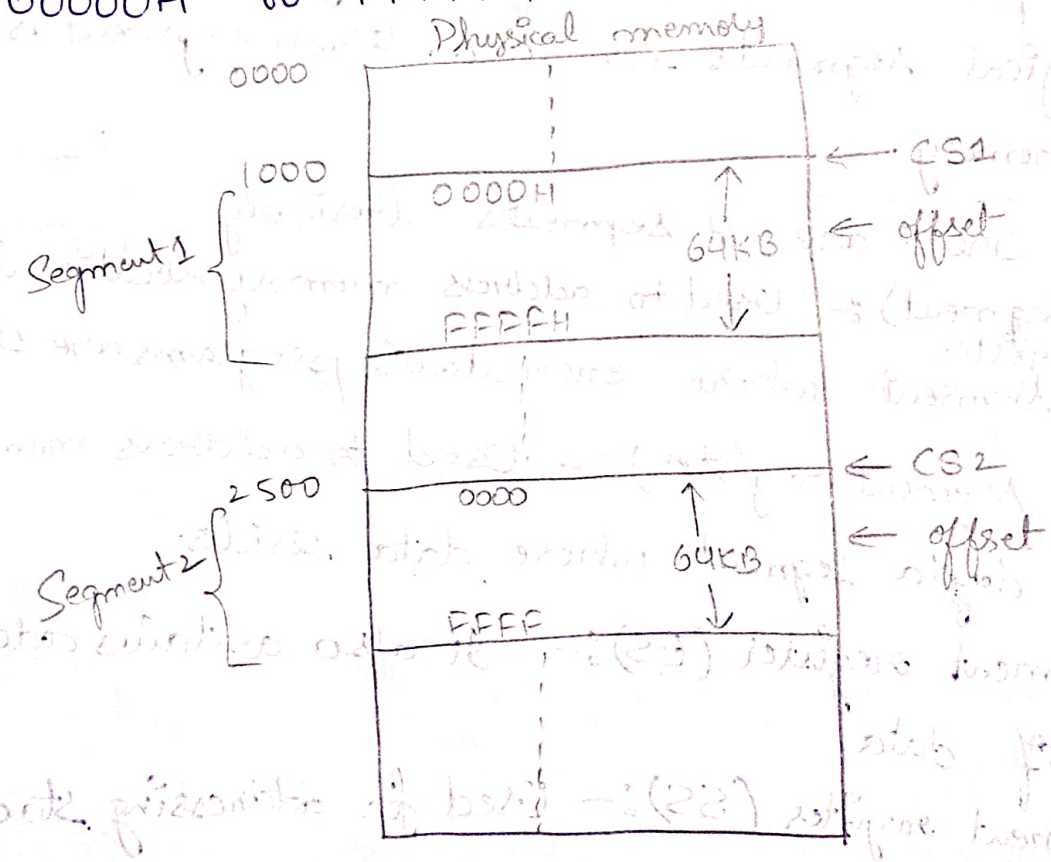
4) Stack segment register (SS):- Used for addressing stack

segment of memory. CPU uses stacks for storing data while addressing any location in the memory bank, 20 bit physical address is used, which is calculated using 16 bit segment address and 16 bit offset address.

Segment address resides in segment register and offset address resides in IP, AX, BX, CX, DX, SI, DI, SP, BP depending on addressing modes.

The addresses of segment are assigned from 0000H to F000H and offset addresses are assigned from 0000 H to FFFFH.

