SUMMER – 2019 EXAMINATION
MODEL ANSWER

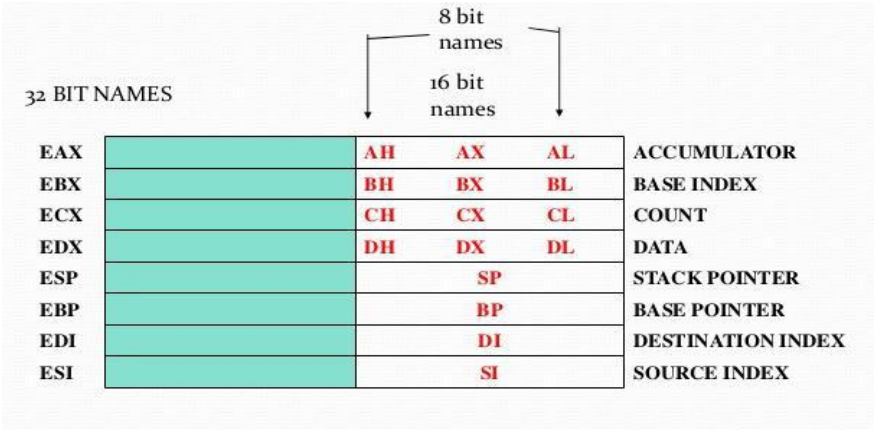
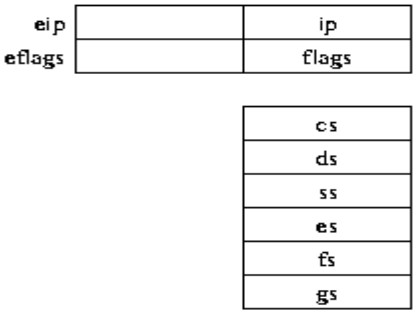
Subject: Advanced Microprocessor **Subject Code: 17627**

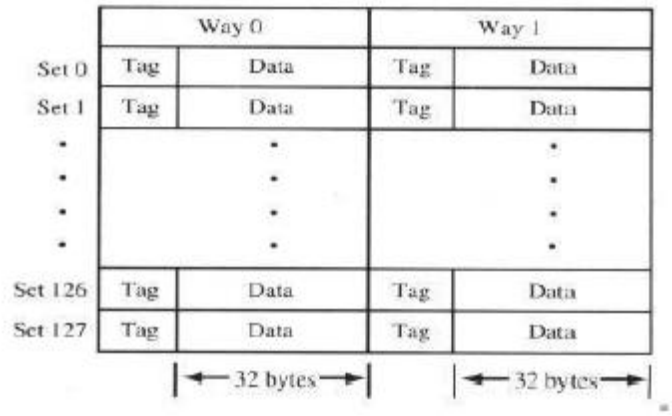
Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q.N.	Answer	Marking Scheme
1.	**(a) Ans.**	**Attempt any FIVE:** **Explain features of 80386 microprocessor. (any eight)** The salient features of 80386 are as follows: 1. It is a 132 PGA (pin grid array) with 32 bits non multiplexed data bus and 32 bits address bus. 2. It works in 3 modes: real, protected and virtual 8086 mode (V-86). 3. It can address total 2^{32} i.e., 4GB physical memory with the help of its 32 bits address lines. 4. The integrated memory management unit in 80386 supports segmentation and paging of memory. 5. It supports the interface of 80387-DX coprocessor IC to perform the complex floating point arithmetic operations. 6. It supports 64TB virtual memory. 7. It has an integrated memory management unit which supports the virtual memory and four levels of protections. 8. It has an on chip clock divider circuitry. 9. It has BIST (built in self-test) feature which tests approximately	**20** **4M** ***Any eight features $^{1/2}$M each***

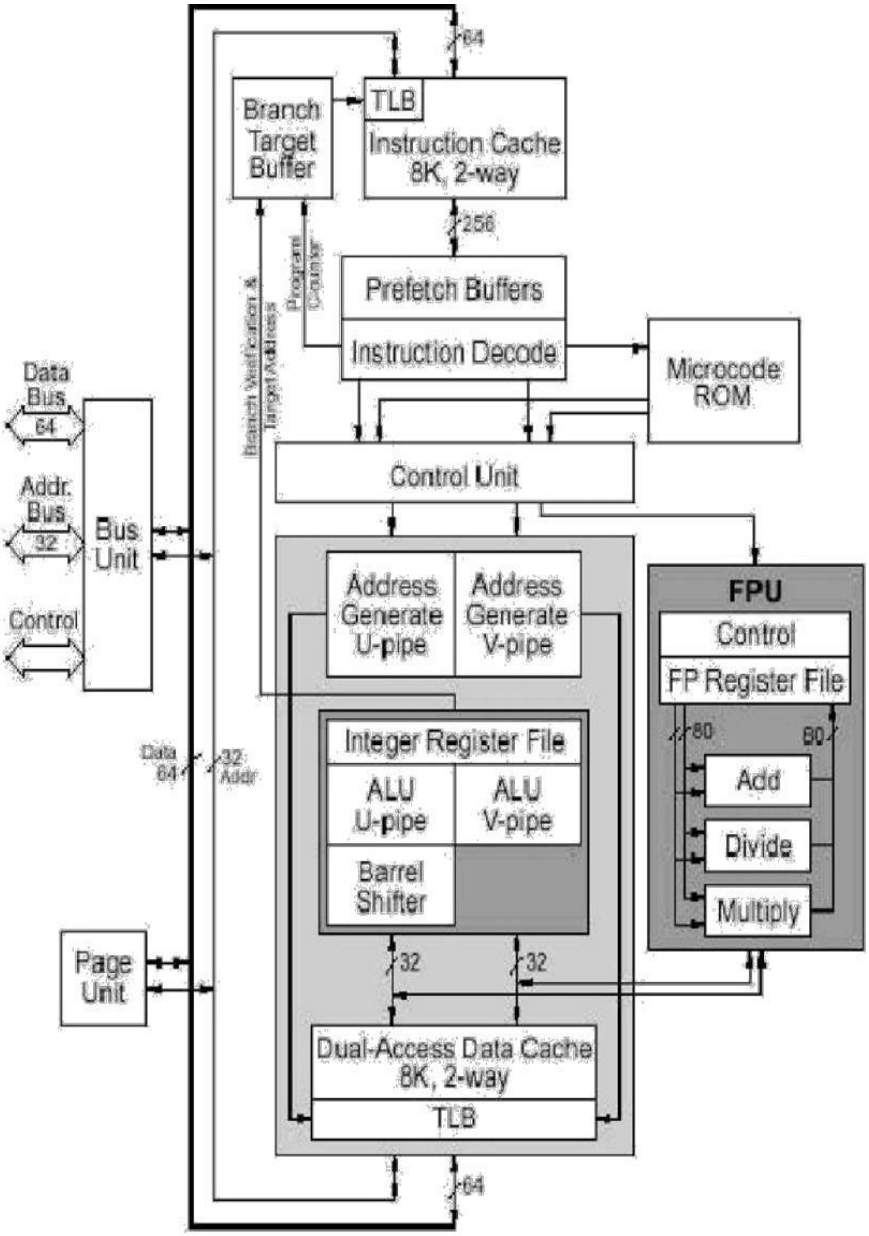
		one half of the 80386 processor when RESET and BUSY are active. 10. It has breakpoint registers to provide the breakpoint traps on code (instructions) execution or data access. 11. It supports instruction pipelining with the help of 16 bytes instruction prefetch queue. 12. It has 8,32 bit General Purpose bits registers to store the data and address at the time of programming. 13. It has 8 debug registers DR0-DR7 for hardware debugging and control. 14. It has a 32 bit E-flag register. 15. It supports the dynamic bus sizing by which the 80386 can be interfaced to 16 bits devices effectively. And also supports the 8bits, 16 bits and 32 bits operands. 16. It operates on 20 MHz and 33 MHz frequency.	
	(b) **Ans.**	**Draw Register organization format of 80386 DX. State the function of segment registers.** **Register organization** (General-purpose register, segment register, status and control register, instruction pointer. The 80386 has 32 bits registers which are classified into following seven types : 1. General purpose registers. 2. segment registers 3. instruction pointer and flags registers 4. Control registers 5. System address registers 6. debug registers 7. Test registers **General purpose registers**	**4M**

		  **Segment registers:** There are total 6 segment registers of 16 bits each in 386. They are corresponding to CS, DS, ES, SS,FS, GS. In real mode size of each segment is 64kB while in protected mode each segment could be of 4GB. The CS and SS are used as code and stack segment respectively while the DS, FS, GS, ES are used as the data segments in 386. The segment registers in protected mode are called as segment selectors. These registers CSR, SSR, DSR, ESR, FSR and GSR are used in the physical address calculation in the different ways in different modes in 80386.	***Diagram of register organization format of 80386 DX 2M*** ***Function of segment registers 2M***
	(c) Ans.	**Justify the use of separate cache in Pentium configuration.** **Separate 8K B instruction and Data Cache :** The following figure shows the organization of instruction and data cache.	**4M**

