Unit No: 5

80386 Microprocessor:-

*) Introduction to 80386 microprocessor:-

The 80386 is an advanced 32-bit microprocessor optimized for multitasking operating systems and designed for applications needing very high performance.

The 32-bit registers and data paths support 32-bit addresses and data types.

The processor can address upto 4GB of physical memory and 64TB of virtual memory

It has two versions

80386 DX

80386 SX

The original 80386 is 80386 DX. The 80386 SX version was introduced in 1988 as low cost alternative.

The internal architecture of SX and DX are same

The 80386 has three operating modes

Real address mode

Protected mode

Virtual 8086 mode

The 80386 DX consist of 32-bit address bus, and 32-bit data bus

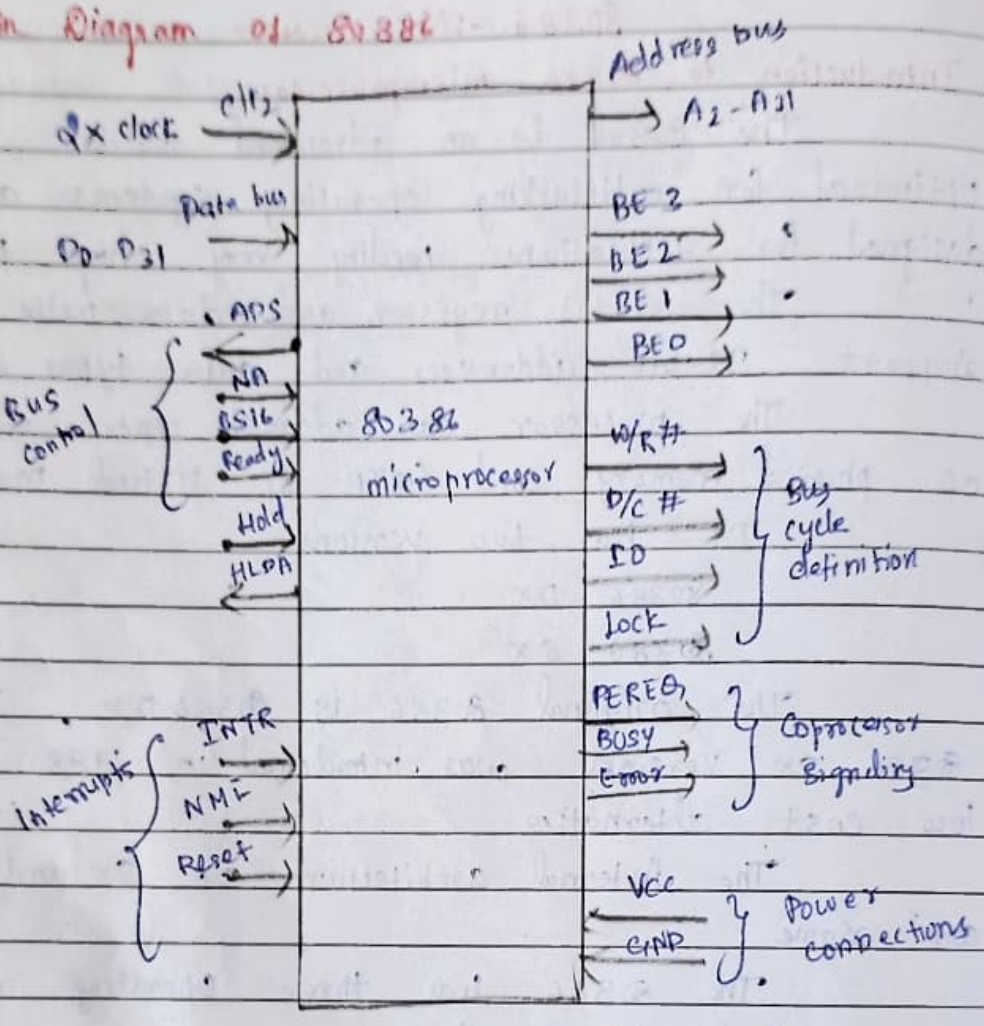
The 80386 SX consist of 24-bit address bus and 16-bit data bus

The 80386 DX used in note book (or) laptop computers

The 80386 SX used in early personal computers

80386 requires +5V power supply for operation.

a) Pin Diagram of 80386



$W/\overline{R}$:- The write/read output distinguishes the write and read cycles from one another

$D/\overline{C}$:- This data/control output pin distinguishes between a data transfer cycle from a machine control cycle like interrupt acknowledge

$M/\overline{IO}$:- This output pin differentiates between the memory & input/output cycles

$\overline{LOCK}$:- The Lock output pin enables the CPU to prevent the other bus masters from gaining the

control of the System bus.

