ASSIGNMENT NO. 6

AIM :- Write ALP to switch from real mode to protected mode and display the values of GDTR, LDTR, IDTR, TR and MSW Registers.

APPARATUS :

- Core 2 duo/i3/i5/i7 - 64bit processor
- OS – ubuntu 32bit/64bit OS
- Assembler used –nasm (the netwide assembler)
- Editor Used – gedit

THEORY :

Protected Mode Definition:

The 80286, 80386, 80486 and Pentium microprocessors are capable of operating in two basic modes of operation:

- Real mode
- Protected mode.

In protected mode, the full power of the processor is available. Registers are 32 bits wide. DOS is an example of a program that runs in real mode, because DOS was created for an 8086 based machine. An example of a program that runs in protected mode is Windows TM /Linux.

Protected Mode Application:

The program presented here takes advantage of the capabilities to execute some privileged instructions such as SGDT and STR, and display the results of the execution. A number of instructions are included in this program.

The SMSW instruction saves a copy of the lower 16 bit of CR 0 . These bits are known as the Machine Status Word and can be loaded with new data using LMSW (Load Machine Status Word) instruction.

Protected mode Registers

1. Global Descriptor Table Register (GDTR):

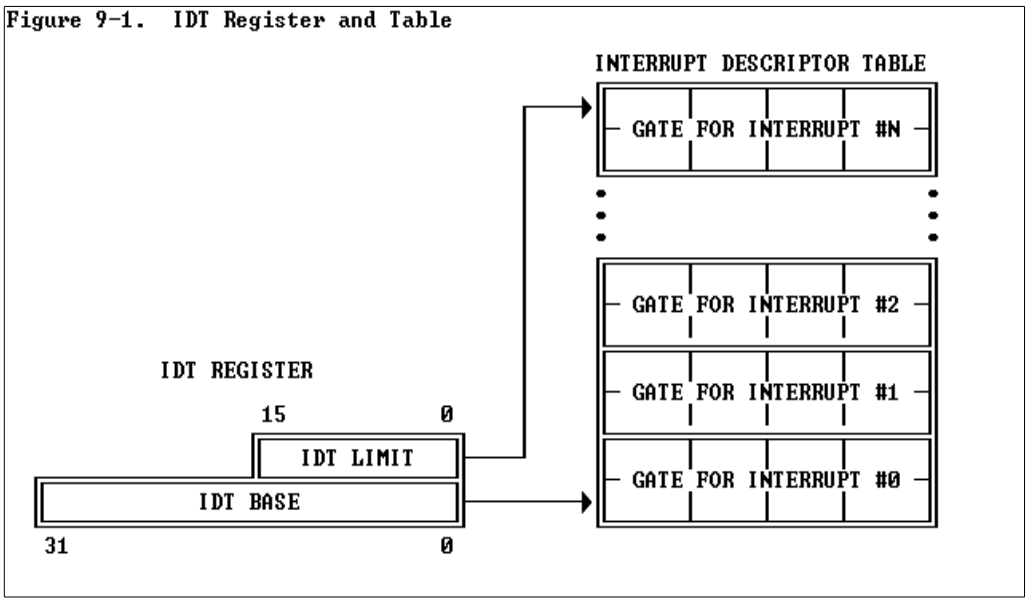
- GDTR is a 48 – bit register of the Pentium processor.
- The lower 2 bytes [bit 0 to bit 15] are called as LIMIT.
- The upper 4 bytes of GDTR is called as BASE. The BASE gives beginning physical address of GDT in memory.
- This 32 –bit base address allows the GDT to be place anywhere in the Pentiums address space.

Eg.

If BASE = 00 10 00 00 H

then

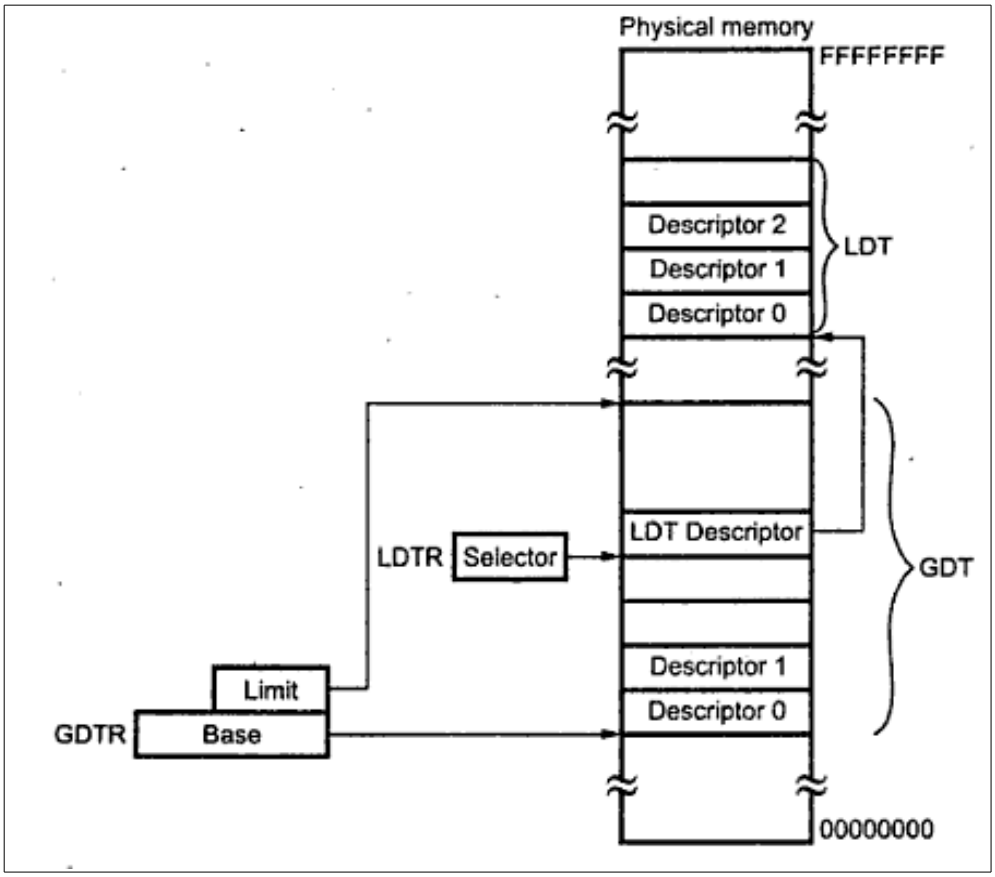
GDT starts at 00 10 00 00 H in physical memory Space