Therefore the physical address ranges from 00000H to FFFFFH

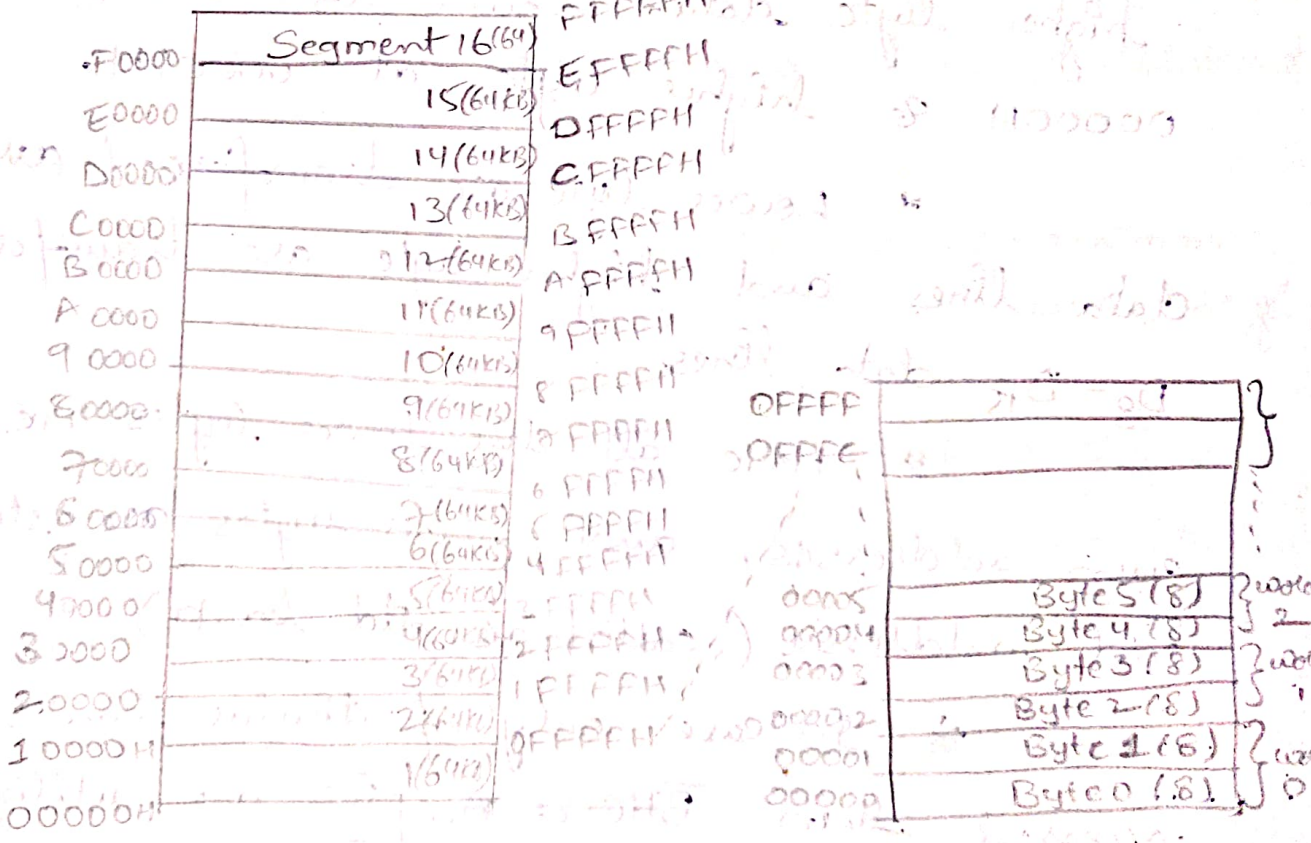


Physical Memory Organisation :-

* On 8086 (2^{20}) 1MB of physical memory is organised as an odd bank and an even bank each of 512 kBytes.