$\overline{NA}$:- The next address input pin, if activated allows address pipeling during 80386 bus cycles.

$\overline{ADS}$:- The address status output pin indicates that the address bus and bus cycle definition pins (W/$\overline{R}$, D/$\overline{C}$, M/$\overline{IO}$, $\overline{BE0}$, to $\overline{BE3}$) are carrying the respecting valid signal.

$\overline{READY}$:- The ready signal indicates to the CPU that the previous bus cycle has been terminated and the bus is ready for the next cycle. The signal is used to insert WAIT states in a bus cycle and is useful for interfacing of slow devices with CPU.

VCC :- These are system power supply lines.

VSS :- These return lines for the power supply

$\overline{BS16}$:- The bus size-16 input pin allows the interfacing of 16 bit devices with the 32 bit wide 80386 data bus.

Successive 16 bit bus cycles may be executed to read a 32 bit data from a peripheral

HOLD :- The bus hold input pin enables the other bus masters to gain control of the System bus if it is asserted.

HLDA :- The bus hold acknowledge output indicates that a valid bus hold request has been received and the bus has been relinquished by the CPU.

$\overline{BUSY}$:- The busy input signal indicates to the CPU that the coprocessor is busy with the allocated task.

$\overline{ERROR}$:- The error input pin indicates to the CPU that the coprocessor has encountered an error while executing its instruction.

PEREQ :- The processor extension request output signal indicates to the CPU to fetch a data word for the coprocessor.

INTR :- This interrupt pin is a maskable interrupt that can be masked using the IF of the flag register

NMI :- A valid request signal at the non-maskable interrupt

RESET :- A high at this input pin responds the current operation and restart the execution from the starting location.

N/C :- No connection pin are expected to be left open while connecting the 8086 in the circuit.

The Memory System of 80386

The physical memory system of the 80386 DX is 4G bytes in sizes and is addressed as such. Virtual addressing is used. 64T bytes are mapped into the 4G bytes of physical space by the memory management unit and descriptors

That Virtual addressing allows a program to be larger than 4G bytes if a method of swapping with a large hard disk drive exists

The memory is divided into four 8-bit wide memory banks. each containing upto 1G bytes of memory

The 32-bit wide memory organization allows bytes, words, (or) double words of memory data

The 80386 DX transfers upto 32-bit wide number in a single memory cycle. the 8085 requires four cycles to accomplish the same transfer and the 80286 and 80386 SX require two cycles

Each memory byte is numbered in hexa decimal as they were in prior version of the family

The difference is that the 80386 DX uses a 32-bit wide memory address, with memory bytes numbered from location 00000000H to FFFFFFFFH

The two memory banks in the 8086, 80286 and 80386 SX system are accessed via A0 on the 8086 and 80286. The 80386 DX memory banks are accessed to four bank enable signals

