80486 Register Organisation

General Purpose

31 16	15 0	
	AX	EAX
	BX	EBX
	CX	ECX
	DX	EDX
	SI	ESI
	DI	EDI
	BP	EBP
	SP	ESP

SEGMENT

CS	code segment
SS	stack segment
DS	data segment
ES	extra segment
FS	
GS	

Instruction Pointer and Flag

31 16	15 0	
	IP	EIP
	FLAGS	EFLAGS

- The 80486 has eight 32-bit general purpose register which may be used as either 8 bit or 16 bit registers
- A 32-bit register known as extended register is represented by the register name with prefix E.
- The six segment registers available are CS, SS, DS, ES, FS and GS
- The CS and SS are the code and the stack segment respectively while DS, ES, FS, GS are 4 data segment registers

EEFLAG Register of the 80486

31	18																	
RESERVED FOR INTEL	AC	VM	RF	0	NT	IOPL	OF	DF	IF	TF	SF	ZF	0	AF	0	PF	1	CF

CF : Carry Flag

AF : Auxillary Carry

ZF : Zero Flag

SF : Sign Flag

TF : Trap Flag

IE : Interrupt Enable

AC : Alignment Check

DF : Direct Flag

OF : Over flow

IOPL : I/O Privilege level

NT : Nested Task Flag

RF : Resume Flag

VM : Virtual Mode

- Carry flag is set to 1 when there is arithmetic carry or borrow has been generated
- Auxiliary flag is set when carry from lower nibble of the flag register is transfered to upper nibble.
- zero flag is set when the result generated is zero
- sign flag is set when the result is negative
- Trap flag permits operation of a processor in single step mode. If flag is available, debuggers can use it to step through the execution of a computer program
- Interrupt flag decides if the microprocessor will handle the maskable hardware interrupt.
- Alignment check is set if a word or doubleword is addressed on a nonword or non doubleword boundary.
- Direction flag determines whether string processing instructions increment or decrement the 16 bit half - registers SI and DI or the 32 bit registers ESI and EDI

Scanned by CamScanner

- Overflow flag indicates an arithmetic overflow after an addition or subtraction.
- IOPL when set, causes the processor to generate an exception on all accesses to I/O devices during protected mode operation.
- NT indicates that the current task is nested within another task in protected mode operation.
- RF allows the programmer to disable debug exceptions so that the instruction can be restarted after a debug exception without immediately causing another debog exception.
- VM allows the programmer to enable or disable virtual 8086 mode, which determines whether the processor runs as an 8086 machine.

Protected Mode

When configured for the protected mode operation, the 80486 provides an advanced software architecture that supports memory management, virtual addressing, paging and multitasking. There are four new registers in the protected mode model

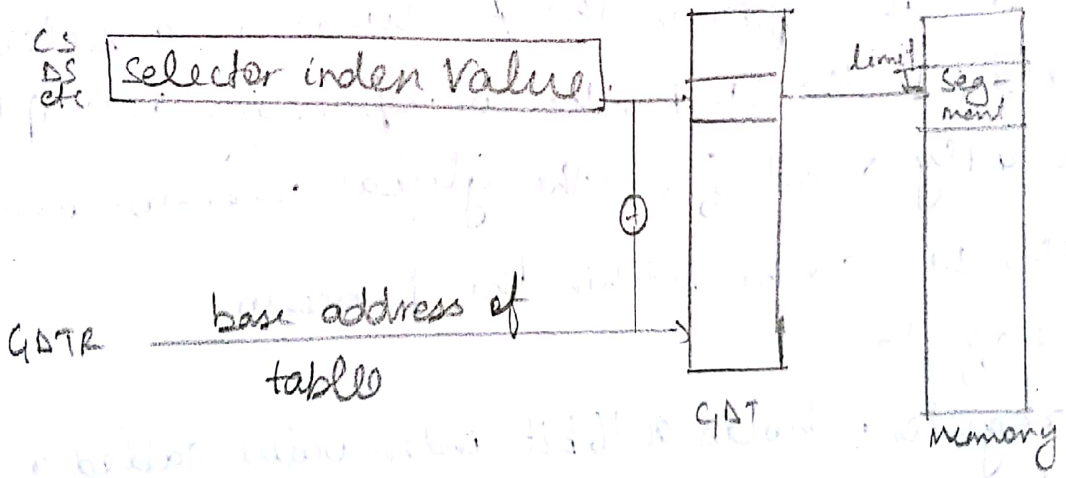
- The Global Descriptor Table Register (GDTR)
- Interrupt Descriptor Table Register (IDTR)
- Local Descriptor table register (LDTR)
- Task Register (LTR)

5ii Global Descriptor Table Register

The contents of the GDTR define a table in the 80486 physical memory address space called Global Descriptor Table, which is one important element in CPU memory management system.

The GDT provides a mechanism for defining the characteristics of the global memory address space. i.e storage locations in the global memory is accessible by any task that runs on the microprocessor.

System segment Descriptors identify the characteristics of the segments of the global memory.



Interrupt Descriptor Table Register

Just like GDTR, the IDTR defines a table in the physical memory. The contents of the table are interrupt descriptors. The register and table provide the mechanism by which the microprocessor passes the program control to the interrupt and exception service routine.

Local Descriptor Table Register

LDTR is also a part of the memory management support mechanism. Each task can have access to its own private descriptor table in addition to the GDT. The private descriptor table is called LDT and defins a local memory address space for use by the task. The LDT holds segment descriptors that provide access to the codes and the data in segments of memory that are reserved for the current task. The contents of LDTR donot directly define the local descriptor table. Instead it holds a selector that points to an LDT descriptor in the GDT. Whenever a selecter is loaded into the LDTR, the corresponding descriptor is apparently read from the global memory and loaded into the LDT cache within the processor.

TASK Register

This register holds a 16 bit index value called a selector. The initial selector must be loaded into TR under software control. This starts the initial task. After this is done, the selector is changed automatically whenever the processor executes an instruction that performs a task switch. When a selector is loaded into the TR, the corresponding Task State Segment descriptor automatically gets read from the memory and loaded into the on-chip Task Descriptor cache.

when a selector is loaded into the TR, the corresponding Task State Segment descriptor automatically gets read from the memory and loaded the on-chip descriptor cache. This descriptor defines a block of memory called the Task state segment. The TSS holds the information needed to initiate a task, such as initial values for the user accessible registers