Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in themodel answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given moreImportance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in thefigure. 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 constantvalues may vary and there may be some difference in the candidate's answers and modelanswer.
6) In case of some questions credit may be given by judgement on part of examiner of relevantanswer based on candidate's understanding.
7) For programming language papers, credit may be given to any other
program based on equivalent concept.

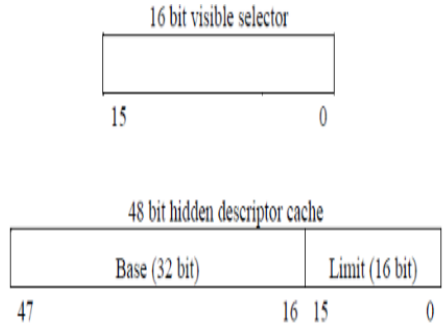
Q.1 Attempt any FIVE of the following **20M**

a. List any eight features of 80386 processor.
(8 Features of 80386 processor : ½ mark each)
Answer:

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 a integrated memory management unit which supports the virtual memory and four levels of protections.
8. It has a on chip clock divider circuitry.
9. It has BIST (built in self test) feature which tests approximately 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 pre fetch 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 DR_0-DR_7 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. Distinguish between LDTR and GDTR (4 POINTS).
(Any 4 differences between LDTR and GDTR 1 mark each)
Answer:

Sr. no.	LDTR (LOCAL DESCRIPTOR TABLE REGISTER	GDTR (GLOBAL DESCRIPTOR TABLE REGISTER)
1	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)	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

	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.	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.
2	The visible component of the LDTR is a 16-bit "selector"	There is no visible component of GDTR.
3	The dedicated, protected instructions LLDT and SLDT are reserved for loading and storing, respectively, the visible selector component of the LDTR register.	To load the GDTR , LGDT instruction is used.
4	Structure of LDTR :  	Structure of GDTR : 

c. List any eight salient features of Pentium.
(Any eight features of Pentium ½ mark each)

Answer:

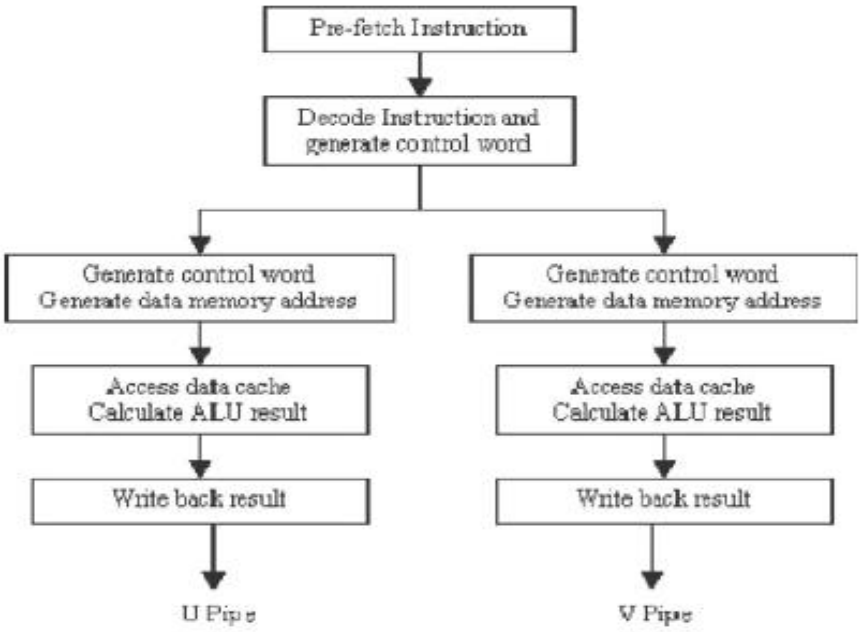
1. Pentium processor has 64 bit data bus
   - 8 bytes of data information can be transferred to and from memory in a single bus cycle with the help of 64 bits data lines.
   - It supports burst read and burst write back cycles
   - It supports pipelining
2. It has a separate Instruction cache
   - Pentium processor has 8 KB of dedicated instruction cache
   - It has Two Integer execution units, one Floating point execution unit
   - It has a Dual instruction pipeline