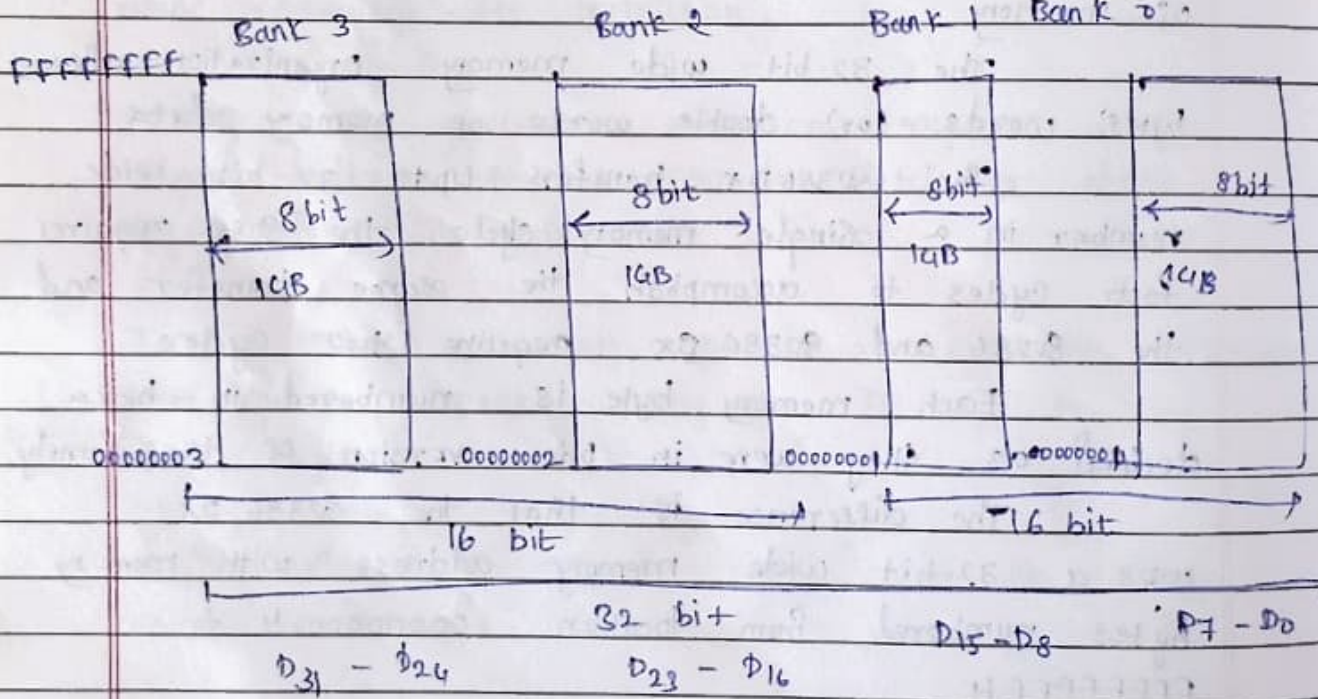
A word is addressed in banks 0, and 1 or in banks 2 and 3.

Memory location 00000000H in bank 0. location 00000001H is in bank 1. location

00000002H is in bank 0 and location 00000003H in bank 3.

The 80386DX does not contain address connections A0 and A1 because these have been encoded as the bank enable signals.

There are three types of memory system
1) Buffered system
2) Pipeline and Cachies
3) Interleaved memory system.



The physical memory address of 80386 ranges from 00000000H to FFFFFFFFH.

Buffered system:

The 80386DX connected to buffer that increase fan-out from its address, data and control connections

This microprocessor is operated at 25MHz using a 50MHz clocks input signal that is generated by an integrated oscillator mode

The HLDA signal is used to enable all buffers in a system that uses direct memory access

otherwise, the buffer enable pins are connected to ground in a non-DMA system

Pipeline and caches :-

A pipeline is a technique used in a advanced microprocessor where the microprocessor begins executing a second instruction before the first has been completed

That is several instruction are pipeline simultaneously, Each at a different processing style

The pipeline is divided into segments and each segment can execute and operation concurrent with the other system.

When the system segment complete its operation it passes the result to the next segment in an pipeline and fetch as the next operation from the presiding segments

The final result to each instruction emerges at the end of pipeline

The pipeline is an interfacing memory because the 80386 supports pipelined memory access

The pipelining allow memory and extra clocking period to access data.

The extra clock extensions the access time to 156 ns on 80386 operation with 16MHz.

Interleaved Memory System :-

An interleaved memory system is another method of improving the speed of a system.

Its only disadvantage is that it costs considerably more memory because of its structure.

An interleaved memory system requires two or more complete sets of address buses and a controller that provides address for each bus.

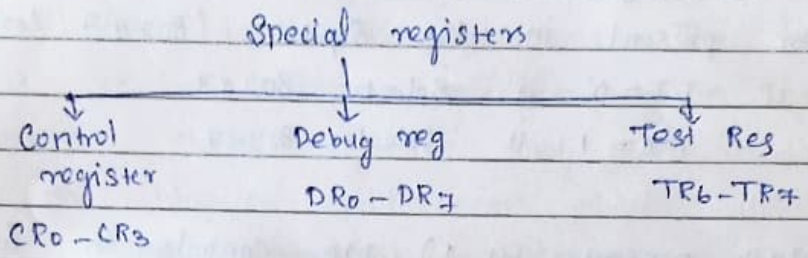
Systems that employ two complete buses are called a two-way interleave.