		 The Pentium processor has 2 separate 8KB data and code Caches. But they need more bandwidth than the unified cache. Both the caches have TLB's associated with them. The TLBs are used to covert the linear addresses to the respective physical addresses. As the data cache stores only 8KB data and code cache stores only instructions, the lookup process speed for Pentium increases. Advantages of separate instruction and data caches : 1. Separate code and data cache memories effectively and efficiently executes the branch prediction. 2. Simultaneous cache look up is achieved by Pentium processor due to the separate data and code cache. 3. The separate cache memories raise the system performance i.e. an internal read request is performed more quickly than a bus cycle to memory. 4. They reduce the use of processor's external bus when the same locations are accessed multiple times.	***Separate code and instruction on cache 4M***
	(d) Ans.	**Draw the block diagram of Pentium System Architecture.**	**4M**

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Advanced Microprocessor **Subject Code: 17627**

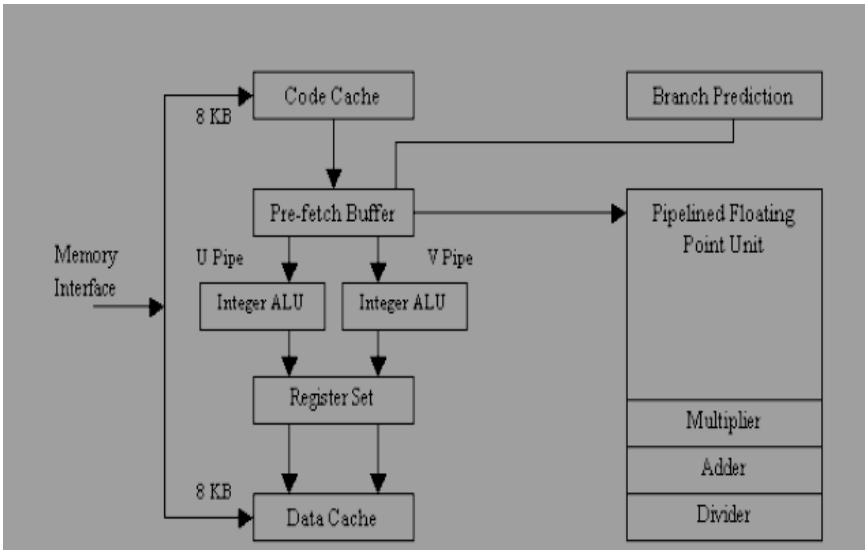


Block diagram of Pentium System Architecture with labels 4M

OR

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Advanced Microprocessor **Subject Code: 17627**

			
	(e) **Ans.**	**List the advantages of RISC. (Any four).** ***(Note: Relevant topic's shorter description shall be considered)*** **Advantages of RISC processor are as follows:** 1. RISC instructions are simple in nature and hence can be hardwired, while CISC architectures nay have to use microprogramming in order to implement microprogramming. 2. A set of simple instructions results in reduced complexity of the control unit and the data path. As a consequence the processor can work at a higher clock frequency and yields greater speed. 3. Several extra functionalities such as MMUs, floating point arithmetic units can also be placed on the same chip. 4. Smaller chips allow the semiconductor manufacturer to place more parts on a single silicon wafer, which can lower the cost of the processor's chip. 5. High level language compilers produce more efficient codes in a RISC processor than CISC, because they tend to use the smaller set of instructions in a RISC computer. 6. Shorter design cycle : a new RISC processor can be designed, developed and tested more quickly since they are simple than CISC processors. 7. Application programmers who use the microprocessor's instructions will find it easier to develop a code with the smaller and optimized instruction set.	**4M** ***Any four advantaged 1M each***

		8. The loading and decoding of the instructions in a RISC processor is simple and fast and it is not needed to wait until the length of the instruction is known in order to start decoding the following one. Decoding is simplified as op-code and address fields are located in the same location for all instructions.	
	(f) Ans.	**Describe any four DOS interrupts.** **INT21** **1) 3CH : to create file** Registers to be used before calling the function using INT 21H: CX=File Attribute DS: DX - full file path (zero terminated) – an ASCIIZ String file descriptor; a start variable in data segment loaded to DX *Syntax:* mov ah,3Ch; function 3Ch - create a file int 21h ; transfer to DOS **2) 3DH: to open file** This function opens the indicated file Registers to be used before calling the function using INT 21H: DS: DX - an ASCIIZ String file descriptor AL=Access Code and sharing modes are as follows 00H- Open for reading mode 01H- open for writing mode 02H – open for read/write mode *Syntax*: mov ah,3Dh; function 3Dh - open the file int 21h; transfer to DOS **3) 3EH: to close the file** This function closes the indicated file Registers to be used before calling the function using INT 21H : BX = file handle *Syntax*: mov ah, 3Eh; function 3Eh - close a file int 21h; transfer to DOS **4) 3FH: to read the file** This function reads up to CX bytes from the Indicated file into the specified memory buffer. On successful return, the AX Register contains the number of bytes actually read. Registers to be used before calling the function using INT 21H: BX = file handle	**4M** ***Description of any four DOS interrupts 1M each***

		CX = number of bytes to read DS:DX -> buffer for data *Syntax:* mov ah,3Fh; function 3Fh – read the file int 21h; transfer to DOS **5) 40H: to write to the file** This function writes the specified number of bytes from a buffer to a file or device. Registers to be used before calling the function using INT 21H: BX = file handle CX = number of bytes to write DS:DX -> data to write *Syntax*: mov ah,40h; function 40h - write to file int 21h; transfer to DOS **6) 41H: to delete the file** This function deletes the specified file Registers to be used before calling the function using INT 21H: ASCIIZ filename DS: DX - zero terminated full paths. *Syntax*: mov ah, 41h; delete file int 21h; transfer to DOS **7) 56H: to rename the file** This functions renames the given file with new name specified by ES: DI Registers to be used before calling the function using INT 21H : DS: DX address of ASCIIZ filename of existing file ES : DI – ASCIZ new filename *Syntax*: mov ah, 56h; delete file int 21h; transfer to DOS **8) 43H: Set/Get file attribute** This function gets or sets the file attributes Registers to be used before calling the function using INT 21H: AL = 00H to get attributes 01H to set attributes CX = file attributes, if AL=01H. Bits can be combined DS: DX = segment: offset of ASCIIZ pathname Syntax: mov ah, 43h; set/get file attributes int 21h; transfer to DOS **9) 57H: Set/Get file time & date** This function gets or sets the file date and time. Registers to be used before calling the function using INT 21H:	

AL = 00h 0r 01H (0 - get 1 - set)
BX = file handle
DS: DX = segment: offset of ASCIIZ pathname
Syntax: mov ah, 57h; set/get file date and time int 21h; transfer to DOS

INT 26H
INT26H Absolute Disk Write
On entry:
AL Drive number (0=A, 1=B)
CX Number of sectors to write

DX Starting sector number
DS:DX Address of sectors to write

Returns: AX Error code (if CF is set; see below)
Flags DOS leaves the flags on the stack
This interrupt reads one or more sectors from a disk drive, and is comparable to the service provided by the ROM BIOS in Interrupt 13h.
INT 25h Absolute Disk Read
eads one or more sectors on a specified logical disk.

On entry:
AL Drive number (0=A, 1=B)
CX Number of sectors to read
DX Starting sector number
DS:DX Buffer to store sector read

Returns: AX Error code (if CF is set; see below)
Flags DOS leaves the flags on the stack

(g) **Differentiate between hardware and software interrupts. (4 points).** **4M**

Ans.

Sr. No.	Hardware interrupts	Software interrupts
1	It is an unsynchronous event	It is synchronous event
2	It is requested by external device or pin.	It is requested by executing instruction.

Any four points 1M each

		3	Program counter is not incremented	Program counter is incremented
		4	The micro processor executes 2 acknowledgement cycles on the receipt of software interrupt	The micro processor does not execute any acknowledgement cycle on the receipt of software interrupt
		5	It can be ignored or masked except for TRAP.	It cannot be ignored or masked
		6	It has comparatively lower priority	It has highest priority among all the interrupts

2. **Attempt any FOUR:** **16**

(a) **Explain with neat diagram DOS. BIOS interface.** **4M**

Ans. **DOS BIOS interface**

Figure shows the DOS-BIOS interface. BIOS contains a set of routines in a ROM to provide the device supports. The BIOS tests and initializes attached devices and provide services that are used for reading to and writing from the devices. One task of DOS is to interface with BIOS when there is a need to access its facilities. When the user program requests a service of DOS, it may transfer the request to BIOS which in turn accesses the requested device.

Sometimes, a program makes a direct request to BIOS, especially for keyboard and screen services.

Explanation 2M

DOS and BIOS Interface :

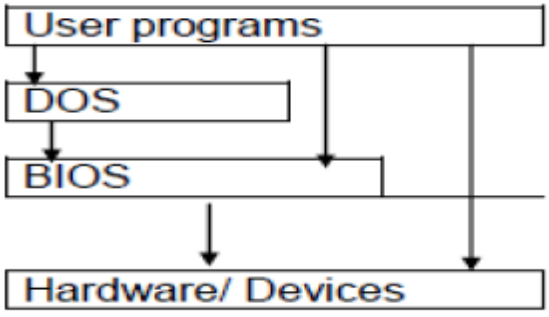


Diagram 2M

	(b) Ans.	**State and explain features of RISC processors. (Any four)** **Features of RISC processors:** 1. **Simple instruction set**: in a RISC machine, the instruction set contains simple basic instructions, from which more complex instructions can be composed. These instructions with less latency are preferred. 2. **Same length instructions**: each instruction is of same length, so that it may be fetched in a single operation. The traditional microprocessors from intel or Motorola support variable length instructions. 3. **Single machine cycle instruction**: Most instructions complete in one machine cycle, which allows the processor to handle several instructions at the same time. RISC processors have unity CPI(clock per instruction), which is due to optimization of each instruction on the CPU and massive pipelining embedded in a RISC processor. 4. **Pipelining**: usually massive pipelining is embedded in a RISC processor. The pipelining is key to speed up RISC machines. 5. **Very few addressing modes and formats**: unlike the CISC processors, where the number of addressing modes are very high. In RISC processors the addressing modes are much less and it supports few formats. 6. **Large number of registers**: the RISC design philosophy generally incorporates a larger number of registers to prevent in large amounts of interactions with memory. 7. **Micro-coding is not required**: Unlike in CISC machines, in RISC architecture, instruction micro-coding is not required. This is because of the availability of a set of simple instructions and simple instructions may be easily built into the hardware. 8. **Load and Store architecture:** the RISC architecture is primarily a Load and Store architecture, implying that all the memory accesses takes place using Load and Store type operations.	**4M** ***Any four features 1M each***
	(c) Ans.	**Describe the floating point unit operation in Pentium processor.**	**4M**

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Advanced Microprocessor **Subject Code: 17627**

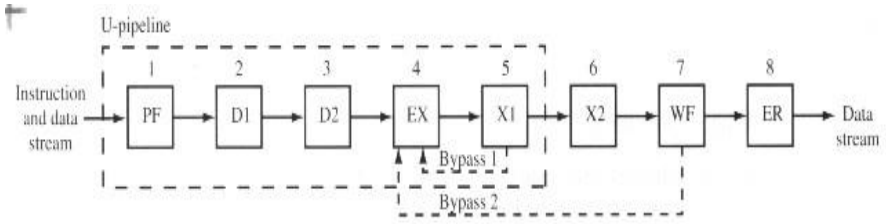


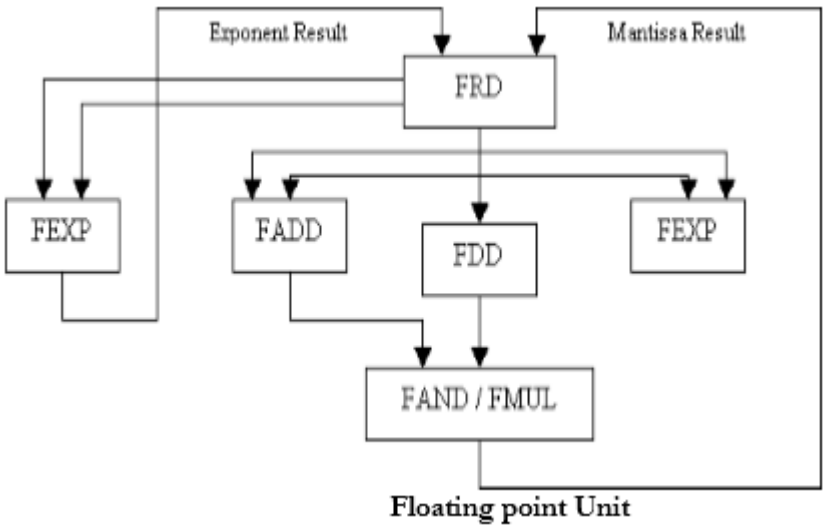
FIGURE 10.16 FPU pipeline stages

Floating point unit diagram 2M

PF	Prefetch
D1	Instruction decode
D2	Address generation
EX	Memory and register read, floating-point data converted into memory format, memory write
X1	Floating-point execute, stage one. Memory data converted into floating-point format, write operand to floating-point register file, bypass 1 (send data back to EX stage)
X2	Floating-point execute stage two
WF	Round floating-point result and write to floating-point register file, bypass 2 (send data back to EX stage)
ER	Error reporting, update status word

Description 2M

OR



Floating point Unit

		Floating Point Pipeline : Floating Point Unit stages in Pentium Processor : The floating point pipeline has **8 stages** as follows: **1. Prefetch (PF) :** • Instructions are prefetched from the on-chip instruction cache **2. Instruction Decode (D1):** • Two parallel decoders attempt to decode and issue the next two sequential instructions • It decodes the instruction to generate a control word • A single control word causes direct execution of an instruction • Complex instructions require micro-coded control sequencing **3. Address Generate (D2):** • Decodes the control word • Address of memory resident operands are calculated **4. Memory and Register Read (Execution Stage) (EX):** • Register read, memory read or memory write performed as required by the instruction to access an operand. **5. Floating Point Execution Stage 1 (X1):** • Information from register or memory is written into FP register. • Data is converted to floating point format before being loaded into the floating point unit. **6. Floating Point Execution Stage 2 (X2):** • Floating point operation performed within floating point unit. **7. Write FP Result (WF):** • Floating point results are rounded and the result is written to the target floating point register. **8. Error Reporting(ER)** • If an error is detected, an error reporting stage is entered where the error is reported and • FPU status word is updated.	
	(d) Ans.	**Explain features of Intel MMX Architecture.** **Features of Intel MMX Architecture:** 1. 57 new microprocessor instructions have been added that are designed to handle video, audio, and graphical data more	**4M**

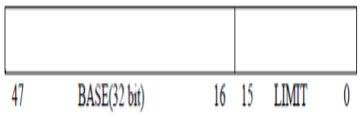
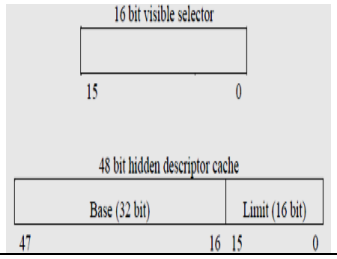
efficiently. Programs can use MMX instructions without changing to a new mode or operating-system visible state.
2. New 64-bit integer data type (Quadword).(4 new MMX data types)
3. A new process, Single Instruction Multiple Data (SIMD), makes it possible for one instruction to perform the same operation on multiple data items.
4. The memory cache on the microprocessor has increased to 32 KB, meaning fewer accesses to memory that is off the microprocessor.
5. 8,64 bits wide MMX technology registers have are added to support the Multimedia .

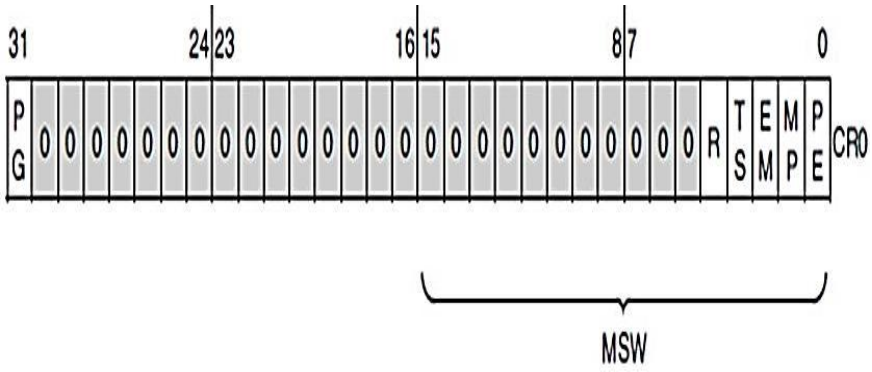
Any four features 1M each

(e) Ans. **Compare between GDTR and LDTR. (four points)** **4M**

Sr. No.	GDTR (Global Descriptor Table Register)	LDTR (Local Descriptor Table Register)
	The Global Descriptor Table Register (GDTR) is a dedicated 48-bit (6 byte) register used to record the base and size of a system's global descriptor table (GDT). Thus, two of these bytes define the size of the GDT, and four bytes define its base address in physical memory. LIMIT is the size of the GDT, and BASE is the starting address. LIMIT is 1 less than the length of the table, then the GDT is 16 bytes long.	The Local Descriptor Table Register (LDTR) is a dedicated 48-bit register that contains, at any given moment, the base and size of the local descriptor table (LDT) associated with the Currently executing task. Unlike GDTR, the LDTR register contains both a "visible" and a "hidden" component. Only the visible component is accessible, while the hidden component remains truly inaccessible to application programs.
	There is no visible component of GDTR.	The visible component of the LDTR is a 16-bit "selector"
	To load the GDTR, LGDT instruction is used.	The dedicated, protected instructions LLDT and SLDT are reserved for loading and storing,

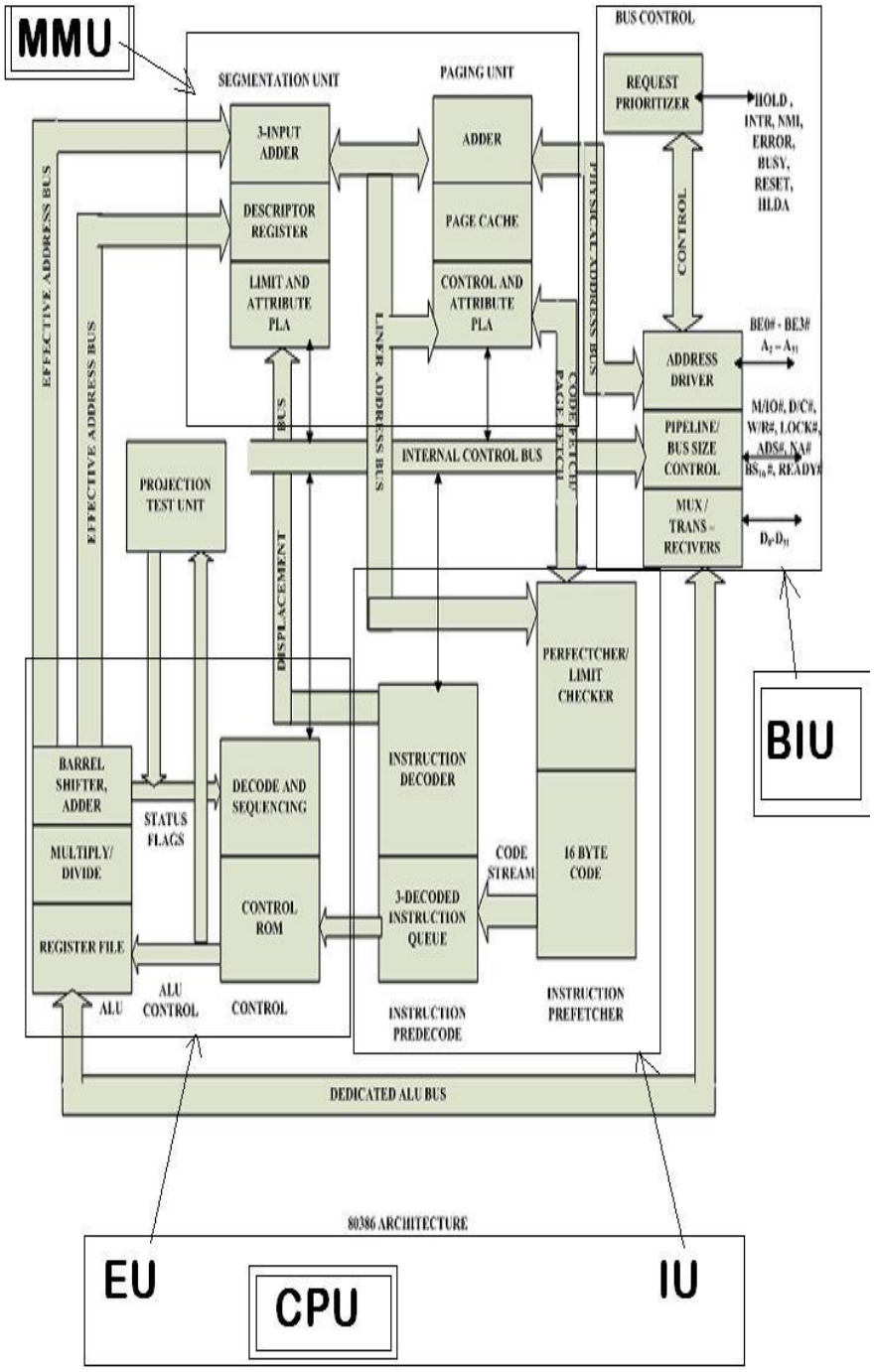
Any four points 1M each

		respectively, the visible selector component of the LDTR register.	
	Structure of GDTR : 	Structure of LDTR : 	
	(f) Ans.	**Draw and explain the format of CRO resister of 80386.** • **CR0** contains system control flags, which control or indicate conditions that apply to the system as a whole, not to an individual task. • **PE** (Protection Enable, bit 0) Setting PE causes the processor to begin executing in protected mode. This can be cleared by resetting the microprocessor. This can be set only in real mode. • **MP** (Monitor processor extension/Coprocessor or Math Present, bit 1) If this bit is set to 1, it allows the Wait instruction to generate a processor extension absent exception i.e. exception number 7.In short when this bit is set to 1 it indicates the absence of coprocessor (processor extension) if its not present and permits the emulation of the processor extension by the CPU. • **EM** (Emulate, bit 2) If this bit is set to 1, it allows the generation of exception 7 (processor extension not present) and will permit the emulation of the processor extension by the CPU.(If this bit is set and the processor extension is absent it will allow the CPU to work as a coprocessor) • **TS** (Task Switched, bit 3) The TS bit of CR0 helps to determine when the context of the coprocessor does not match that of the task being executed by the 80286 CPU. The 80386 sets TS each time it performs a task switch (Whether triggered by software or by hardware interrupt). If, when interpreting one of the ESC instructions, the CPU finds TS already	**4M** ***Explanation 2M***

		set, it causes exception 7. The WAIT instruction also causes exception 7 if both TS and MP are set. Operating systems can use this exception to switch the context of the coprocessor to correspond to the current task. • **ET** (Extension Type, bit 4) ET indicates the type of coprocessor present in the system (ET=0 -> 80287 or ET=1 ->80387) • **PG** (Paging, bit 31) PG indicates whether the processor uses page tables to translate linear addresses into physical addresses. 	***Diagram 2M***
3.	**(a)** **Ans.**	**Attempt any TWO:** **Draw the Architecture of 80386 and explain any two units in detail.**	**16** **8M**

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Advanced Microprocessor **Subject Code: 17627**



Neat labelled architecture if 80386 4M

		The internal architecture of 80386 can be divided into 3 sections as shown in the above figure such as : **1. Central processing unit (CPU)** **2. Memory management unit(MMU)** **3. Bus interface unit(BIU)** **1. The Central processing unit** It consists of **a. Execution unit &** **b. Instruction unit** a. **Instruction unit** has **Instruction prefetcher** and **instruction predecode unit** The Instruction prefetcher fetches the 16 instruction bytes ahead of time and stores them into the 16 byte instruction prefetch queue(16 byte code).This speeds up the program execution process. The instruction predecode unit has the instruction decoder and 3 decoded instruction queue. The instruction decoder decodes 3 instructions ahead of time and stores them in the 3 decoded instruction queue. b. **Execution unit** has **ALU and control unit.** The **control unit** stores the control signals in the control ROM, which are generated at the time of decoding .The decode and sequencing unit decodes the control signals and sends the control signals sequentially to the ALU. **ALU (arithmetic and logic unit):** ALU performs all the arithmetic and logical operations. It has a register file containing registers such as general purpose registers, control and flag registers, debug and test registers, special purpose registers etc. The barrel shifter is of 64 bits which can shift/rotate 64 bits at a time and hence can perform multiplication and divide operations within a microsecond. **2. The memory management unit** It has a. **segmentation unit** and b. **paging unit.** a. **segmentation unit** : The segmentation unit allows the use of two address components such as segment base address and offset address to calculate the	***Explanation of any 2 units 4M***

		physical address. It allows the size of the segment upto 4GB maximum. It provides the 4 level protection level mechanism for protecting and isolating the system's code and data from application programs and unauthorized access. This unit converts logical address spaces to the linear addresses. The Limit and Attribute PLA checks the segment limits and attributes at segment level to avoid invalid access to the code. b. **paging unit.** The paging unit converts the linear addresses to the physical addresses. The control and attribute PLA checks the privileges at page level. Each of the pages maintain the paging information of the task. The paging unit organizes the physical memory in the terms of pages of 4KB each. This unit works under the control of segmentation unit i.e., each segment is further divided into pages. The virtual memory is also organized in the terms of segments and pages by the MMU. **3. Bus Interface Unit :** BIU is responsible for interfacing the microprocessor with the system with the help of all buses. Control of all buses 1.e. address bus, data bus, control bus is in the hands of BIU. The BIU has a bus control unit which has a request prioritizer which resolves the priorities of the various bus request operations. It also controls the access of the bus. The address drivers drives the bus (byte) enable signals BE0#-BE3# and the address signals A0-A31. The pipeline and bus size control unit handle the related control signals and supports the dynamic bus sizing feature. The pipeline unit supports the instruction pipelining feature in 80386. Fetching other instruction while the other is in execution is known as instruction pipelining. The data buffers (mux / transceivers) interface the internal data bus with the system data bus.	

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Advanced Microprocessor **Subject Code:** **17627**

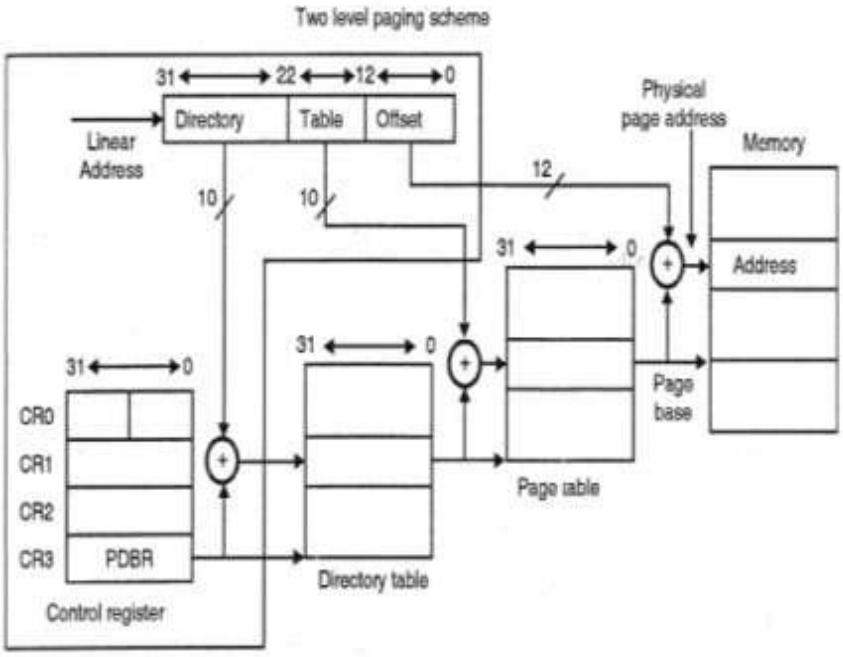
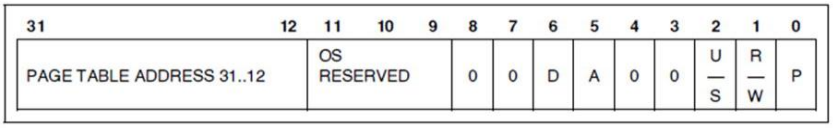
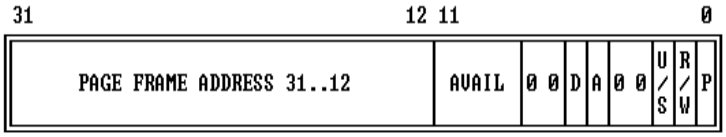
		PROJECTION TEST UNIT : This unit performs the built in self test for the 80386 microprocessor along with the other tests.	
	(b) **Ans.**	**Describe the paging mechanism of 80386 with the help of neat diagram.**  Paging Mechanism • **Page Directory Entry(**Points to Page Table)  	**8M** ***Paging mechanism diagram 4M***

Figure 5-10. Format of a Page Table Entry



P - PRESENT
R/W - READ/WRITE
U/S - USER/SUPERVISOR
D - DIRTY
AVAIL - AVAILABLE FOR SYSTEMS PROGRAMMER USE

NOTE: 0 INDICATES INTEL RESERVED. DO NOT DEFINE.

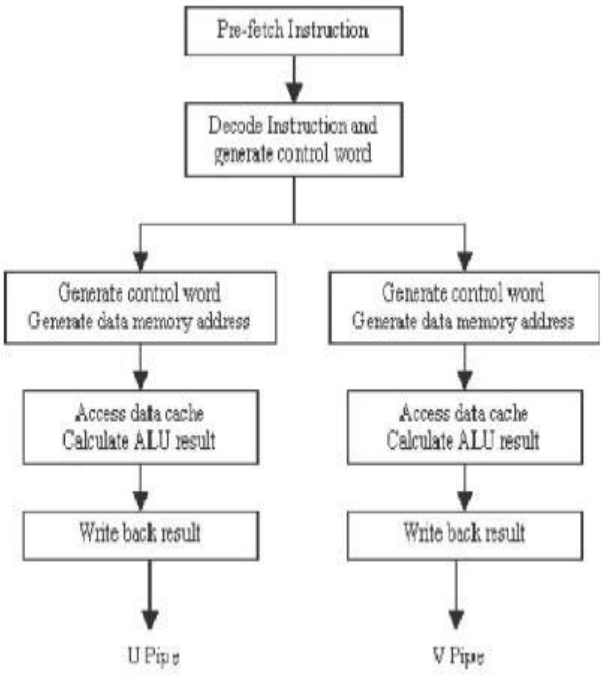
Paging is one of the memory management techniques used for virtual memory multitasking operating system.
• The segmentation scheme may divide the physical memory into a variable size segments but the paging divides the memory into a fixed size pages.
• The segments are supposed to be the logical segments of the program, but the pages do not have any logical relation with the program.
• The pages are just fixed size portions of the program module or data. Thus paging mechanism provides an effective technique to manage the physical memory for multitasking systems.

Description 4M

• **Paging Unit:** The paging unit of 80386 uses a two level table mechanism to convert a linear address provided by segmentation unit into physical addresses.
• The paging unit converts the complete map of a task into pages, each of size 4K. The task is further handled in terms of its page, rather than segments.
• The paging unit handles every task in terms of three components namely page directory, page tables and page itself.

Paging Descriptor Base Register: The control register CR2 is used to store the 32-bit linear address at which the previous page fault was detected.
• The CR 3 is used as page directory physical base address register, to store the physical starting address of the page directory.

		• The lower 12 bit of the CR3 are always zero to ensure the page size aligned directory. A move operation to CR 3 automatically loads the page table entry caches and a task switch operation, to load CR 0 suitably. **Page Directory**: This is at the most 4Kbytes in size. Each directory entry is of 4 bytes, thus a total of 1024 entries are allowed in a directory. • The upper 10 bits of the linear address are used as an index to the corresponding page directory entry. The page directory entries point to page tables. • Page Tables: Each page table is of 4Kbytes in size and many contain a maximum of 1024 entries. The page table entries contain the starting address of the page and the statistical information about the page. The upper 20 bit page frame address is combined with the lower 12 bit of the linear address. The address bits A12- A21 are used to select the 1024 page table entries. The page table can be shared between the tasks. • The P bit of the above entries indicates, if the entry can be used in address translation. • If P=1, the entry can be used in address translation, otherwise it cannot be used. • The P bit of the currently executed page is always high. • The accessed bit A is set by 80386 before any access to the page. If A=1, the page is accessed, else unaccessed. The D bit (Dirty bit) is set before a write operation to the page is carried out. The D-bit is undefined for page director entries. • The OS reserved bits are defined by the operating system software. • The User / Supervisor (U/S) bit and read/write bit are used to provide protection. These bits are decoded to provide protection under the 4 level protection model. • The level 0 is supposed to have the highest privilege, while the level 3 is supposed to have the least privilege. • This protection provide by the paging unit is transparent to the segmentation unit.	
	(c) Ans.	**Draw and explain super scalar execution of Pentium processor.** The five stages pipelining mechanism of Pentium is as shown in the diagram below:	**8M**



Five stages pipelining mechanism diagram 4M

- **The first stage of the pipeline is Prefetch (PF)** stage in which instructions a prefetched from the on chip instruction cache or memory . Because the Pentium processor has separate caches for instructions and data, prefetches no longer conflict with data references for access to the cache. If the requested line is not in the code cache, a memory reference is made. In the PF stage, two independent pairs of line-size (32-byte) prefetch buffers operate in conjunction with the branch target buffer. This allows one prefetch buffer to prefetch instructions sequentially, while the other prefetches according to the branch target buffer predictions. The prefetch buffers alternate their prefetch paths.

Description of all stages 4M

- **The next pipe-line stage is Decode1 (D1)** in which two parallel decoders attempt to decode and issue the next two sequential instructions. The decoders determine whether on e or two instructions can be issued contingent upon the instruction pairing rules described in the section titled "Instruction Pairing Rules."

		The Pentium processor will decode near conditional jumps (long displacement) in the second opcode map (0Fh prefix) in a single clock in either pipe-line. • The D1 stage is followed by **Decode 2 (D2)** in which the address of memory resident operands are calculated. • The **Execute (EX) stage** of the pipe line for both ALU operations and for data cache access; therefore those instructions specifying both an ALU operation and a data cache access will require more than one clock in this stage. In EX all u-pipe instructions and all v-pipe instructions except conditional branches are verified for correct branch prediction. Microcode is designed to utilize both pipe-lines and thus those instructions requiring microcode execute. • The final stage is **Writeback (WB)** where instructions are enabled to modify processor state and complete execution. In this stage v-pipe conditional branches are verified for correct branch prediction. All the registers and memory locations are updated in this stage.	
4.		**Attempt any TWO:**	**16**
	(a)	**Describe the segment descriptor cache registers with suitable diagram in 80386 microprocessor.**	**8M**
	Ans.	**Segment descriptor cache registers:** There are 6 segment descriptor cache registers in 80386 microprocessor. These registers are not available for the users. These registers are associated with the segments and the segment registers in 80386 i.e. CS,DS,ES,SS,FS,GS Every segment descriptor cache register is 72 bits long. Every segment descriptor cache register holds a. 32 bits segment base address b. 32 bits segment limit c. Other required segment attributes. The segment descriptor cache registers are not available for the users. They are invisible to users. The visible part is only 16 bits associated segment registers in the microprocessor.	***Description 4M***

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Advanced Microprocessor **Subject Code: 17627**

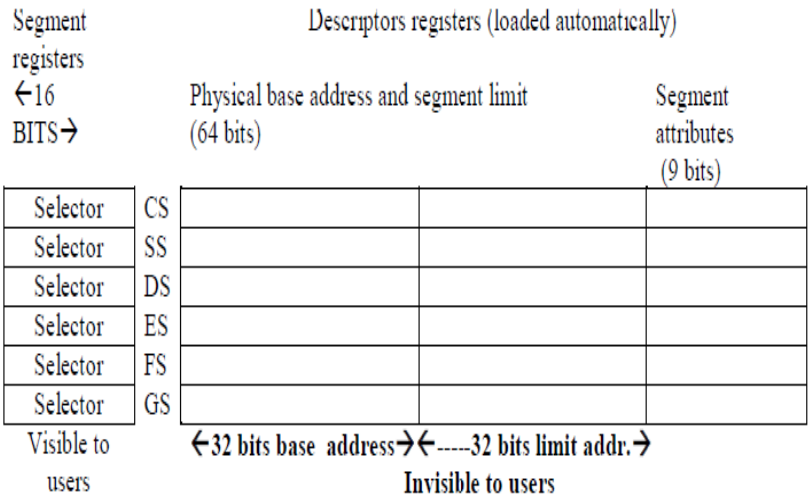


Diagram of segment descriptor cache register of 80386 4M

When a selector is loaded, its associated segment descriptor cache register is automatically get loaded with the values from descriptor table. Either from LDT or GDT.