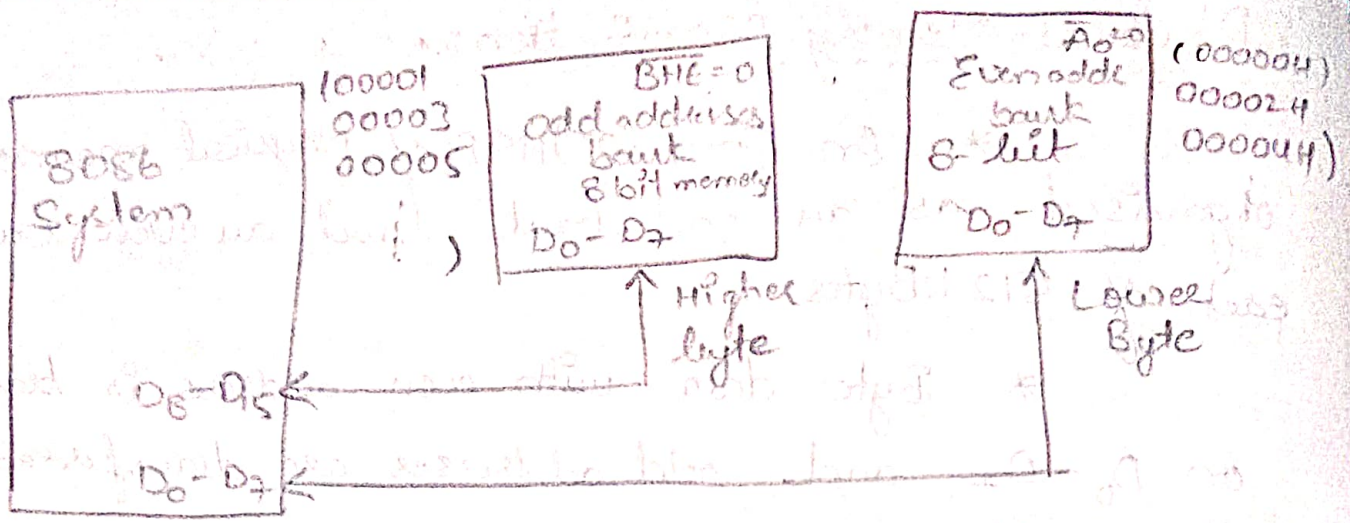
* Byte data with even address is transferred on $D_0 - D_7$ and odd addresses are transferred on $D_{15} - D_8$ lines.

* $\overline{BHE}$ and A_0 pins are used to select either even or odd or both banks.



→ Each location consists of a byte of data and two locations combinely forms a word data.

This complete system byte memory addresses are divided into two memory banks as in fig below,



Physical memory organization

∴ The total 16 bit is divided as lower byte & higher byte data. Lower byte are stored at 00000H & higher byte at 00001H.

* Lower byte are transfered over $D_0 - D_7$ data lines and higher byte are transfered over $D_8 - D_{15}$ data lines

* Here all the lower bytes are stored at even addresses, and higher bytes are stored at odd addresses (called as odd banks).

* In case only particular banks are to be accessed then $\overline{BHE}$ & A_0 pins are utilized as follows

$\overline{BHE}$	A_0	Indication
0	0	Both are active, 16 bit data transfer $AD_{15} - AD_0$
0	1	Only higher (odd bank) active, 1 byte data on $AD_{15} - AD_8$
1	0	Only lower (even bank) active, 1 byte data on $AD_7 - AD_0$

1 1 — No bank active.

* The location of FFFF0H to FFFFFH are reserved for jump, initialization of programme and I/O process initialization.

* 00000H to 003FFH are reserved for interrupt vector table which handles 256 types of interrupts.

General Bus operation:-

* 8086 has a combined address and data bus commonly refered to as a time multiplexed address and data bus.

* Reason for multiplexing, is to maximum utilization of processor pins and facilitates 40 pin use of IC package of 8086.

Let us consider the 40-pins of IC 8086 in minimum and maximum mode and its specification as well.

$AD_{15}-AD_0$ → These are multiplexed memory I/O and Data lines. Address remains on bus during T_1, while data is available during T_2, T_3, T_w, T_4 (T_w - wait cycle).

T_1, T_2, T_3, T_4 are machine cycles.