Systems that use four complete buses are called a four way interleave.

Interleaving increases the amount of access time provided to memory because the address is generated to select the memory before the microprocessor accesses it.

Special registers of 80386 :-

There are three special type of registers in 8086. These are discussed below in details

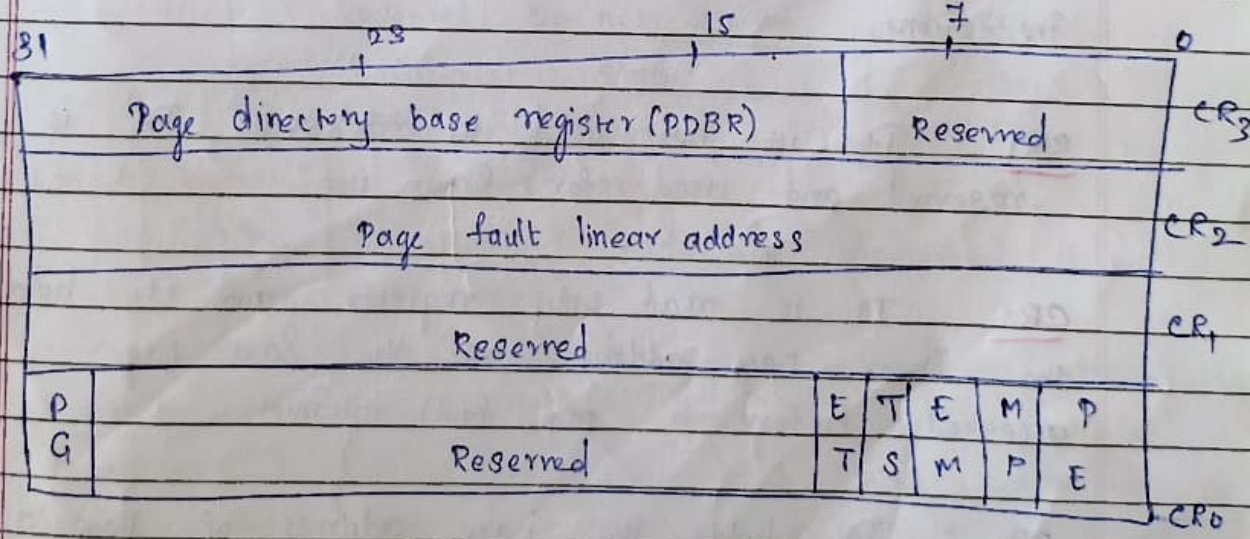


Control Registers:-

A Control register is a processor register which controls the general behavior of a CPU. Common tasks performed by control register include interrupt control, switching the addressing mode and paging control

The protected mode includes the 4 system control registers, identified as CR_0 to CR_3

These are all 32-bit registers



CR_0 Contains system control flags, which control (or) indicate conditions that apply to the system as a whole, not to an individual task

→ EM (Emulation, bit 2) EM indicates whether coprocessor functions are to be emulated

→ ET (extension type, bit 4) ET indicates the type of coprocessor present in the system (80287 (or) 80387)
if ET = 0 it selects 80287
ET = 1 it selects 80387

→ MP (Math present, bit 1) MP controls the function of the wait instruction, which is used to coordinate a coprocessor.

→ PE (Protection Enable) Setting PE cause the processor to begin executing in protected mode. Resetting PE returns to real-addressing mode

→ PG indicates whether the processor uses page tables to translate linear address in to physical addresses

→ TS The processor sets TS with every task switch and tests TS when interpreting coprocessor instructions

CR1 It is not used in 80386 DX. It is reserved and used for future use

CR2 It is read only register and it holds the linear page address of the last page accessed before a page fault interrupt.

CR3 It holds the base address of page directory. It is also called page directory base register (PDBR)

Debug Registers :-

Debug registers are 32-bit registers. There are total 8 debug registers in 8386.

The first four debug registers contain 32 bit linear breakpoint addresses. The linear address is a 32-bit address generated by a microprocessor instruction that may (or) may not be the same as the physical addresses.