2. Interrupt Descriptor Table Register [IDTR]:

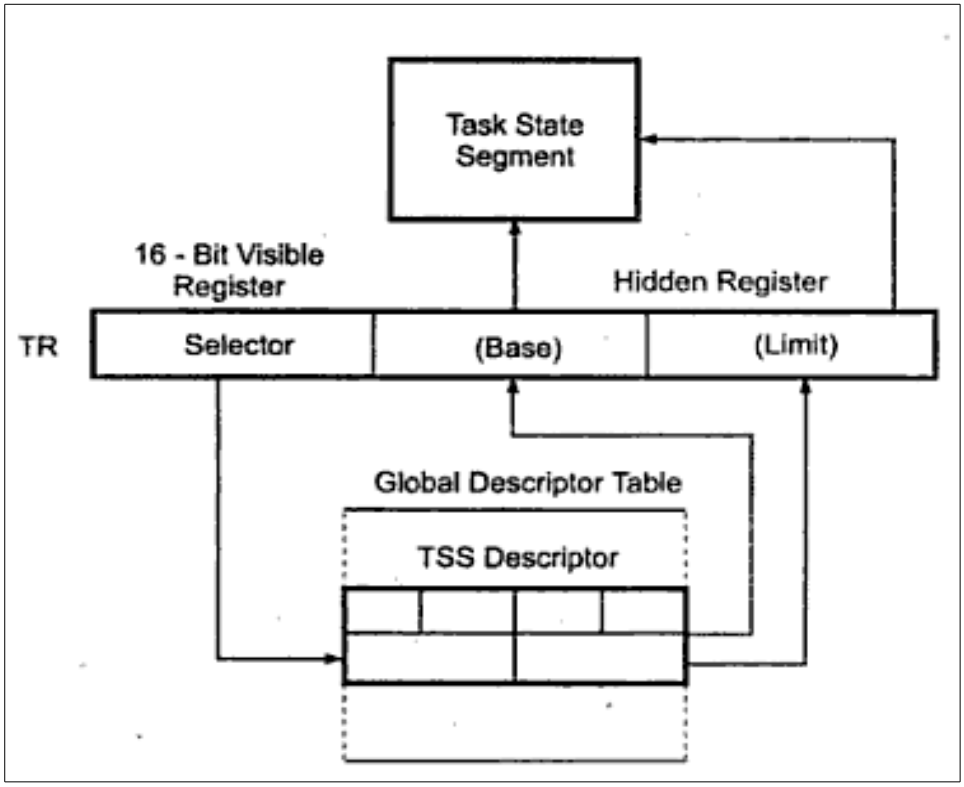
- IDTR is a 48 bit register of the Pentium processor
- The lower 2 bytes of the register [LIMIT] gives the size of the IDT and upper four bytes [BASE] identifies the starting address of the IDT in physical memory.

- The size of the IDT is equal to LIMIT +1 and IDT can be up to 65,536 bytes long.
- Similar to GDTR, the IDTR has to be loaded before switching the Pentium from the real mode operation to the protected mode operation.



3. Local Descriptor Table Register (LDTR):

- LDTR is a 16-bit register of the Pentium processor
- The LDTR does not directly select the LDT, rather it gives a selector which points to an LDT descriptor in the GDTR.
- The 32-bit base value in local descriptor table cache of Pentium gives the starting point of the LDT table in physical memory and the value of the 16-bit limit in it gives the size of the LDT.



4. Task Register

- Whenever a task Switch occurs, Pentium automatically saves the complete context dof the old task in a TSS and loads the context of a new task specified in another TSS.
- The Task register consist of two parts a visible part and an invisible part. The visible part is accessible to the user. The invisible part is automatically loaded with the information associated with the TSS descriptor.
- It is a 16- bit selector for (TSS) Task State Segment Descriptor.

ALGORITHM :-

1. Start
2. Declare & initialize the variables in .data section.
3. Declare uninitialized variables in .bss section.

4. Declare Macros for print and exit operation.
5. Store MSW.
6. Rotate right MSW by 1 bit, so last bit will be available at carry flag.
7. If carry is set then, print the message “Processor is in protected mode” else print the message “Processor is in Real mode”.
8. Save the contents of GDTR, IDTR, LDTR and TR.
9. Print the contents of GDTR (Global Descriptor Table Register).
10. Print the contents of IDTR (Interrupt Descriptor Table Register).
11. Print the contents of LDTR (Local Descriptor Table Register).
12. Print the contents of TR (Task Register).
13. Print the contents of MSW (Machine Status Word) (16 bits of CR0 [0-15]).
14. Using Macro terminate the process.
15. Define display procedure.
16. Stop

CONCLUSION: