UNIT- V

MULTITASKING AND VIRTUAL 8086 MODE

Page No.
Date

① Draw and explain task state segment of 80386.

→

- Task state segment is a special type of segment, used to manage the task. The 80386 uses TSS like a scratch-pad. It stores everything it needs to known about task in TSS. This means the task environment (context) is stored in TSS.
- TSS is not accessible to general user program or program even at privilge level 0. The fields within TSS are accessible to only 80386.

- The fields of TSS are divided into two sets:

(i) Dynamic set:

- The 80386 updates dynamic set when it switches from one task to another task. This set includes:-
  - General Registers (EAX, EBX, ECX, EDX, ESP, EBP, ESI, EDI).
  - The segment Registers (CS, SS, DS, ES, FS, GS)
  - The flag registers (EFLAGS)
  - The Instruction pointer (EIP)
  - Back Link.

- the first 4 fields save state of microprocessor, 80386. Saving EIP gaurantees that task will restart at point at which it was stop. & saving EFLAGS allows 80386 to execute conditional instructions properly, when the task is restarted.

- The back link is used to keep track of a previous task.

31		0	
Bit map offset	0000000000000000	T	64
0000000000000000	LDT		60
0000000000000000	GS		5C
0000000000000000	FS		58
0000000000000000	DS		54
0000000000000000	SS		50
0000000000000000	CS		4C
0000000000000000	ES		48
0000000000000000	EDI		44
ESI			40
EBP			3C
ESP			38
EBX			34
EDX			30
ECX			2C
EAX			28
EFLAGS			24
EIP			20
CR3			1C
0000000000000000	SS2		18
EIP2			14
0000000000000000	SS1		10
EIP1			0C
0000000000000000	SS0		8
EIP0			4
0000000000000000	Back link		0

fig. Task State segment

(ii) Static set :

The 80386 only reads field from this set. This set includes:

- The selector for task's LDT
- The register (PDBR) that contains base address of task's page directory.
- Points to stacks for priviuge level 0-2.
- The T- bit (debug exception on Task switch)
- I/O map offset

OR

2) Explain TSS descriptor of 80386 with a neat diagram.

→ - The task state segment is defined by descriptor called TSS descriptor. It contains field like other segments.

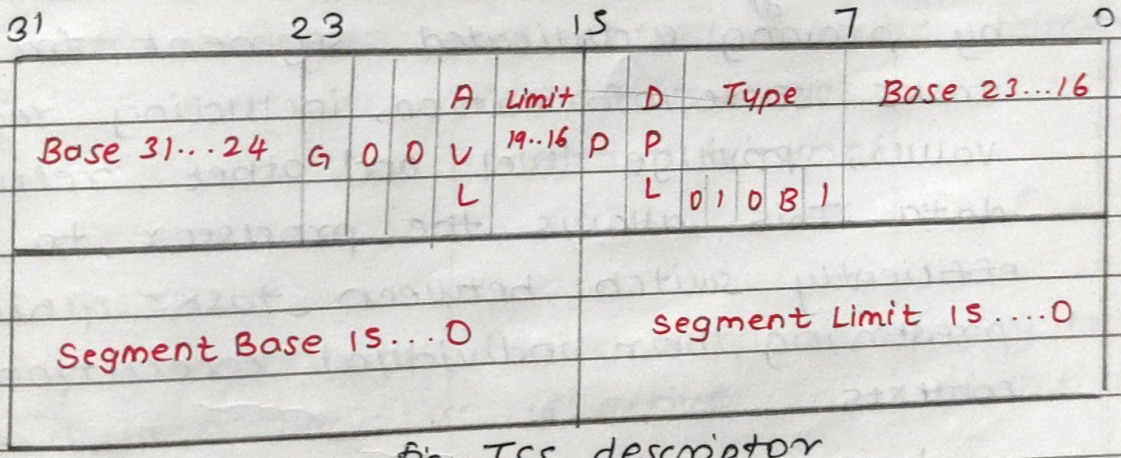


fig. Tss descriptor

- The B- bit in the type field indicates whether the task is busy & also allows to detect & attempt to switch to a task that is already busy.
- The BASE, LIMIT and DPL fields and the G bit and P-bit have functions similar to other descriptors.

