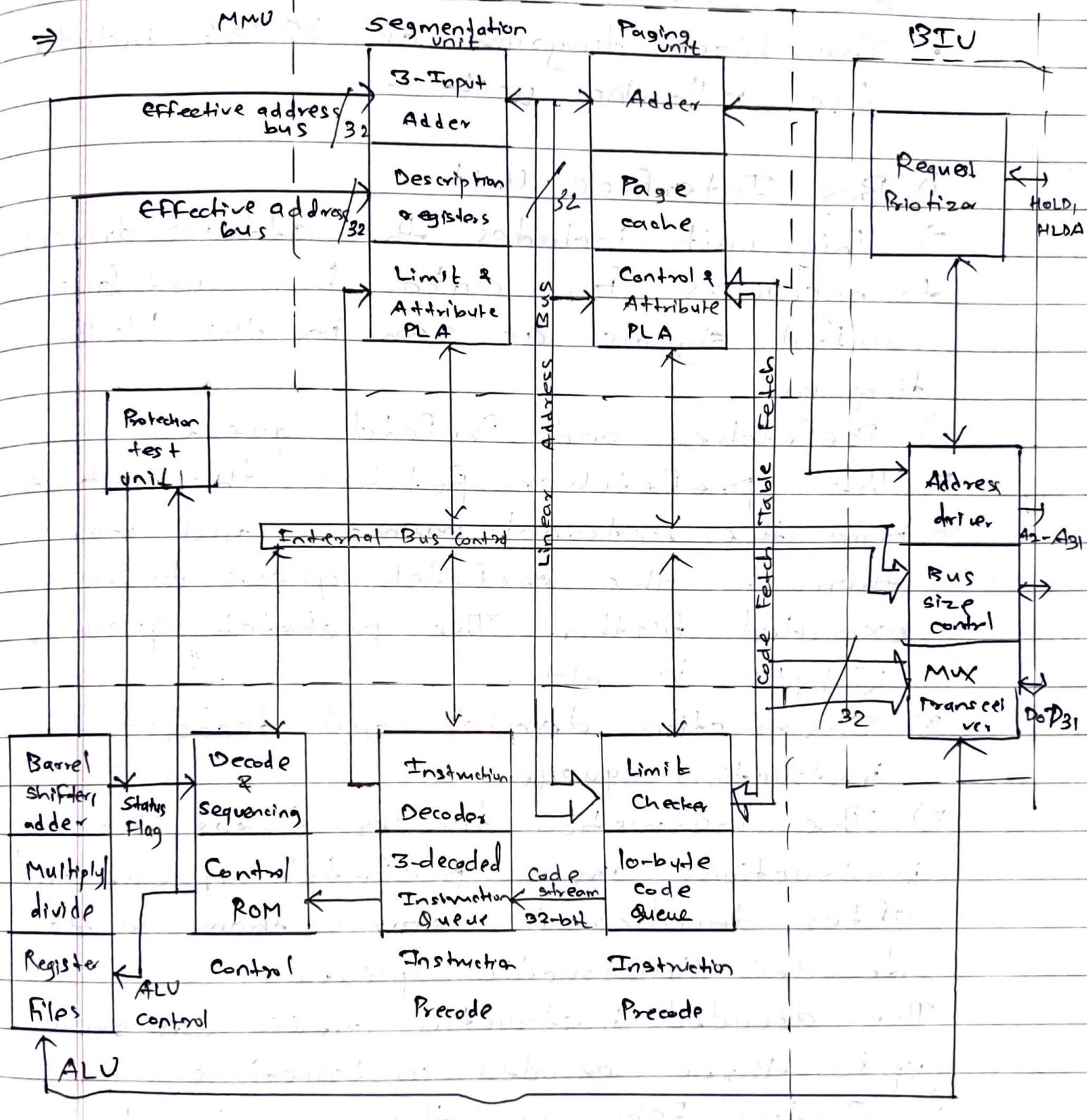


Q1 Draw and explain the block diagram of 80386 DX.



- The 80386 consists of a CPU, MMU, and BIU.
- The block diagram of 80386 includes the following units -

1) Bus Interface Unit

⇒ This unit includes the address drivers, transceivers for data bus and bus control signals as seen in the block diagram.

2) Prefetcher and Prefetch queue

⇒ The prefetcher fetches the instruction from the external memory and stores them in the prefetch queue to be executed further. The prefetch queue is 16-byte in size.

3) Instruction decoder and decoded instruction queue

⇒ The instruction decoder takes the instructions from prefetch queue and after decoding it, stores them in the decoded instruction queue.