$A_{19}/S_6, A_{18}, S_5, A_{17}/S_4, A_{16}/S_3$:- Multiplexed address and status lines, ① During T_1 it is significant address lines

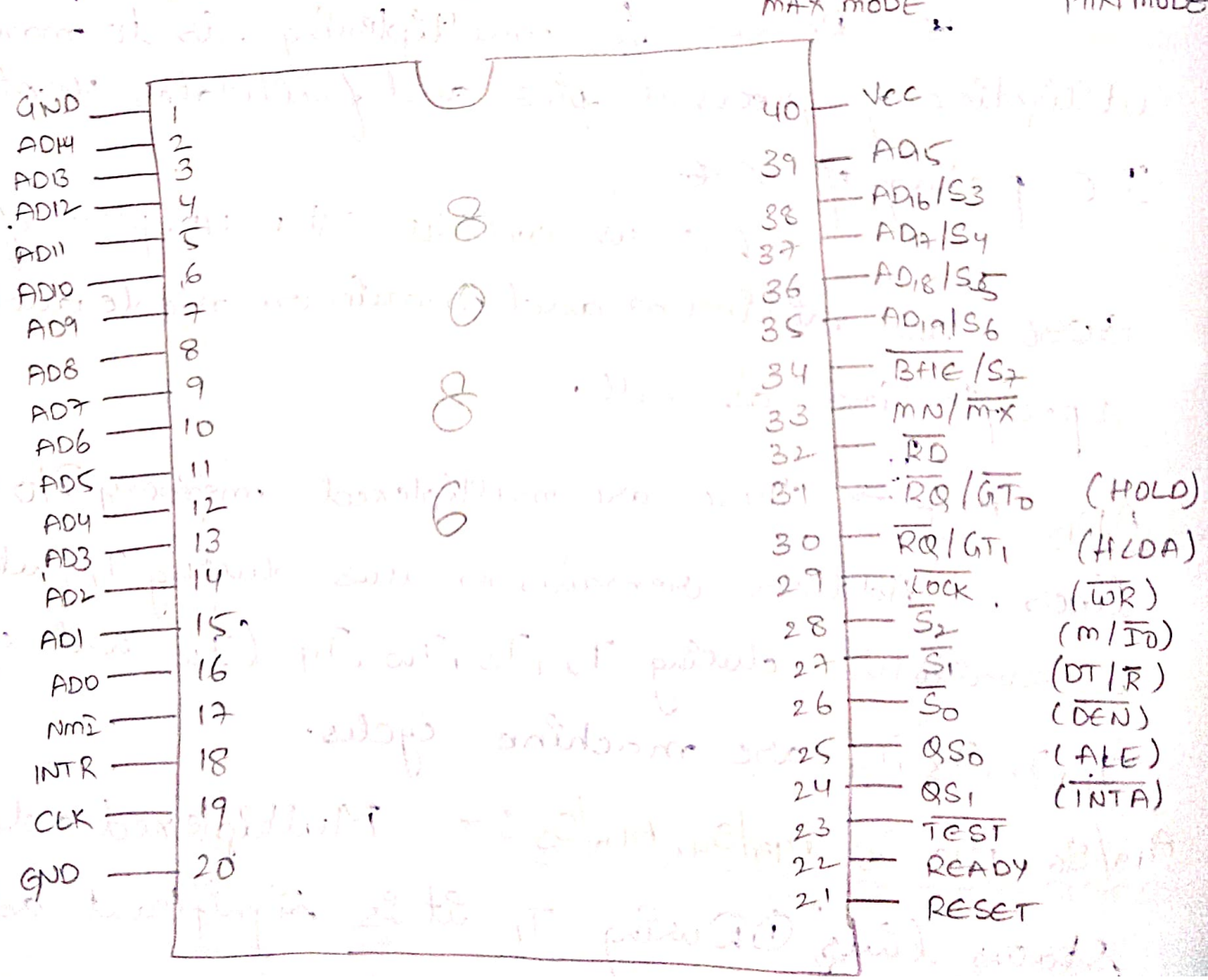
for memory operations.

(2) During I/O operations, all the lines are low.

3) Interrupt enable flag bit is displayed on S_5 line at the beginning of each cycle.

4) S_3 & S_4 indicates segment register along with memory accesses as below table;

S_4	S_3		Indications.
0	0	→	Extra segment
0	1	→	Stack Segment
1	0	→	Code or none.
1	1	→	Data segment.