- The limit field however must have a value equal to greater than 103, because 80386 requires a minimum 104 bytes of storage in order to perform a context save. A Larger limit is permissible and it is required if an I/O permission map is present. The maximum limit for TSS is 4 GB.

- To access TSS Descriptor, the procedure must have privilge value less or equal to privilge level specified by DPL in descriptor. Usually this access is restricted for only trusted softwares, whose privilge level is 0.

- The TSS descriptor facilitates multitasking by proving a dedicated segment for each task's state information, including register values, privilge level and other relevant data. This allows the processor to efficiently switch between tasks while preserving their individual execution contexts.

Page No.
Date

3) Different between all operating modes.

→

	Real mode	Protected mode	Virtual mode
(i)	It is default mode of processor when it is switched ON.	To support multi-tasking 80386 has as special mode	switching b/t real & protected is quite complication. It allows 8086 tasks to executed without restarting.
(ii)	It is used to execute 8086 tasks	It is used to execute multi-tasking based task of 80386	It is used to execute 8086 task without restarting processor
(iii)	Memory size is limited to 1MB	It increases linear address space to 64TB and allows virtual programs to run of 64TB	memory size is limited to 1MB
(iv)	Paging mechanism is in-active	Paging mechanism is active	Paging mechanism is active
(v)	Protection mechanism is not available	protection mechanism is available	Provides mechanism to selectively trap & manage I/O interrupts
(vi)	multitasking is not supported	multitasking is supported	Multitasking is supported
(vii)	only $A_2 - A_{19}$ address lins are active	All address lines are active	All address lines are active.

4] List and explain features of V86 mode.

→

(i) Segmentation:
V86 mode retains the segmented memory of 8086 processor, allowing each virtual machine to access up to 1MB of memory.

(ii) Protected Environment:
Despite running in V86 mode, each virtual machine operates within the protected mode environment of the processor, providing memory protection and multitasking capabilities.

(iii) Interrupt Handling:-
V86 supports interrupt handling, allowing each virtual machine to respond to hardware interrupts and software interrupts (such as system calls).

(iv) Task switching:
Like real mode, V86 mode supports task switching, enable rapid context switches between multiple virtual machines.

(v) Compatibility:
V86 mode ensures compatibility with legacy 8086 software, allowing older applications designed for real mode to run unchanged within a protected mode enviroment.

(vii) Access to protected mode features:
V86 mode allows virtual machines to access protected mode features such as paging, which enables efficient memory management and protection.

(viii) Performance :-
While not as efficient as native protected mode, V86 mode offers improved performance over full virtualization techniques by leveraging the processor's native execution capabilities.

5) With the necessary diagram, explain entering & leaving the virtual mode of 80386.

→

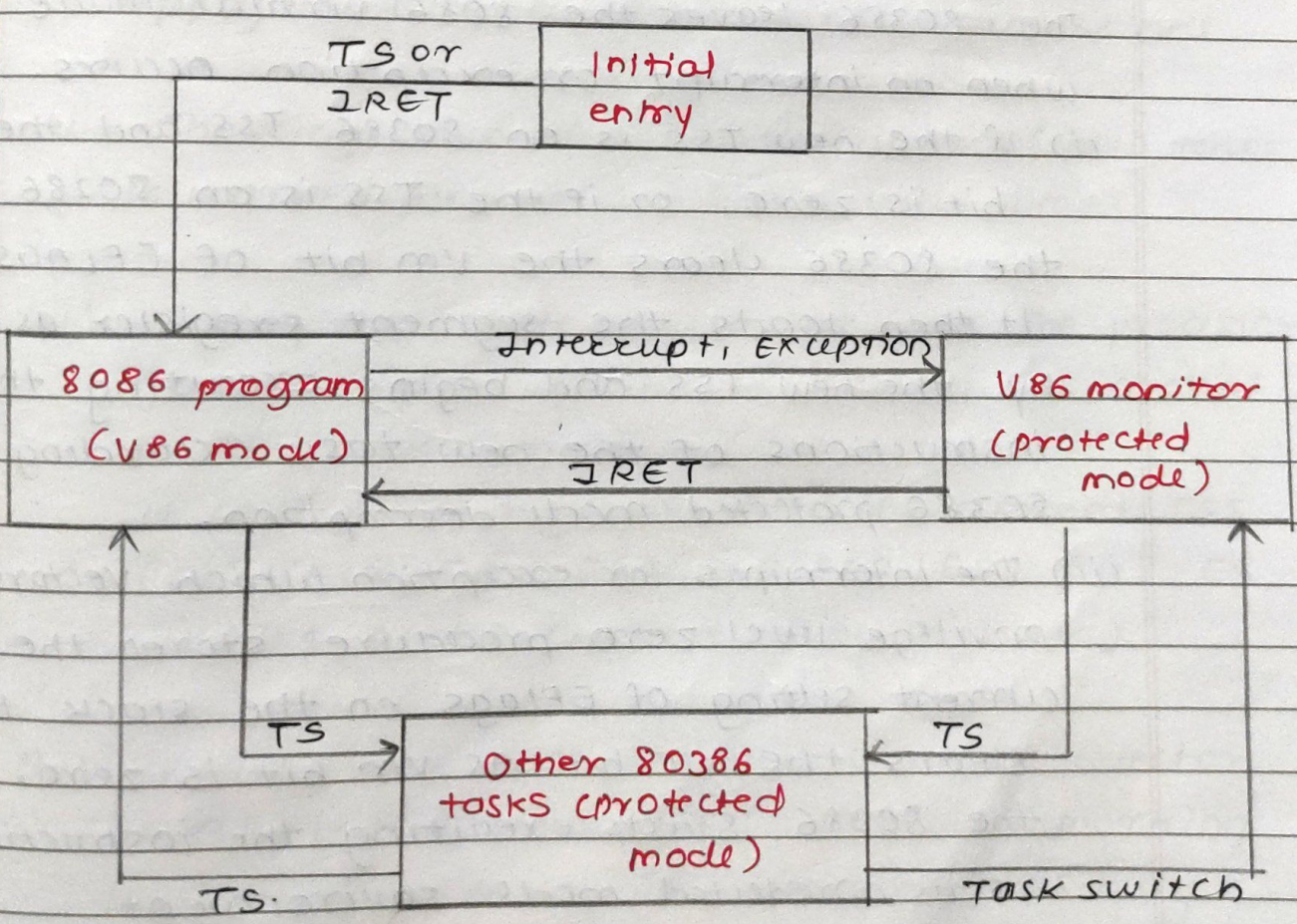


fig. Entering & leaving an 8086 program

* Entering 8086 virtual mode :-

The 80386 can enter 8086 virtual mode by either of two means:

(i) If the VM bit in EFLAGS register is set, the 80386 enters virtual 8086 mode to execute the new task. If the VM bit is not set, the 80386 executes the new task as a normal protected mode task.

(ii) An IRET from a procedure that loads the EFLAG's image changes the VM bit if CPL at time of IRET is zero. If changed status of the VM bit is 1 then 80386 enters in 8086 virtual mode.

* Leaving 8086 virtual mode :-

The 80386 leaves the 8086 virtual mode when an interrupt or exception occurs.

(i) If the new TSS is an 80386 TSS and the VM bit is zero, or if the TSS is an 80286 TSS, the 80386 clears the VM bit of EFLAGS It then loads the segment sregister as defined by the new TSS and begins executing the instructions of the new task according to 80386 protected mode description.

(ii) The interrupts or exception which vectors to a privilge level zero procedure, stores the current setting of EFlags on the stack, then clears the VM bit. As VM bit is zero, the 80386 starts executing the instructions in its protected mode environment.

Page No.
Date

(6) Define task switching and explain step involved in task switching operation.

→

- The task switching is the process of changing the currently executing task on a CPU. In context of 80386DX microprocessor, which supports multitasking and multitasking OS, task switching involves saving the state of the currently running task and loading the state of new task to be executed.

- A task switching operation involves the following steps:-

(i) checking that current task is allowed to switch to designated task. Data access priviliges apply in case of Jmp or CALL instructions.

(ii) Checking that the TSS descriptor of new tasks is marked present and has a valid limit.

(iii) saving the state of current task. The processor finds the base address of current TSS cached in the task register.
It copies the registers into the current TSS (EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI, ES, CS, SS, DS, FS, GS and flag register).

(iv) Loading the task register with the selector of the called task's TSS descriptor, marking the incoming task's TSS descriptor as busy and setting the TS bit of CR0.

(v) Loading the called task's state from its TSS and resuming execution.
The registers loaded are LDT registers, the flag register and the general registers, segment register and PDBR.

- The state of caller task is always saved when a task switch occurs.

- If the execution of that task is resumed, it starts after the instruction that caused the task switch. The registers are restored to the values they held when the task stopped executing.

- Every Task switch sets 'TS bit in CR0. The TS flag is useful to systems software when a coprocessor is present.

Page No.
Date

* Explore memory management in the virtual 8086 mode.

⇒

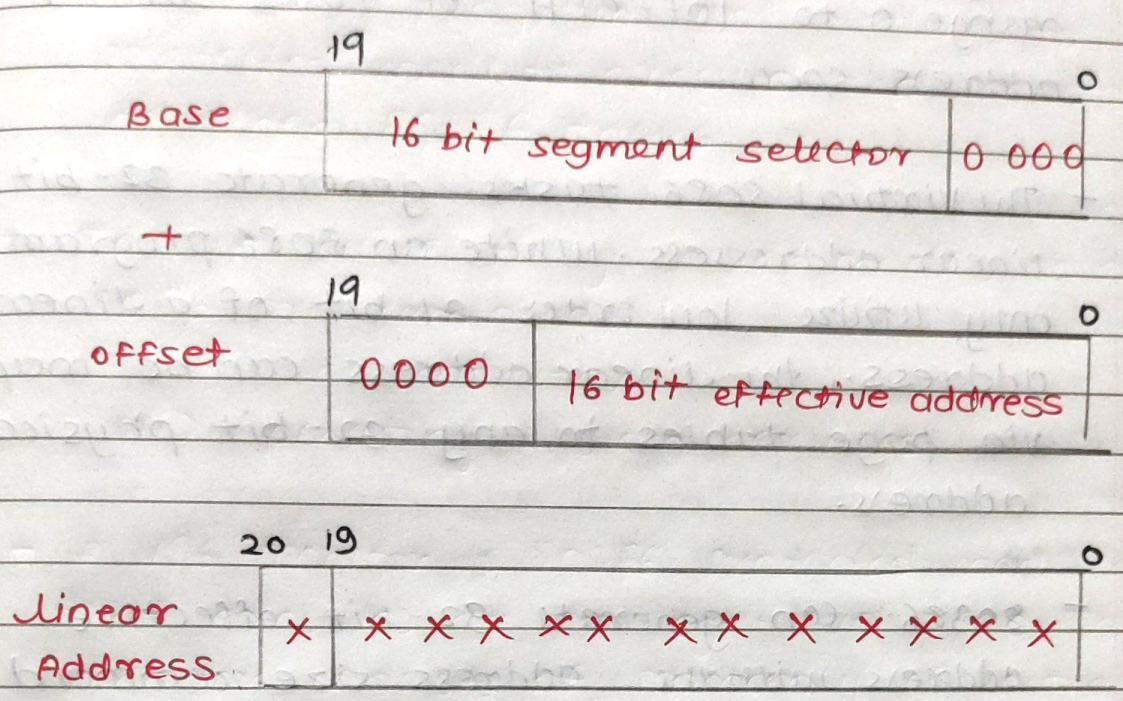


fig. Virtual 8086 mode address generation

- In V86 mode, content of segment registers are not used as a selector to point the descriptor. But the segment register contents are used to generate linear address with help of offset.

- The linear address is generated by adding the contents of appropriate segment register which are shifted left by 4 bit to an effective address / offset. {multiply by 16}

- If there is a carry generated after addition of shifted segment registers contents and effective address, unlike 8086, resulting 21 bit address is a linear address.

- An 80386 in virtual 8086 mode is allowed to generate linear address anywhere in range 0 to 10FFEFH of tasks linear address space.

- The virtual 8086 tasks generate 32-bit linear addressess. While an 8086 program can only utilize low order 21-bit of a linear address, the linear address can be mapped via page tables to any 32-bit physical address.