In the real mode, only the base address is updated directly by shifting the selector values 4 bits to the left.

In the protected mode, the base address, limit and all attributes are loaded.

Base address is the 32 bits base address of the corresponding selected segment in the main memory.

The limit is of 32 bits and shows the maximum size of the segment.

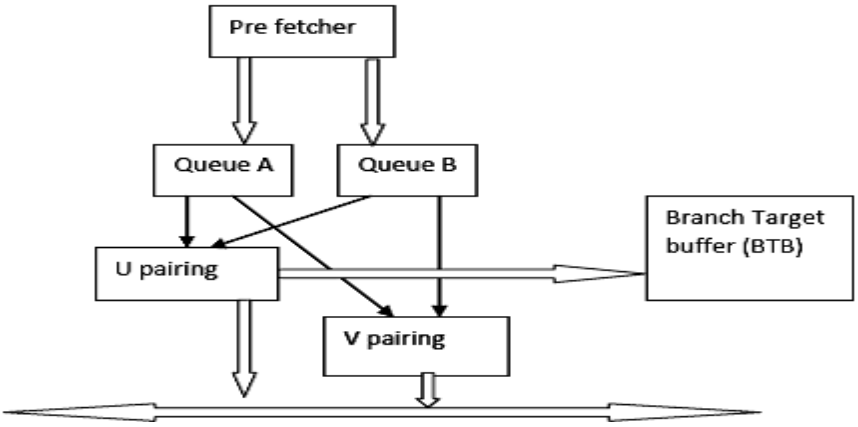
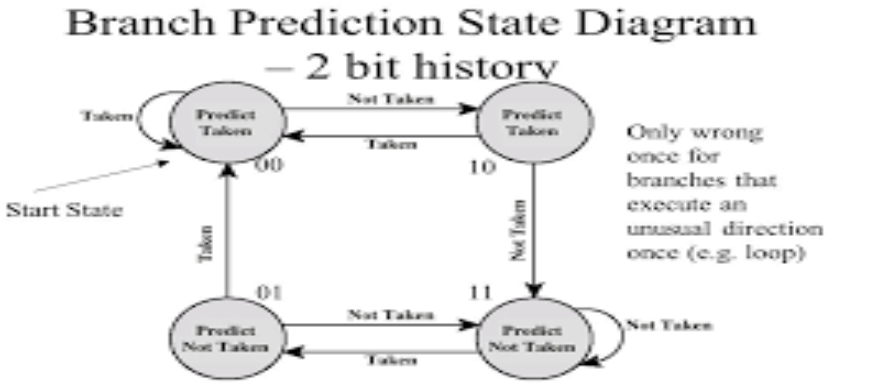
The segment attributes have the access rights of segment like read/write permission, whether the segment address is mapped to physical memory, the descriptor privilege bits etc.

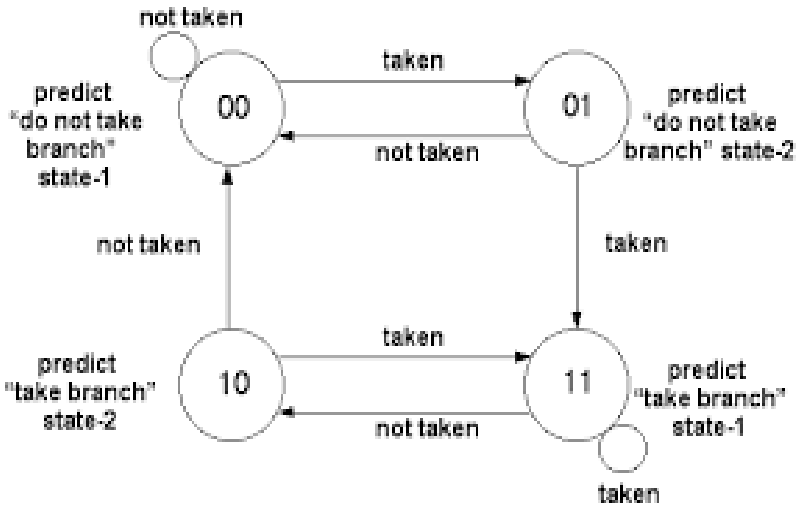
(b) **Describe the concept of branch prediction implemented in Pentium in detail.** **8M**

Ans.

Branch Prediction: The Pentium processor includes branch prediction logic to avoid pipeline stalls, if correctly, predict whether or not branch will be taken when branch instruction is executed if branch prediction is not correct recycle penalty is applicable to u pipeline & 4 cycle penalty if branch is related to v pipeline. The branch instructions occur frequently while running any application. These instructions change the normal sequential control flow of the program and may stall the pipelined execution in the Pentium system. Branches may be of two types: Conditional branch and unconditional branch. In case of conditional branch, the CPU has to wait till the execution stage to

Branch prediction explanation 4M

		determine whether the condition is met or not. The Pentium processor makes the dynamic branch prediction using a Branch Target Buffer (BTB). To efficiently predict branches, the Pentium uses two prefetch buffers. One buffer prefetches code in linear fashion, while the other prefetches instructions based on address in the branch target buffer. As a result the needed code is prefetched before it is required for execution.The Pentium processors prediction algorithm not only forecast the simple branch choices but also supports more complex branch prediction for example, within nested loops. This is achieved by storing multiple branch address in the branch prediction buffer. The design of the branch target buffer allows 256 addresses to be stored and thus the prediction algorithm can forecast up to 256 branches  **OR**  **OR**	***Diagram 4M***



(c) **Describe the inter MMX architecture w.r.t. its data types & instruction set.** **8M**

(Note : the instruction sets of MMX are not expected in detail from students ,but the registers for the access to these instructions can be explained and shall be given marks)

Ans. **Instuction sets/ registers:**

There are 57 new SIMD instructions added to the instructions set of intel MMX for the support of the multimedia functions.

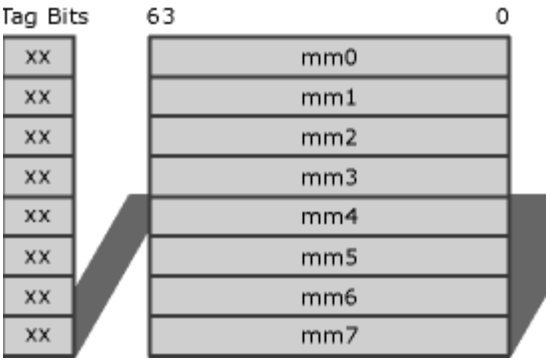
For these instructions new 8 general purpose registers have been introduced.

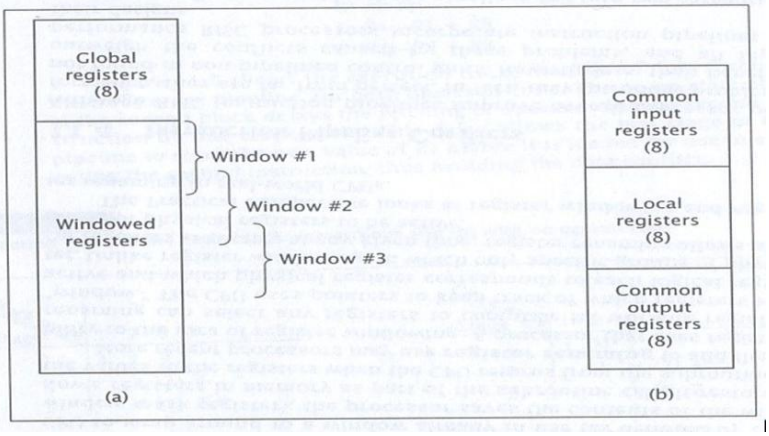
The description of which are as given below:

1. In Pentium there are eight general purpose floating point registers in a floating point unit.
2. Each of these eight registers are 80-bit wide for floating point operations, 64 bits are used for mantissa and rest of 16 bit for exponent.
3. Intel MMX instructions use these floating point registers as MMX registers and used only 64 bit mantissa portion of these registers to store MMX operands.
4. Thus MMX programmers virtually get new MMX registers each of 64bits.
5. It is possible to use same set of registers as floating point registers and MMX register in the same program; it is preferable not to use them concurrently.