$\overline{BHE}$ / S_7 - Bus high Enable/Status :-

* $\overline{BHE}$ is active low pin and is used send data over higher order (D_{15}-D_8) data bus.
* It is inactive at T_1 cycle.
* S_7 is not currently in use.

$\overline{RD}$ - Read :- If it low it indicates that processor is performing a memory or I/O read operation.

Ready :- This is an acknowledgement given to 8086 by slower devices or memory that they have completed the data transfer.

INTR - Interrupt request :- It is an input - which indicates the availability of the request.

$\overline{TEST}$:- If test pin goes low, the execution will continue, else it will remain idle.

NMI :- (Non maskable interrupt) :- If this pin is enabled, it indicates this interrupt is not maskable (not avoided)

RESET :- The input indicates to terminate the current activity and start execution from FFFF0H.

Clock (CLK) :- It provides basic timing for processor operation. 8086 provides 5MHz to 10MHz freq.

V_{CC} :- +5V power supply for operation of internal circuit.

GND :- Ground for the internal ckt.

MN / $\overline{MX}$:- 0 - indicates to operate in maximum mode.
1 - indicates to operate in minimum mode.

MINIMUM MODE operation of 8086 :-

Memory / $\overline{I/O}$:- If it is 1 → indicates CPU is having memory operation.
If it is 0 → indicates CPU is having I/O operation.

$\overline{INTA}$:- Interrupt acknowledge :- If it is low it indicates that the Interrupt has been accepted.

ALE :- Address latch enable :- If it is 1 it indicates that the valid address is on the address/data line.

DT / $\overline{R}$:- (Data transit or Receive) :-
- If the value is 1 it indicates transit operation.
- If the value is 0 it indicates read operation.

$\overline{DEN}$:- (Data Enable) :- This signal indicates that the data is available over address/data line.

Hold / HLDA :- If HOLD pin is high, it indicates that another master is requesting for bus access.
* If HLDA is high, it indicates that the processor has released the bus for access.

MAXIMUM MODE PINS :-

$\overline{S_2}$, $\overline{S_1}$, $\overline{S_0}$:- These are the status lines which indicates the type of operation carried out by the processor.

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Indication
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	write I/O port
0	1	1	Halt.
1	0	0	Code access.
1	0	1	Read memory
1	1	0	Write memory.
1	1	1	Passive.

$\overline{LOCK}$:- Whenever this pin is activated. it indicates that the CPU is executing critical instruction which requires bus and bus cannot be released.

QS_1, QS_0 - Queue status :- The operation of Queue status is given in below table.

QS_1	QS_0		Indication
0	0	→	No operation for execution.
0	1	→	First byte of opcode from the queue.
1	0	→	Empty queue.
1	1	→	Susequent byte from the queue.

$\overline{RQ}/\overline{GT_0}$, $\overline{RQ}/\overline{GT_1}$:- (Request/Grant) :- Used by local bus masters in max mode to force the processor to release the bus.

RQ/GT0 is having high priority than RQ/$\overline{GT_1}$. And it is a bidirectional pin which works with HOLD, HLDA pin. synchronously.

* GENERAL BUS OPERATION :-

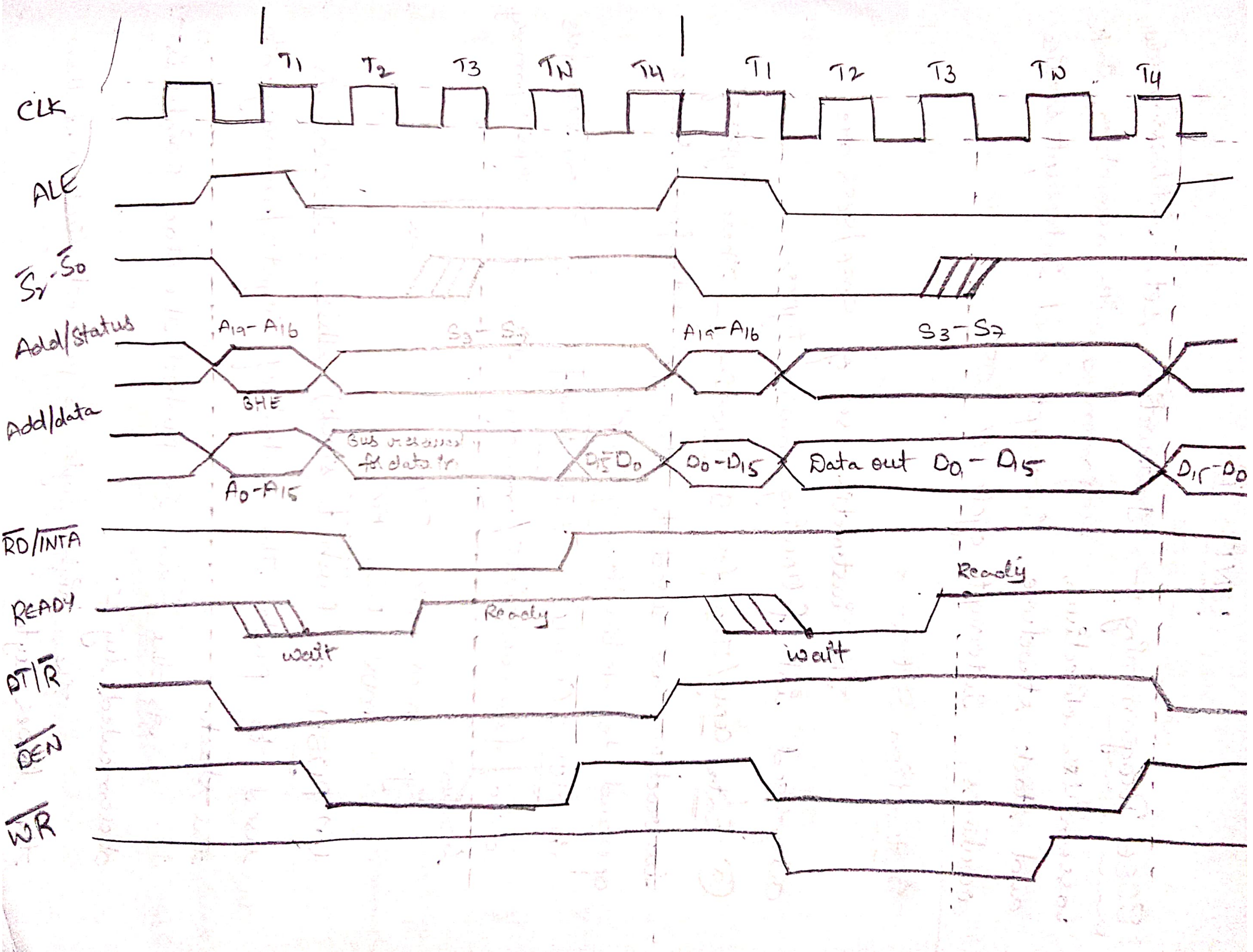
Basically all the processor bus cycles consists of at least 4 clock cycles referred as T_1, T_2, T_3 & T_4.

At T_1 → address location is transmitted by the processor.

At T_2 → data read cycle is initiated.

At T_3 & T_4 → Data transfer takes place.

In case, the address device is slow, it shows "NOT READY" status, then Tw time is inserted b/w T_3 & T_4.


T1
T2
T3
TW
T4
T1
T2
T3
TW
T4
CLK
ALE
$\overline{S_7}$-$\overline{S_0}$
Add/status
A19-A16
S3-S7
A19-A16
S3-S7
BHE
Add/data
Bus released for data in
A0-A15
D0-D15
Data out D0-D15
D15-D0
$\overline{RD}$/$\overline{INTA}$
READY
Ready
Ready
wait
wait
DT/$\overline{R}$
$\overline{DEN}$
$\overline{WR}$