- 80386 can generate 32-bit effective address with the address size command prefix. This address should not exceed beyound 65535 to maintain compatibility with 80286 real mode; otherwise 80386 generates pseudo-protection fault (INT 12 OR INT 13 with no error code)

Page No.
Date

* Explain the structure of V86 task in detail.

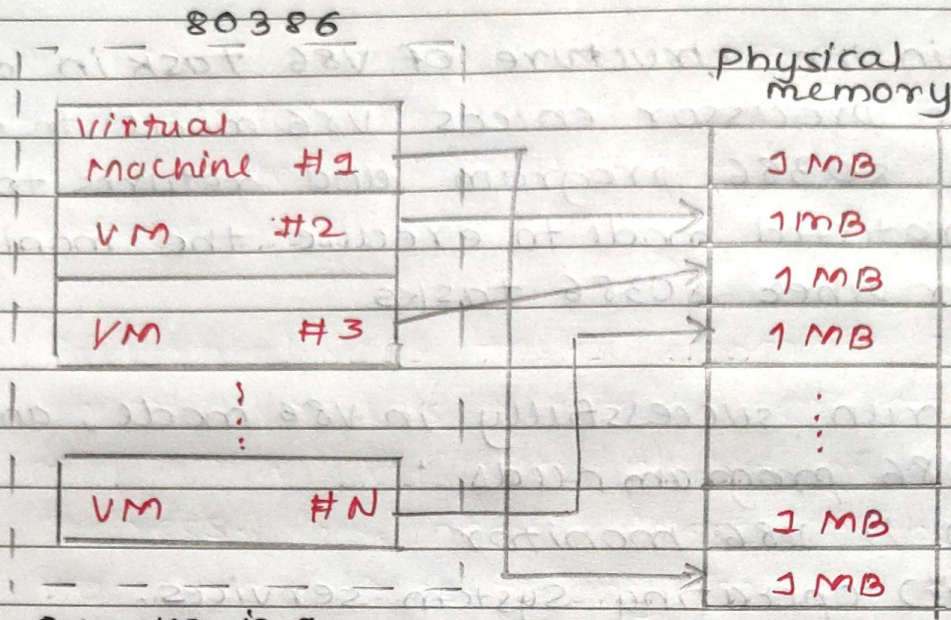
→ - The processor enters V86 mode to execute the 80386 program and returns to protected mode to execute the monitor or other 80386 tasks

- To run successfully in V86 mode, an existing 8086 program needs:
  (i) A V86 monitor
  (ii) Operating system services.

- A V86 monitor is 80386 protected mode code that executes at privilege-level zero. The monitor consists of primarily of initialization and exception handling procedures.

- In 80386 program, executable segment descriptor for the monitor must exists in GDT or task's LDT. The linear address above 10FFEFH are available for V86 monitor, OS and other softwares.

- The monitor may also need data-segment descriptors so that it can examine the interrupt vector table or other parts of 8086 program in the first Mgabyte of address space.



Each VM is a separate 8086 system

Paging mechanism allows 1MB space to be anywhere in 4GB physical memory

- In general, there are two options for implementing the 8086 operating system:

① The 8086 OS may run as part of the 8086 code. This approach is desirable for any of following reasons:
- 8086 applications code modifies the Operating system.

② The 8086 operating system may be implemented or emulated in the V86 monitor.
- Operating system functions can be more easily coordinate among several V86 task.

7.3.4 Protection within a V86 Task

- In V86 mode, protection mechanisms offered by descriptors is not available. Thus in V86 mode, to protect the systems software designers may follow either of these approaches :
  - Reserve the first megabyte (plus 64 kilobytes) of each task's linear address space for the 8086 program. An 8086 task cannot generate addresses outside this range.
  - Use the U/S bit of page-table entries to protect the virtual-machine monitor and other systems software in each virtual 8086 task's space. When the processor is in V86 mode, CPL is 3. Therefore, an 8086 program has only user privileges. If the pages of the virtual-machine monitor have supervisor privilege, they cannot be accessed by the 8086 program.

Do not write your name or seat no. below this line

Q. No.						Q. No.					

प्र. क्र.
Q. No.

(4) Control transfer:-

- Finally the Processor transfers control to ~~other~~ offset specified in call gate.

2] Explain role of TASK register in multitasking and instruction used to modify and read TR.

→

- Multitasking is ability of Computer to run more than one program or task at some time.

-

*Role of TASK Register in multitas king :-

- The TASK Register (TR) identify curr enty executing task by pointing to TSS.
- The TASK register has both visible portion which can read and change by instruction and invisible portion which cannot read by an instruction.
- The selector in visible portion is used to ~~select~~ specify TSS descriptor in GDT.

-while the invisible portion is used as a cache which hold base and limit value from TSS descriptor.
-Holding the base and limit in invisible portion of TOSK Register make execution of tosk faster, because processor does not need to repeadetely fetch this value from memory.

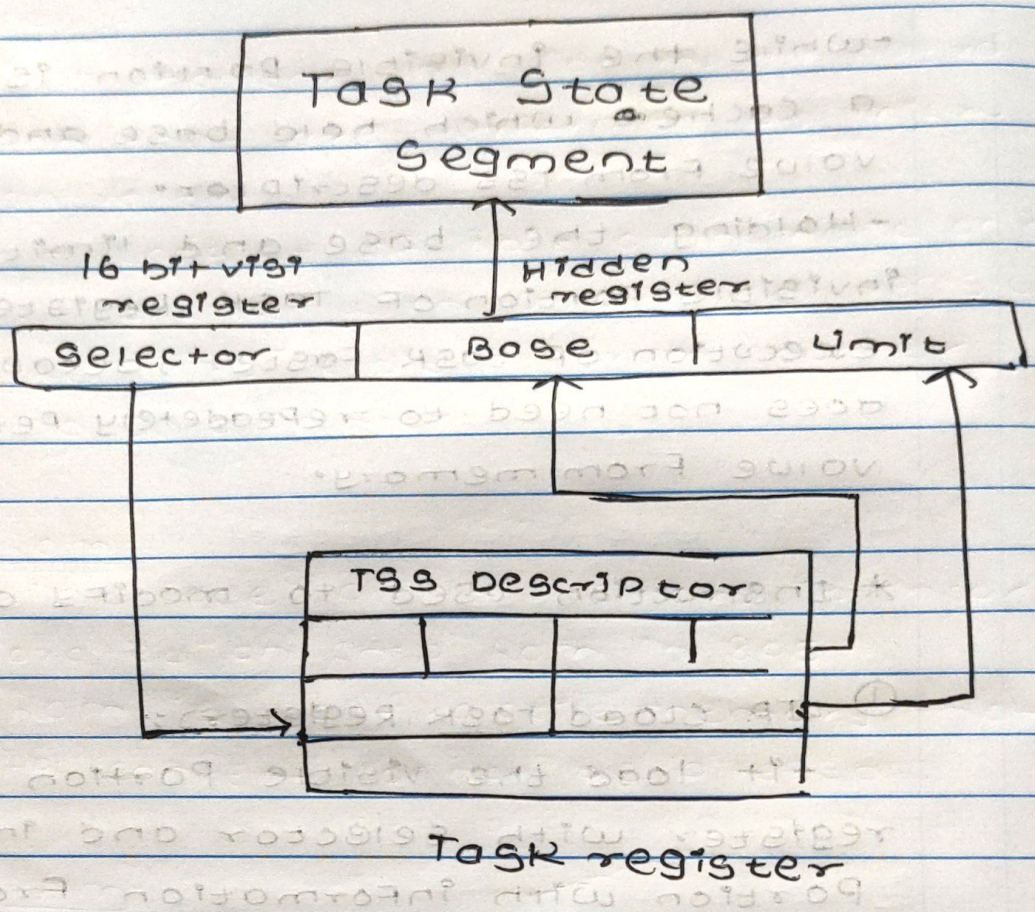
* Instruction used to modify and read

① LTR (Load TOSK Register):-
-It load the visible portion of tosk register with selector and invisible portion with information from TSS descriptor selected by selector. LTR is Privilege instruction.

② STR (Store TOSK Register):-
- It store the visible portion of tosk register in general register or memory word.
-STR is not privilege instruction.

Q. No.						Q. No.					

प्र. क्र.
Q. No.



Q3] a] With help of neat diagram explain the architecture of typical microcontroller.

→