- It has 256 lines between instruction cache and prefetch buffers; allows 32 bytes to be transferred from cache to buffer

3. It has a separate Data cache
   - It has a 8 KB dedicated data cache gives data to execution units
   - It has 32 byte lines .
4. Pentium processor has Two parallel integer execution units
   - It Allows the execution of two instructions to be executed simultaneously in a single processor clock
5. It has a Floating point unit for Faster internal operations
6. It has a Local advanced programmable interrupt controller , it speeds up upto 5 times for common operations including add, multiply and load, than 80486
7. It has a Branch Prediction Logic
   - To reduce the time required for a branch caused by internal delays.
   - When a branch instruction is encountered, microprocessor begins prefetch instruction at the branch address.
8. It has a Data Integrity and Error Detection logic
   - Has significant error detection and data integrity capability.
   - Data parity checking is done on byte – byte basis.
   - Address parity checking and internal parity checking features are added.
9. It has a Dual Integer Processor which allows execution of two instructions per clock cycle
10. It has a Functional redundancy check to provide maximum error detection of the processor and interface to the processor .
    - A second processor 'checker' is used to execute in lock step with the 'master' processor.
    - It checks the master's output and compares the value with the internal compute d values.
    - An error signal is generated in case of mismatch
11. It has a Superscalar architecture**,** which has Three execution units
    - One execution unit executes floating point instructions.
    - The other two (U pipe and V pipe) execute integer instructions.
    - Parallel execution of several instructions – superscalar processor.

d. Describe fire state pipelining mechanism of Pentium with neat diagram.
(Five stages pipelining mechanism diagram 2 mark , description of all stages 2 marks)
Answer:

The five stages pipelining mechanism of Pentium is as shown in the diagram below:



- **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.

- **The next pipe-line stage is Decode1 (D1)** in which two parallel decoders atte mpt to decode and issue the next two sequential instructions. The decoders determine whether on e or two instructions can be issued contingent upon the ins truction 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.

e. What is RISC ? Explain in brief.
(Definition of RISC 1 MARK, description/ features of RISC 3MARKS)

Answer:
RISC, or Reduced Instruction Set Computer is a type of microprocessor architecture that utilizes a small, highly-optimized set of instructions, rather than a more specialized set of instructions often found in other types of architectures.

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.

f. List any eight features of sun ultra sparc processor.
(Any eight features of sun ultra sparc processor 1/2 mark each)
Answer:

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 endian 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.

g. Describe maskable and non-maskable interrupts of X-86 processor.
(Description of maskable interrupts 2 marks and non-maskable interrupts 2 marks)
Answer:

Maskable interrupts: These are the most common way used by the X86 processor to respond to asynchronous external hardware events.Hardware interrupts occurs when the INTR is pulled high and the Interrupt flag bit is enabled.
The processor only responds to interrupts between instructions.
When an interrupt occurs the processor reads the 8 bit vector code of interrupt supplied by hardware which identifies the source of interrupt (one of the 224 user defined interrupts.)
The IF bit in the flag register is reset when as interrupt is being serviced. This effectively disables servicing additional interrupts during an Interrupt service routine.
To allow nesting of interrupts this IF bit can be set explicitly by interrupt handler.
When an IRET instruction is executed the original state of IF is restored.

Non maskable interrupts: Non maskable interrupts provide a method of servicing very high priority interrupts. NMI is an example of non maskable interrupt . It is an external pin to the microprocessor.
A common example of the use of non maskable interrupt (NMI) would be to activate a power failure routine.
When a NMI is pulled high it causes an interrupt with an internally supplied vector value of 2.
No interrupt acknowledgement cycle is performed by the processor when NMI occurs.
While executing NMI, no further NMI is serviced until the next IRET instruction is executed or the processor is reset.
If NMI occurs at the time of servicing a NMI, its occurrence will be saved and it will be processed when the servicing of the first will be over.
The IF bit is cleared at the beginning of NMI interrupt to inhibit further INTR requests.