The breakpoint addresses, which may locate an instruction (or) datum, are constantly compared with the addresses generated by the program.

Linear Breakpoint address 0	DR_0
Linear breakpoint address 1	DR_1
Linear breakpoint address 2	DR_2
Linear breakpoint address 3	DR_3
Reserved. Do not define	DR_4
Reserved. Do not define	DR_5
Breakpoint status	DR_6
Breakpoint control	DR_7

Linear breakpoint address registers :-

The breakpoint addresses specified are 32-bit linear addresses

$DR_0 - DR_3$:- while Debugging, intel 386 hardware continuously compares the linear breakpoint addresses in $DR_0 - DR_3$ with the linear addresses generated by executing software. If match found an exception 1 (debug fault) is generated.

The debug address registers are effective whether (or) not paging is enabled.

The addresses in these registers are

linear addresses. If paging is enabled, the linear addresses are translated into physical addresses by the processor's paging mechanism

If paging is not enabled, these linear addresses are the same as physical addresses.

Debug control Register:- (DR7)

The debug control registers helps to define the debug condition and selectively enables and disables those conditions.

00 - Break on instruction execution only
01 - Break on data writes only
10 - Undefined
11 - Break on data reads (or) writes but not instruction fetches.

Debug Status Register :- (DR6)

The debug status register permits the debugger to determine which debug conditions have occurred.

DR7	LEN 3	R/W 3	LEN 2	R/W 2	LEN 1	R/W 1	LEN 0	R/W 0	0 0	GE	LE	G3	L3	G2	L2	G1	L1	G0	L0
DR6	0 0 0 0 0 0 0 0 0 0									BT	BS	BD	0 0 0 0	B3	B2	B1	B0		
DR5	Reserved																		
DR4	Reserved																		
DR3	Breakpoint 3 linear address																		
DR2	Breakpoint 2 linear address																		
DR1	Breakpoint 1 Linear address																		
DR0	Breakpoint 0 Linear address																		

L1	G1	L0	G0
1	0	0	0

Test Registers :- TR6 & TR7

Two Registers TR6 and TR7 were provided for the purpose of testing.

TR6 was the Test command register and TR7 was the test data register.

These registers were accessed by variants of the Mov instruction.

A test register may either be the source operand or the destination operand.

The Mov instruction are defined in both real-address mode and protected mode

The test registers are privileged resources.

G	L
0	0

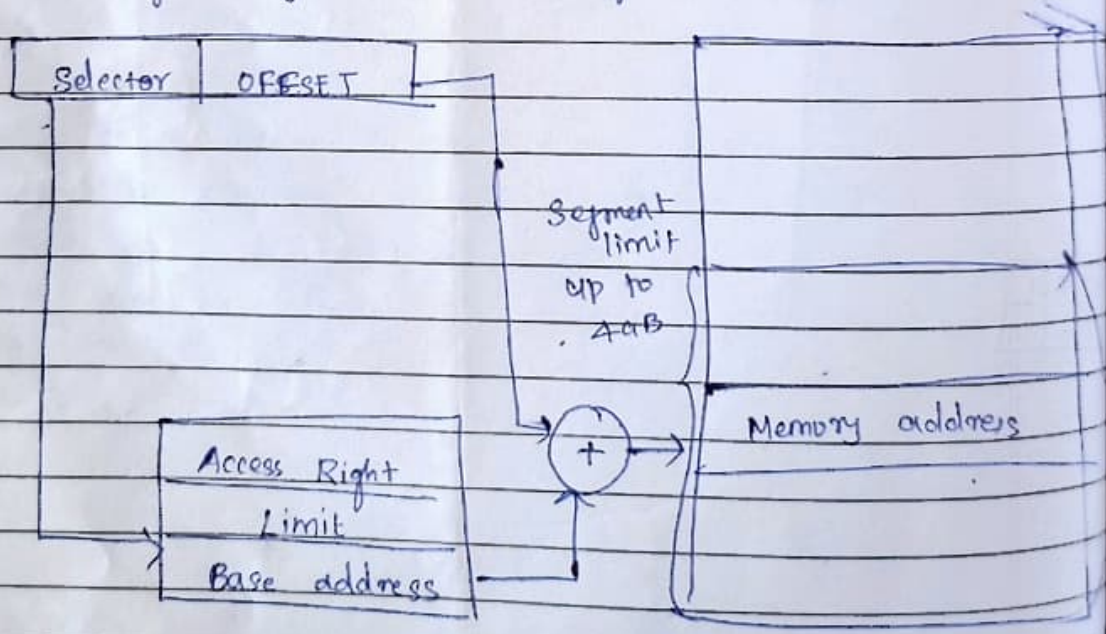
4) Introduction to 80386 protection mode :-

All the capabilities of 80386 are available for utilization in its protection mode of operation.