The decoded instruction queue can store up to three decoded instructions.

4) Control ROM and the sequencing logic

⇒ The control ROM provides the control signals to be issued for corresponding instructions, which are then sequenced by sequential logic.

5) Execution Unit

⇒ The execution units includes a multiply unit, adder, barrel shifter and divide unit.

6) Protection Unit

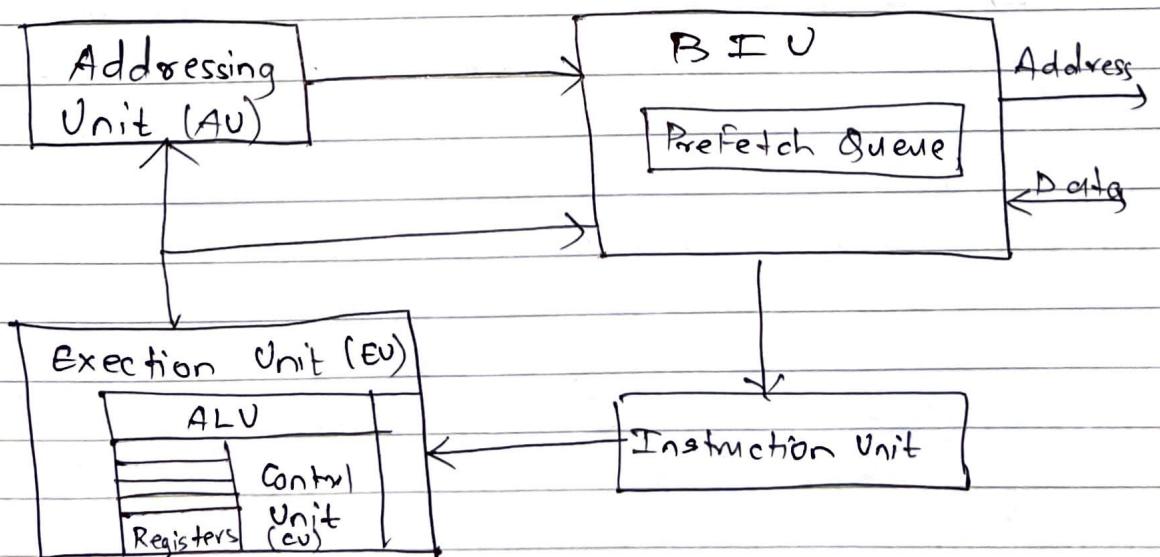
⇒ This unit is responsible for protected mode of 80386 which supports multi-tasking.

7) Segmentation Unit

⇒ This unit is responsible for segmentation mechanism. It is also an important feature that supports multi-tasking in protected mode.

8) Paging Unit

⇒ This unit converts the linear address to physical address.



Intel 80386 DX Processor

CLASSMATE

Date _____

Page _____

Q.1 Explain VM, RF, IOPL, NT and TF Flags of 80386 up.

1) Virtual 8086 Mode (VM)

When the processor enters in virtual 8086 mode, which is an emulation of the programming environment of the 8086 microprocessor, this bit is set to '1'.

- When the processor is in protected mode and this bit is set, the processor moves to virtual 8086 mode.

- In this mode processor handles the segment loads as in 8086.

2) Resume Flag (RF)

When resume flag RF is set to 1, this ignores the debug exception on execution of the next instruction.

- It is automatically reset at the successful completion of every instruction.

3) I/O Privilege Level (IOPL)

→ The IOPL encoded values indicates the privilege level at which the task should be executed to access the I/O devices.

- Privilege levels are used in protected mode to maintain multiple tasks being executed at different privileges and hence can access things accordingly.

4) Nested Task (NT)

⇒ IF $NT=1$, it indicates that the currently executing task is nested within another task and it has a valid link to caller task i.e. this task is executed using the call instruction.

5) Trap Flag (TF)

⇒ When $TF=1$, the processor is put into single-step mode used for debugging.
In this mode, processor generates single stepping interrupt after each instruction, which allows a program to be inspected as it executes.