intel MMX architecture w.r.t. its data types 4M

instruction sets/registers 4M

		6. After a sequence of MMX instruction is executed, these registers should be cleared by an instruction 'EMMS' which implies empty MMX stack. 7. The floating point users should use same instruction after executing floating point instructions. 8. Although content switching between multimedia program execution and floating point execution is permissible. It is not recommended. 9. It is advisable that multimedia program developers should partition MMX instruction into separate library routine.  **The MMX technology supports the following four data types.** 1. Packed bytes-In this data types, eight bytes can be packed into one 64 bit quantity. 2. Packed word-Four words can be packed into 64 bit. 3. Packed double word-Two double words can be packed into 64 bit 4. One quadword-One single 64 bit quantity.	
5.		**Attempt any TWO:**	**16**
	(a)	**Mention design issues involved in the design of RISC processors. Describe any one issue in detail.** ***(Note: Any other valid issue shall be given marks)***	**8M**
	Ans.	**The design issues involved in the design of RISC processor are :** **1. Register window** **2. Memory speed** **3. Instruction latency** **4. Dependency** **1. Register Window :** The reduced hardware requirements of RISC processors leave additional space available on the chip for the system designer. RISC	***List design issues 2M***

		CPUs generally use this space to include a large number of registers (> 100 occasionally). The CPU can access data in registers more quickly than data in memory so having more registers makes more data available faster. Having more registers also helps reduce the number of memory references especially when calling and returning from subroutines. The RISC processor may not be able to access all the registers it has at any given time provided that it has many of it. Most RISC CPUs have some global registers which are always accessible. The remaining registers are windowed so that only a subset of the registers is accessible at any specific time. To understand how register windows work, we consider the windowing scheme used by the Sun SPARC processor. The processor can access any of the 32 different registers at a given time. (The instruction formats for SPARC always use 5 bits to select a source/destination register which can take any 32 different values. Of these 32 registers, 8 are global registers that are always accessible. The remaining 24 registers are contained in the register window. The register window overlap. The overlap consists of 8 registers in SPARC CPU. Notice that the organizations of the windows are supposed to be circular and not linear; meaning that the last window overlaps with the first window. ***Example:*** the last 8 registers of window 1 are also the first 8 registers of window 2. Similarly, the last 8 registers of window 2 are also the first 8 registers of window 3. The middle 8 registers of window 2 are local; they are not shared with any other window. FIGURE 11.4 Register windowing in the SPARC processor  	***Explanation of any one issue in detail 6M***

		2. **Memory speed issue:** Memory speed issues are commonly solved using caches. A cache is a section of fast memory placed between the processor and slower memory. When the processor wants to read a location in main memory, that location is also copied into the cache. Subsequent references to that location can come from the cache, which will return a result much more quickly than the main memory. Caches present one major problem to system designers and programmers, and that is the problem of coherency. When the processor writes a value to memory, the result goes into the cache instead of going directly to main memory. Therefore, special hardware (usually implemented as part of the processor) needs to write the information out to main memory before something else tries to read that location or before re-using that part of the cache for some different information. **3. Instruction Latency issue:** A poorly designed instruction set can cause a pipelined processor to stall frequently. Some of the more common problem areas are: Highly encoded instructions such as those used on CISC machines that require complex decoders. Those should be avoided. Variable-length instructions which require multiple references to memory to fetch in the entire instruction. Instructions which access main memory (instead of registers), since main memory can be slow. Complex instructions which require multiple clocks for execution (many floating-point operations, for example.)Instructions which need to read and write the same register. For example "ADD 5 to register 3" had to read register 3, add 5 to that value, then write 5 back to the same register (which may still be "busy" from the earlier read operation, causing the processor to stall until the register becomes available.) Dependence on single-point resources such as a condition code register. If one instruction sets the conditions in the condition code register and the following instruction tries to read those bits, the second instruction may have to stall until the first instruction's write completes. **4. Dependencies issues:** One problem that RISC programmers face is that the processor can be slowed down by a poor choice of instructions. Since each instruction takes some amount of time to store its result, and several instructions are being handled at the same time, later instructions may have to wait for the results of earlier instructions to be stored. However, a simple rearrangement of the	

		instructions in a program (called Instruction Scheduling) can remove these performance limitations from RISC programs.	
	(b) Ans.	**List and explain features of Sun Ultra SPARC processor. (Any 8)** It contains an integer unit, a FPU and a optional coprocessor. The 64 bits Ultra SPARC architecture has following features : 1. It has 14 stages non-stalling pipeline. 2. It has 6 execution units including two for integer, two for floating point, one for load/store and one for address generation units. 3. It has a large number of buffers but only one load/store unit, it dispatches them one instruction at a time from the instruction stream. 4. It contains 32KB L1 instruction cache, 64KB L1 data cache, 2KB prefetch cache and 2 KB write cache. It also has 1MB on chip L2 cache. 5. Like Pentium MMX it also contains the instructions to support multimedia. These instructions are helpful for the implementation of image processing codes. 6. One of the major limitations of SPARC system is its low speed compared to most of the modern processors. 7. SPARC stores multi-byte numbers using BIG Indian format, i.e. the MSB will be stored at the lowest memory address. 8. It supports a pipelined floating point processor. The FPU has 5 separate functional units for performing the floating point operations. The floating point instructions can be issued per cycle and executed by the FPU unit. The source and data results are stored in 32 register files. Majority of the floating point instructions have a throughput of one cycle and a latency of three cycles. Although the single precision (32 bit) or double precision (64 bit) floating point computations can be performed by hardware, quad precision i.e. 128 bits operation can be performed only in the software.	**8M** ***Any eight features of sun ultra SPARC processor 1M each***
	(c) Ans.	**Distinguish between DOS and BIOS interrupts. (Any 8 points)** ***(Note: Any other valid difference shall be given marks)***	**8M**

SUMMER – 2019 EXAMINATION
MODEL ANSWER

Subject: Advanced Microprocessor **Subject Code: 17627**

		<table><tr><th>Sr. No.</th><th>DOS</th><th>BIOS</th></tr><tr><td>1</td><td>They are stored in memor y system</td><td>They are stored in system RO M</td></tr><tr><td>2</td><td>DOS interrupts provide DOS utilities</td><td>BIOS Interrupts provide basic IO utilities.</td></tr><tr><td>3</td><td>They provide the interface between BIOS and Devices</td><td>They provide interface between hardware and DOS</td></tr><tr><td>4</td><td>DOS interrupt calls are a facility that operating systems and application programs use to invoke the facilities of the operating System</td><td>BIOS interrupt calls are a facility that operating systems and application programs use to invoke the facilities of the Basic Input/ Output System on computers.</td></tr><tr><td>5</td><td>They are stored in locations 00000H to 003FFH.</td><td>They are stored in locationsFE000H to FFFFFH.</td></tr><tr><td>6</td><td>They are int 21H, int 22h, int 23h etc</td><td>They are INT 10h, 14h, 16h,17h</td></tr><tr><td>7</td><td>DOS interrupts provide the functions like termination of program, disk read, disk write etc.</td><td>BIOS interrupts provide display functions, printer functions, keyboard functions.</td></tr><tr><td>8</td><td>Example: int 21h function 3ah is used to create a directory</td><td>Example : int 10h function 02h is used to locate the cursor on screen on some position</td></tr></table>	***Any eight points 1M each***
6.	**(a)**	**Attempt any FOUR:**	**16**
	Ans.	**Explain the interrupt processing sequence of X86 microprocessor.**	**4M**
		Interrupt processing sequence is as given below: When INT n instruction is executed: 1. The processor pushes flag register on stack then the contents of CS and IP register on stack 2. It clears two flags TF (trap flag) and IE (Interrupt enable flag).	

		3. Number of interrupt is used to find correct address of ISR in the IVT. 4. Interrupt number (is called as interrupt type) is used to find out the correct address of ISR in the IVT. 5. The interrupt number is multiplied by 4 to get the address with the IVT that contains the addresses of ISR. ISR ADDRESS = Interrupt type x 4 6. All addresses are 4 bytes long. The Interrupt vector address is then filled in CS and IP register. 7. Finally CPU control is transferred to new address. 8. It decrements stack pointer by 2 & push flag register on stack. 9. It clears the interrupt request by clearing interrupt flag. 10. It also reset trap flag in flag register. 11. Decrement stack pointer by 2 & store code segment in it. 12. Decrement stack pointer by 2 & pushes IP in it. 13. If fetches the ISR & jumps on it. After the completion of ISR, it decodes the instruction IRET & retrieves the main program address & status of flag register.	***Interrupt processing sequence 4M***
	(b) **Ans.**	**Differentiate between Hardware & Software interrupts. (any four)** **Sr. No. | Hardware interrupts | Software interrupts** 1 | It is an unsynchronous event | It is synchronous event 2 | It is requested by external device or pin. | It is requested by executing instruction. 3 | Program counter is not incremented | Program counter is incremented 4 | The micro processor executes 2 acknowledgement cycles on the receipt of software interrupt | The micro processor does not execute any acknowledgement cycle on the receipt of software interrupt 5 | It can be ignored or masked except for TRAP. | It cannot be ignored or masked 6 | It has comparatively lower priority | It has highest priority among all the interrupts	**4M** ***Any four points 1M each***

(c) Compare between CISC and RISC processor designs. (Any 4 points) — **4M**

Ans.