The 80386 in protection mode support all the software written for 80286 and 8086 to be executed under the control of memory management and protection abilities of 80386.

The protected mode allows the use of additional instruction, addressing modes, and capabilities of 80386.

Addressing in protected mode : In this mode, the ~~segment~~ contents of segment registers are used as selectors to address descriptors which contain the segment limit, base address and access rights byte of the segment.



Protected mode of 80386

The Selector is used to specify an index into a table defined by the operating system. The table includes the 32 bit base address of a given segment.

The physical address is obtained by summing the base address obtained from the table with the offset

e) Virtual 8086 mode:-

The 80386 allows the execution of 8086 program in both real mode and virtual mode i.e virtual mode.

Virtual 8086 mode provides the system designer the most flexibility of out of two modes (i.e) Real and protection virtual addressing mode.

The virtual mode allows the execution 8086 programs, by taking full advantage of the 80386 protection mechanism.

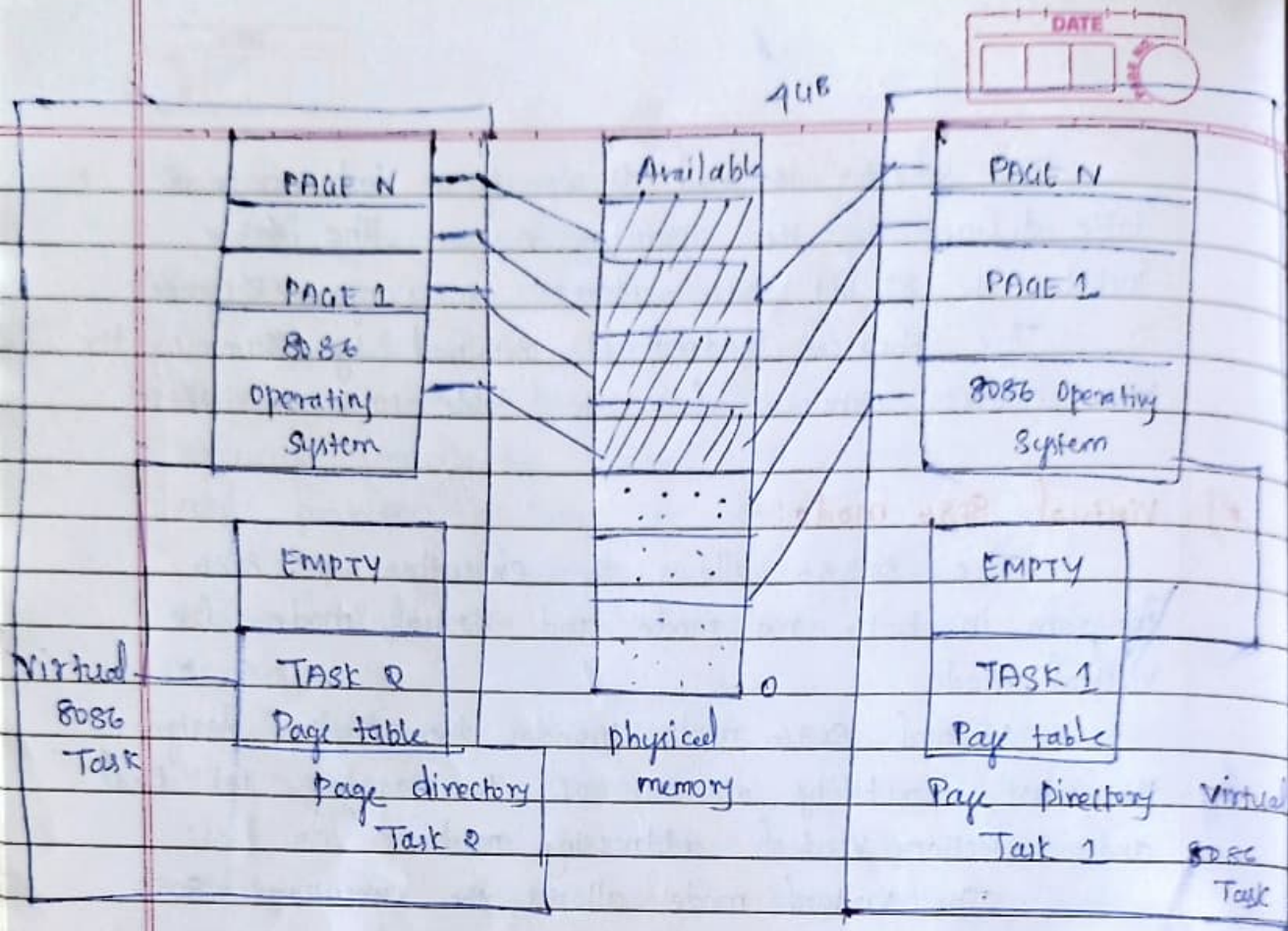
Hence it allows the simultaneous execution of 8086 OS with its application, and 80386 OS