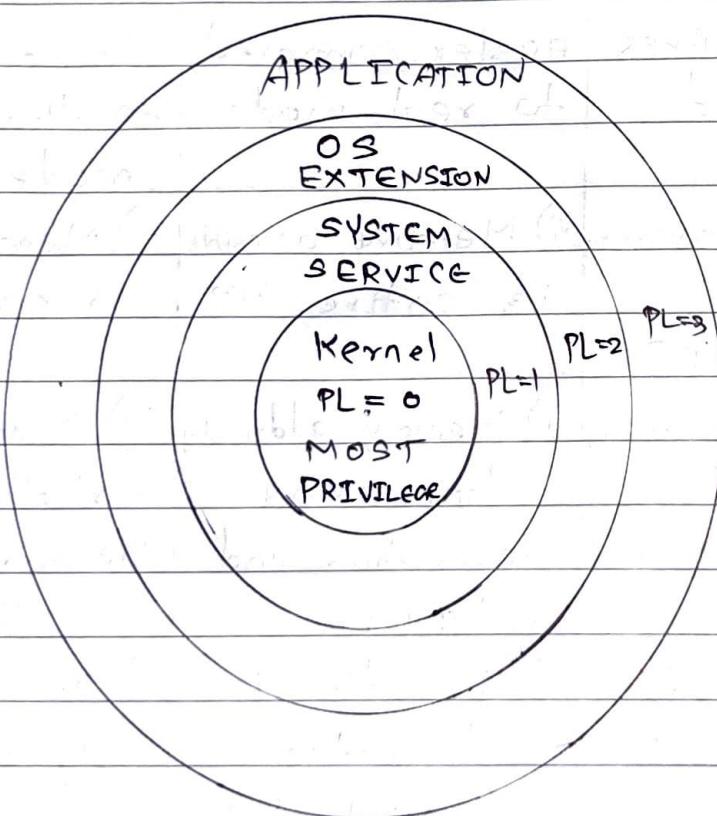
Q.2 Differential between real, protected and virtual mode.

Real Mode	Protected Mode	Virtual Mode
i) Only one task can be executed at any given instant.	i) Multiple tasks can be executed simultaneously.	i) Only one task can be executed at any given instant.
ii) Switching between real and protected mode requires complicated process.	ii) Protected mode switching with virtual mode is easier compared to real mode.	ii) Switching between virtual and protected mode is easy compared to that of real mode.
iii) Maximum memory accessible is $1MB + 16KB - 16B$	iii) Memory accessible is entire 4GB	iii) Memory accessible is entire 4GB
iv) Memory addressing is similar to that of 8086.	iv) Memory addressing is done using descriptors and selectors	iv) Memory addressing virtually seems to be similar to that of 8086
v) No protection amongst task	v) Protection is implemented amongst task	v) No protection amongst task

Q.3

Explain the protection mechanism of 80386 DX with a diagram.

- i) 80386 DX has four levels of protection mechanism which isolates and protects user programs from each other and operating system.
- ii) It offers an additional type of protection on a page basis, when paging is enabled.
- iii) The four-level hierarchical privilege system is illustrated as -



- iv) The privilege levels control the privilege instruction, I/O instructions and access to segments and segment descriptor.

Requested PL \Rightarrow RPL

Descriptor PL \Rightarrow DPL

Current PL \Rightarrow CPL

Effective PL \Rightarrow EPL

$$EPL \Rightarrow \max(RPL, CPL)$$

- v) The PLs are numbered 0, 1, 2 and 3. Level 0 is the most privileged level. Level 3 is the least privileged level.
- vi) As shown in the figure, level 3 is used for user application, level 2 is used for OS extensions, level 1 for system services, and the most privileged level 0 is used for kernel.
- vii) Intel 80386 DX controls both data and procedures according to the following rules:
- Data stored in segment with $PL = p$ can be accessed only by code executed at $PL = p$, numerically, at least as privileged as p .
 - A code segment with $PL = p$ can be called only by a task executing at numerically the same or lower PL than p .
 - A stack segment with $PL = p$ can be used only by a task executing at same PL .
- viii) There are three different types of privilege level entering into the privilege level checks -
- 1) CPL
 - 2) DPL
 - 3) RPL

1) Current PL (CPL)

→ CPL is stored in the selector of currently executing CS register.

- It represents PL of currently executing task.

2) Descriptor PL (DPL)

→ It is the PL of the object which is being attempted to be accessed by the current task.

- It is the PL of target segment and is contained in the descriptor of the segment.

3) Requested PL (RPL)

→ It can be used to weaken the CPL if desired.

- The effective privilege level (EPL) is = $EPL = \max(CPL, RPL)$.

- Thus the task becomes less privileged.

Q.4 Explain descriptor.