Q. 2 Attempt any TWO of the following: 16M

a. Describe paging mechanism with suitable diagram in 80386 processor with TLB.
(Paging mechanism with TLB diagram 4 marks, description 4marks)

Answer:
Paging mechanism with TLB:
As the conversion of 32 bits linear address to physical address is too longer, the Paging unit of 80386 uses TLB.
The paging unit receives a 32 bit linear address from the segmentation unit. The structure of linear address is shown below.

The upper 20 linear address bits (A12 – A31) are compared with all the entries in the translation look aside buffer to check if it matches with any of the entries. If it matches, the 32 bit physical address is calculated from the matching TLB entry and placed on the address bus. For converting all t he linear addresses to physical addresses, if the conversion process uses the two level paging for every conversion, a considerable time will be wasted in the process. Hence to optimize this, a 32 entry page table cache is provided which stores the 32 recently accessed page table entries. Whenever a linear address is to be converted to physical address, it is first checked to see,

whether it corresponds to any of the page table cache entries. This page table cache is also known as Translation Look-aside buffer (TLB).
If the page table entry is not in the TLB, the 80386 reads the appropriate page directory entry. It then checks the P bit of the directory entry. If P=1, it indicates that the page table is in the memory. Then the 80386 refers to the appropriate page table entry and sets the accessed bit A. If P=1, in the page table entry, the page is available in the memory. Then the processor updates the A and D bits and accesses the page. The upper 20 bits of linear address, read from the page table are stored in TLB for future possible access. If P=0, the processor generates a page fault exception (Interrupt number 14).When a page fault exception is generated, CR2 is loaded with the page fault linear address.

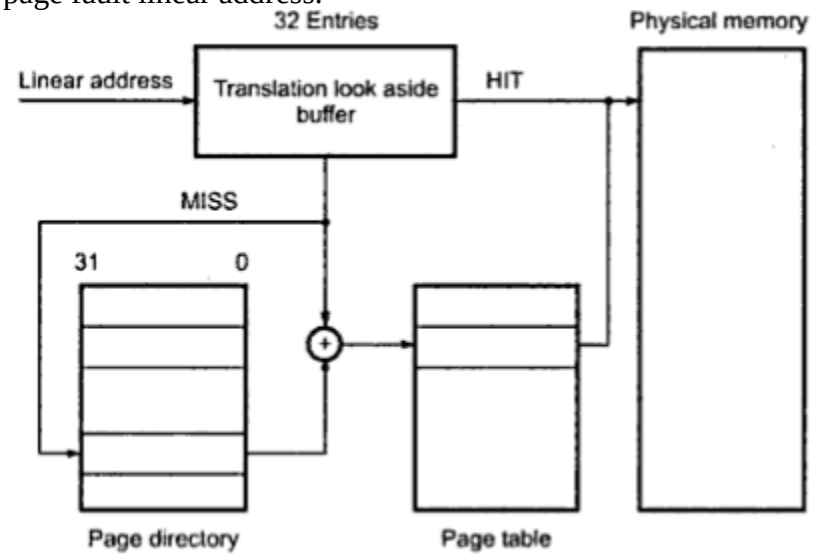
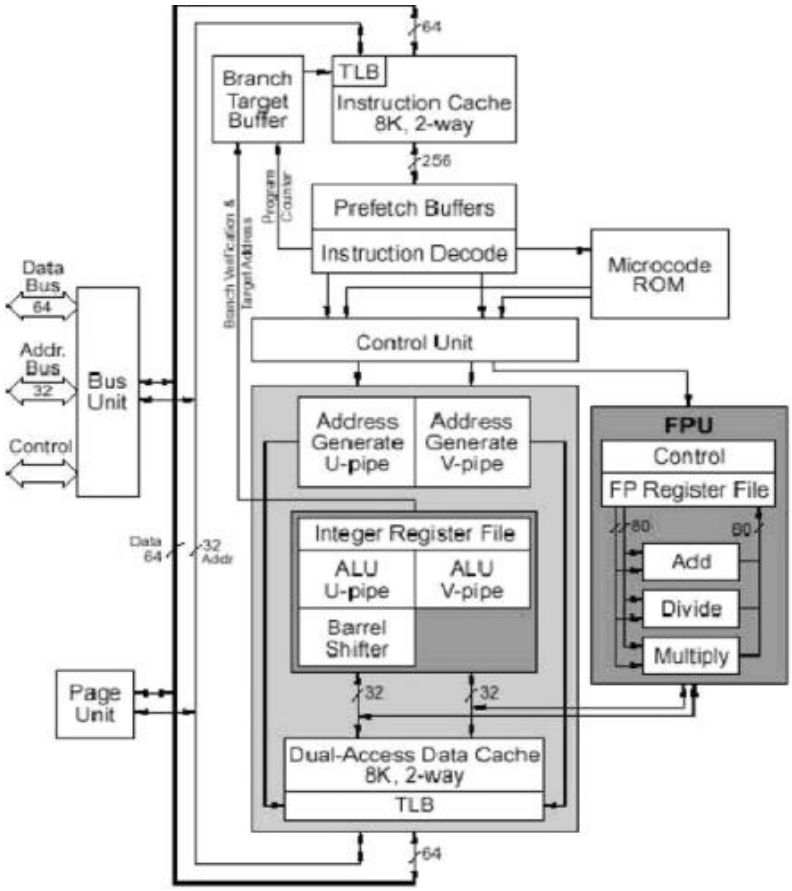


Fig. Translation lookaside buffer

b. Draw the block diagram of Pentium system architecture and explain each block in it.
(Block diagram of Pentium 4 marks, description of blocks 4 marks.)

Answer:
Pentium Block diagram



- It has two integer pipelines U and U. they are responsible for executing 80x86 instructions.
- A floating point unit is included on to the chip to execute instructions previously handled by the external 80X87 math processor. During execution U and U pipeline are capable of executing two integers in terms at the same time.
- It communicates with outside world with 32 bit address bus and 64 bit data bus.
- The bus unit is capable of bus read & write of 32 bytes to memory & through bus cycle to memory be in progress simultaneously.
- An 8 kB instruction cache is used to provide quick access to frequency used instruction. If an instruction is not found in cache then it is read from external memory & copy is placed in cache future reference.
- The branch target buffer and pre-fetch buffer works together with item cache to fetch instruction as fast as possible. The pre-fetch buffer maintains a copy of the next 32 byte of pre-fetched instruction code.
- Branch prediction technique is used to maintain a steady flow of instruction in to the pipeline.
- A separate 8 kB data cache is required to store a copy of frequently accessed instructions.

- Both instruction cache and data cache stores a copy can be enabled or disabled with hardware or software.
- A translation look side buffer converts logical address into physical address when virtual memory is employed.

Register set: The purpose of the Register is to hold temporary results, and control the execution of the program. General-purpose registers in Pentium are EAX, ECX, EDX, EBX, ESP, EBP,ESI, or EDI.
The 32-bit registers are named with prefix E, EAX, etc, and the least 16 bits 0-15 of these registers can be accessed with names such as AX, SI Similarly the lower eight bits (0-7) can be accessed with names such as AL & BL. The higher eight bits (8-15) with names such as AH & BH. The instruction pointer EAP known as program counter(PC) in 8-bit microprocessor, is a 32-bit register to handle 32-bit memory addresses, and the lower 16 bit segment IP is used for 16-bit memory address.
Flag Register: The flag register is a 32-bit register, however 14-bits are being used at present for 13 different tasks; these flags are upward compatible with those of the 8086 and 80286. The comparison of the available flags in 16-bit and 32-bit microprocessor is may provide some clues related to capabilities of these processors. The 8086 has 9 flags, the 80286 has 11 flags, and the 80286 has 13 flags. All of these flag registers include 6 flags related to data conditions (sign, zero, carry, auxiliary, carry , overflow, and parity) and three flags related to machine operations.(interrupts, Single-step and Strings). The 80286 has two additional : I/O Privilege and Nested Task. The I/O Privilege uses two bits in protected mode to determine which I/O instructions can be used, and the nested task is used to show a link between two tasks.
The processor also includes control registers and system address registers , debug and test registers for system and debugging operations.