The major difference between 80386 Real & protected mode is the way that segment selectors are interpreted.

When the processor is operating in virtual mode the segment registers are used in identical to real mode.

The 80386 gives facility to the OS for specifying which programs use 8086 style address mechanism, and which program use protected mode addressing on a per task basis.

By the use of paging memory management, the one MB address space of virtual mode task can be mapped anywhere in the 4GB of linear address space of 80386



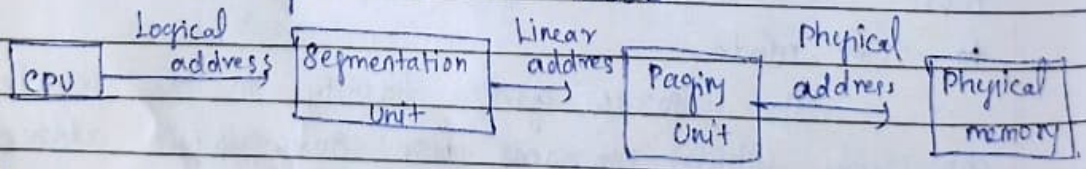
*) **Memory Management of 80386 :-**

The memory management of 80386 is divided into two

→ Segmentation

→ Paging

address generation in 80386



Paging :-

Paging is one of the memory management technique used for virtual memory multitasking operating system

The segmentation scheme may be divide the physical memory into a variable size segments but

the paging divides the memory into a fixed size pages. The segments are supposed to be the logical segments of the program but the pages do not have any logical relation with the program.

The pages are the just fixed size portions of the program module (or) data.

The advantage of paging scheme is that the complete segment of a task need not be in the physical memory at any time.

Only few pages of the segments, which are required currently for the execution need to available in physical memory.

Whenever the other pages of task are required for execution. they may be fetched from the secondary storage.

Thus paging mechanism provides an effective technique to manage the physical memory for multitasking system.

Paging unit :-

The paging unit of 80386 uses a low level table mechanism to convert a linear address by segmentation unit in to physical addresses.

The paging converts the complete map of a task into pages, each size 4K.

The task is further handled in terms of its pages rather than segment.

Paging unit handles every tasks in terms of three components namely page directory, page tables and pages itself.

Page directory :-

This is at most 4k bytes in size.