Sr. No.		CISC	RISC
1	Acronym	It stands for 'Complex Instruction Set Computer	It stands for 'Reduced Instruction Set Computer'.
2	Definition	The CISC processors have a large set of instructions with many addressing nodes.	The RISC processors have a smaller set of instructions with few addressing nodes.
3	Emphasis	Emphasis on hardware	Emphasis on software
4	Memory unit	It has a memory unit to implement complex instructions.	It has no memory unit and uses a separate hardware to implement instructions.
5	Program	It has a microprogramming unit.	It has a hard-wired unit of programming.
6	Time	Execution time is very high.	Execution time is very less
7	Design	It is an easy compiler design	It is a complex compiler design
8	Applications	Used in low end applications such as security systems, home automations, etc.	Used in high end applications such as video processing, telecommunications and image processing.

Any four points 1M each

(d) Describe any four architectural features of Pentium 3 processor. — **4M**

(Note: Any other significant features to be given marks)

Ans.

Features of Pentium 3 processor:

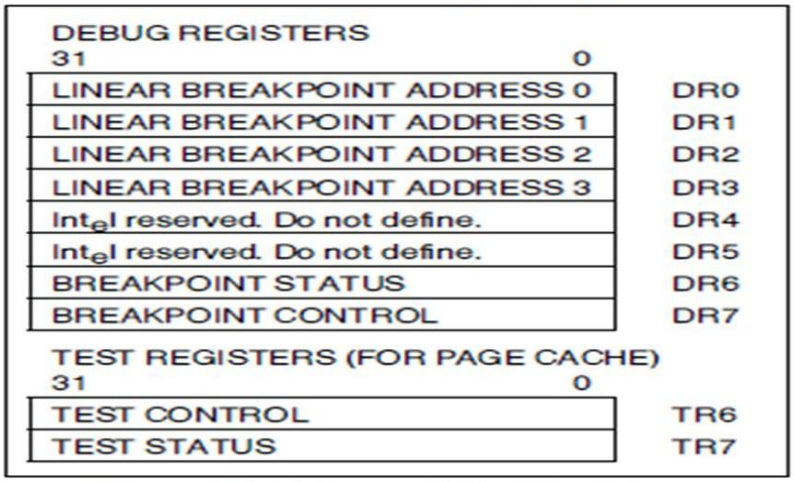
1. Pentium III processor has 512KB full speed on chip L2 Cache with ECC(ERROR CORRECTING CODE) for high performance workstations/ servers. Can work on WINDOWS 98, WINDOWS NT,

		2000, LINUX OS. 2. PIII is incorporated with MMX technology. 3. Dynamic execution, micro-architecture incorporates unique combination of multiple branch prediction, data flow analysis and speculative execution. 4. The Pentium III processor has Multiple low power states. 5. Pentium III is optimized for 32 bits applications running on advanced 32 bits OS. 6. It has 32KB L1 cache divided as 16KB instruction cache and 16KB data cache. 7. It has Quad quad word wide i.e. 256 bits cache data bus, ways set associative cache 8. It provides improved cache hit rate. 9. It supports Multiprocessor system. 10.It Works on 1.0 GHz, 850, 800, 750, 700, 650 MHZ.	***Any four features 1M each***
	(e) **Ans.**	**List and explain advantages of Pentium-2 processor over Pentium – Pro.** **Advantages of Pentium 2 processor over Pentium pro are as follows:** 1. Dynamic Execution Technology: ⇒ Dynamic execution incorporates the concepts of out-of-order and speculative execution. The Pentium II processor's implementation of these concepts removes the constraint of linear instruction sequencing between the traditional fetch and execute phases of instruction execution. Up to 3 instructions can be decoded per clock cycle. These decoded instructions are put into a buffer, which can hold up to 40 instructions. Instructions are executed from this buffer when their operands are available (versus instruction order). Up to 4 instructions can be executed per clock cycle. 2. Super pipelining: ⇒The pipeline of the P6 processor family consists of approximately 12 stages, versus 5 for the Pentium processor and 6 for the Pentium processor with MMX technology. This enables the Pentium II processor to achieve about a 50% higher frequency than the Pentium processor on the same manufacturing technology. The sophisticated, two-level, adaptive training, branch prediction mechanism of the Pentium II processor is key to maintaining the efficiency of the super pipelined micro architecture.	**4M** ***Any four advantages of Pentium -2 processor over Pentium -Pro for 1M each***

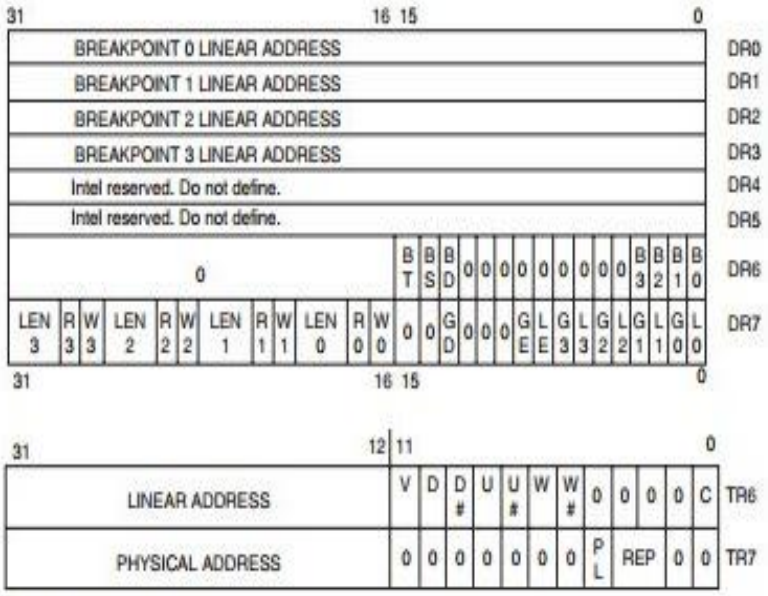
		3. Dual Independent Bus (DIB) Architecture: ⇒ This architecture consists of two distinct buses emanating from the Pentium II processor: the L2 cache bus and the system bus (used for memory and I/O requests). The L2 cache bus speed scales with processor frequency. For the Pentium II processor at 266 MHz, the L2 cache bus operates at 133 MHz, which is twice the speed of Pentium processor systems. The system bus for both processors runs at 66 MHz. The net result is that the Pentium II processor at 266 MHz has about 3 times the peak bus bandwidth of the highest speed Pentium processor system, which has but one bus running at a peak of 66 MHz. Also, since speed of L2 cache accesses is one of the more important system factors in determining overall performance, system performance will scale well with higher processor frequencies. Unlike the Pentium processor's system bus, the Pentium II processor's system bus supports up to 8 outstanding bus requests (4 per processor). This allows more parallelism between processors and I/O, as well as supporting smooth performance scaling to a 2 processor system. 4. High Performance Intel MMX Technology: ⇒ Intel's MMX media enhancement technology is a major extension of the Intel Architecture that makes PCs into richer multimedia and communications platforms. This technology introduces 57 instructions oriented to highly parallel operations with multimedia and communications data types. These instructions use a technique known as SIMD (Single Instruction, Multiple Data) to deliver better performance for multimedia and communications computation. Intel processors that provide MMX technology support are fully compatible with previous generations of the Intel Architecture and the installed base of software. ⇒To further improve performance, the Pentium II processor, like the Pentium processor with MMX technology, can execute 2 Intel MMX instructions at a time. 5. Write Combining: ⇒ The Write Combining technology of the P6 architecture can be used to achieve very high graphics I/O performance. This feature combines multiple writes to a region of memory (for example, a video controller's frame buffer) declared as WC type into a single-burst write operation. This is well suited for the bus, which is optimized for burst transfers. The combining also leads to burst writes of cache line sizes. These writes are	

		further combined by the chipset, leading to high throughput for graphics I/O. The result is enhanced multimedia performance, more realistic full-motion video, and realistic, fast graphics performance. 6. Caches: ⇒ The Pentium II processor has 32 KB of non-blocking L1 cache, which is divided into a 16K instruction cache and a 16K data cache. Each of these caches runs at the processor frequency and provides fast access to heavily used data. ⇒ The Pentium II processor has a 512K L2 cache which is unified for code and data, and is non-blocking. There is a dedicated 64-bit bus to facilitate higher data transfer rates between the processor and the L2 cache. 7. The floating-point pipeline supports the 32-bit and 64-bit IEEE 754 formats as well as the 80-bit format. The FPU is object code-compatible with the Pentium and 486 processor FPUs. 8. The GTL+ (gunning trans-receiver logic) bus provides glue-less support for two processors, giving a cost-effective SMP (symmetric multiprocessing technology) solution. This can be used to significantly enhance OS and application performance in multithreaded or multitasking environments or for functional redundancy checking. 9. Testing and Performance Monitoring Features: ⇒ Built InSelf Test (BIST) provides single stuck-at fault coverage of the microcode and large PLAs, as well as testing of the instruction cache, data cache, Translation Look a side Buffers (TLBs), and ROMs. ⇒ IEEE* 1149.1 Standard Test Access Port and Boundary Scan Architecture mechanism allows testing of the Pentium II processor through a standard interface. ⇒ Internal performance counters can be used for performance monitoring and event counting 10. Pentium-II works on 233,266,333,350,400 and 450MHZ which is more than Pentium pro.	
	(f) **Ans.**	**Draw the diagram showing debug and test registers of 80386 processor.**	**4M**

Debug and Test Registers



OR



Correct diagram 4M