- i) Descriptors are the data structures that describes the segment of memory and are used by the processor to provide memory protection and control access to memory.
- ii) The descriptor gives the different details of segment as shown below-

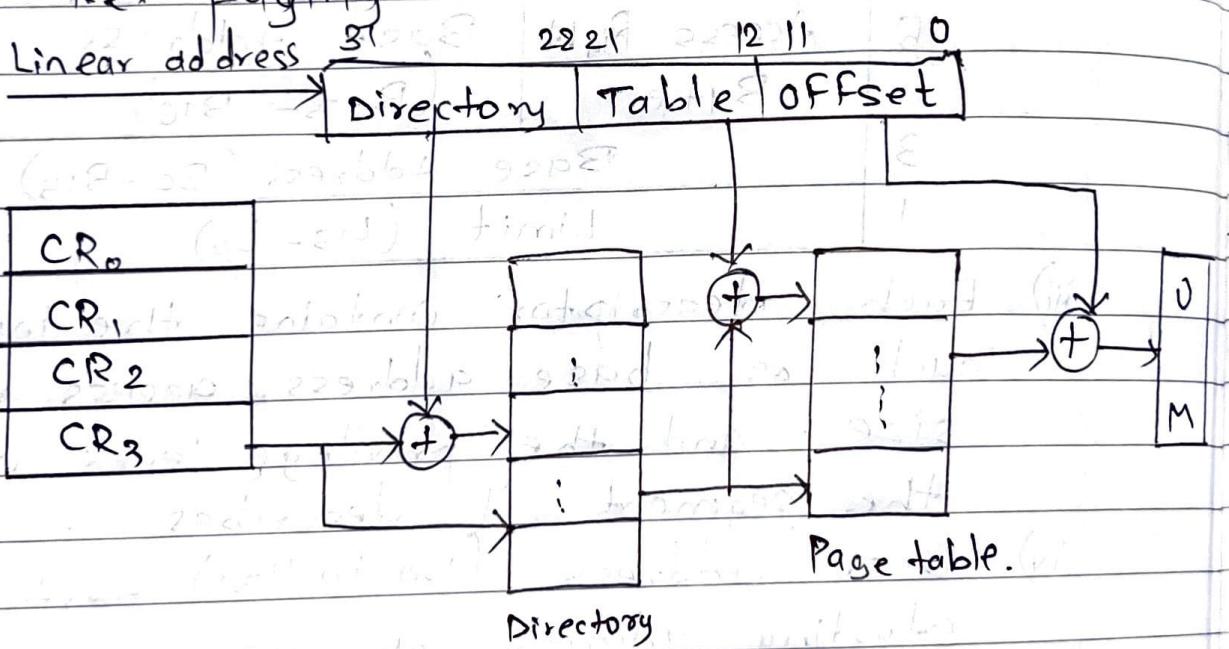
Byte No	Base (B ₃₁ -B ₂₄)	G	D	O	AV	Limit (L ₁₅ -L ₀)	Byte No
7	Base (B ₃₁ -B ₂₄)					Limit (L ₁₅ -L ₀)	6
5	Access Right Byte	Access Right	Base Address (B ₂₃ -B ₁₆)	Base Address	(B ₂₃ -B ₁₆)		4
3			Base Address (B ₀ -B ₁₅)				2
1			Limit (L ₁₅ -L ₀)				0

- iii) Each descriptor contains the information such as base address, access rights, size, and the privilege levels of the segment it describes.
- iv) Base address (B₀ to B₃₁) part indicates starting address of memory segment.
- v) Limit address (L₀ to L₁₅) when added with base address gives the last address of segment.
- vi) AV indicates the availability of segment. When AV=0, it indicates the corresponding segment is used by some other task and hence it is not available. When AV=1, it indicates the corresponding segment is not used by any other task and hence is available.

vii) D indicates the data size, when $D=0$, it indicates 16-bit OS instruction and when $D=1$, it indicates 32-bit OS instructions.

Q.5 Explain paging in protected mode of 80386

\Rightarrow i) 80886 processor use the virtual memory system with the help of on chip Memory Management Unit (MMU) i.e. paging.



ii) If paging unit is enabled, then paging unit converts linear address into physical address.

iii) Page directory Base Register

\Rightarrow The CR3 is used as page directory base register to store starting address of page directory.

iv) Page directory

- ⇒ This is the at most 4KB in size.
- Each directory entry is of 4B, thus total of 1024 entries are allowed in a directory.
 - The page directory entries points to the page table.

v) Page Table

- ⇒ Each page table is of 4KB in size and maximum contains 1024 entries.
- The address bits $A_{12} - A_{21}$ are used to select 1024 page table entries.
 - The page table entry contains the starting address of page.
 - The page table can be shared between the tasks.