Each directory contains pointers to 1024 pages

31-12	11-9	8	7	6	5	4	3	2	1	0
Address	OS	0	0	D	A	0	0	U/S	R/W	P

P = 1 means pages is present in memory

R/W = 0 means Readonly for user

U/S = 1 means user page

A = 1 means page has been accessed

D = 1 means undefined for page directory entry

OS = bit can be used for LRU & are defined by OS

Memory Segmentation 80386

Logical address : Consist of Segment Selector and offset

The Selector is available in Segment register

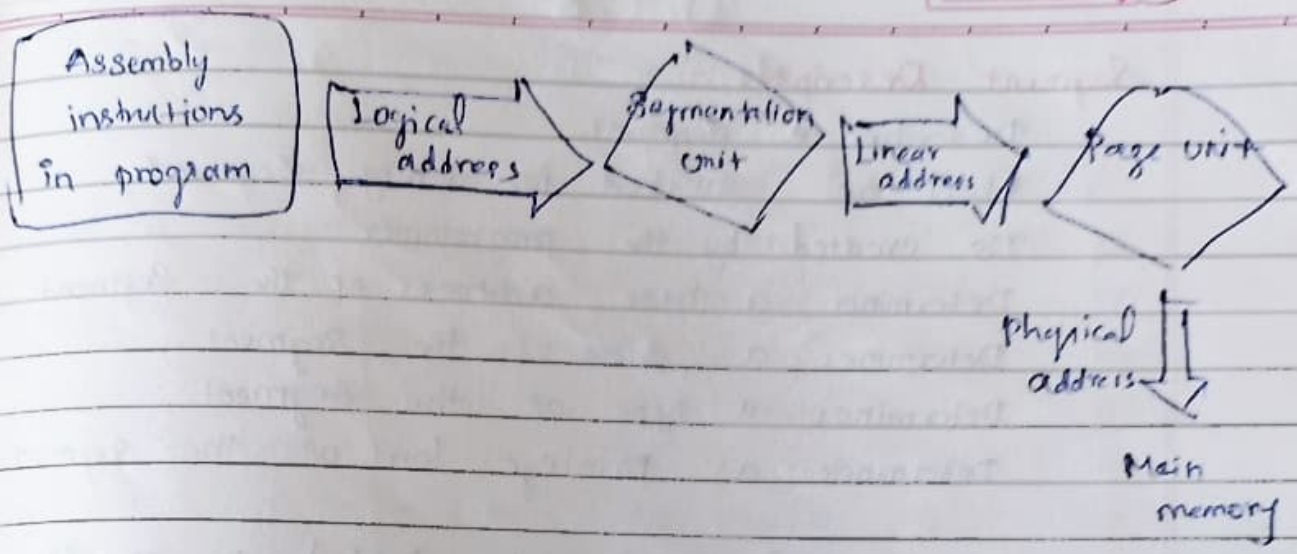
The offset is calculated by the processor by adding base, index, and displacement fields.

The Segmentation unit converts the logical address to linear address using the segment descriptor indicated by the Segment Selector.

Linear address :-

When the paging is disabled the linear address is actually the physical address.

When the paging is enabled the linear address is converted to physical address through paging



Selector Offset	Logical address
Directory Table offset	Linear address
	Physical address

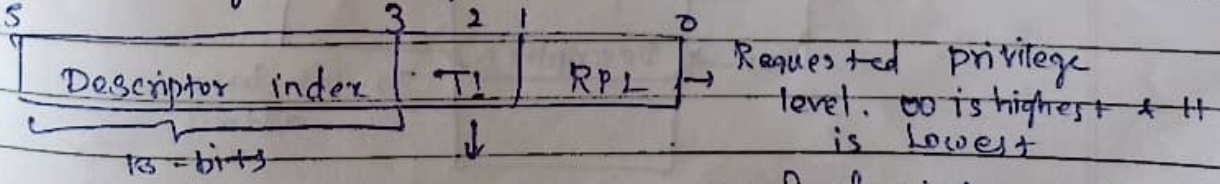
Segment Translation :-

To perform this translation, the processor uses the following data structures

- Descriptors
- Descriptor tables
- Selector
- Segment Registers

Segment Selector - 16 bit :-

A segment selector is loaded into a segment register (cs, ds, etc) to select one of the regular segments in the system as the one addressed via that segment register.



TI = 0 Global descriptor table

TI = 1 Local descriptor table

Segment Descriptor :-

Describes a segment
Must be created for every segment
Is created by the programmer
Determines a base address of the segment
Determines a size of the segment
Determines a type of the segment
Determines a privilege level of the segment

The physical memory is divided in small parts

2^n = physical memory where n = no. of address lines

80386 has 32 bit address lines so physical memory = 2^{32} = 4 bytes

Small part of memory is called segment
So dividing this 4GB of physical memory into segment is segmentation.

80386 has 6 segments.
Code segment (CS)
Data segment (DS)
Stack segment (SS)
Extra segment (ES) + FS, GS

Logical to Linear address Translation