c. Explain Dos function 01H (Keyboard Input with echo) and O2H (display a character on screen) of INT21 H with suitable example.
(Function 01H with example 4 MARKS , function 02H with example 4marks)
Answer

Function 1- Character input with echo:

Action: Reads a character from the standard input device and echoes it to the standard output device.
If no character is ready it waits until one is available. I/O can be re-directed, but prevents detection of OEF.

On entry: AH = 01h

Returns: AL = 8 bit data input

Notes: Equivalent to CP/M BDOS call 01h, except that if the character is CTRL-C an INT

23h is performed.

Example : mov ah, 01h ; load 01h in AH register
Int 21h ; call interrupt 21h
After the input is given by the keyboard , it's ASCII value is loaded in AL register.

Function 2 - Character output:

Action: Outputs a character to the standard output device. I/O can be re-directed, but prevents detection of 'disc full'.

On entry: AH = 02h
DL = 8 bit data (usually ASCII character)

Returns: Nothing

Notes:

Action: Sends a Character to the current listing device.

On entry: AH = 05h
DL = 8 bit data

Returns: Nothing

Notes: If the printer is busy this call will wait until the data is sent. There is no way to poll the printer status in DOS.

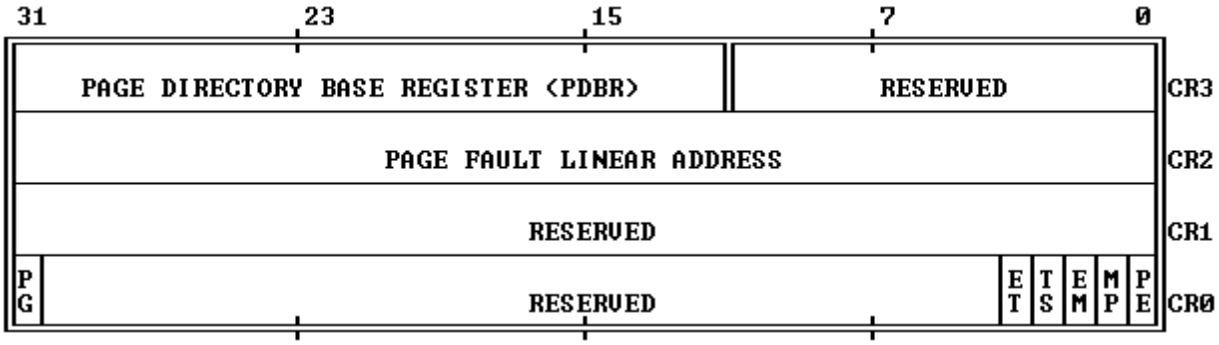
Example : mov ah, 02h ; load 02h in ah register
mov dl, '*' ; load the character to be displayed in DL register
int 21h ; call int 21h

Q3. Attempt any FOUR of the following **16M**

a. Draw and explain the format of CRO register of 80386.
(Diagram 2Marks, Description 2Marks)

Answer
Control Registers: The 80386 has three 32 bit control registers CR0, CR2 and CR3 to hold global machine status independent of the executed task.
CR0 contains system control flags, which control or indicate conditions that apply to the system as a whole, not to an individual task.
EM (Emulation, bit 2): EM indicates whether coprocessor functions are to be emulated.
ET (Extension Type, bit 4): ET indicates the type of coprocessor present in the system
MP (Math Present, bit 1): MP controls the function of the WAIT instruction, which is used to coordinate a coprocessor.

PE (Protection Enable, bit 0): Setting PE causes the processor to begin executing in protected mode. Resetting PE returns to real-address mode.
PG (Paging, bit 31): PG indicates whether the processor uses page tables to translate linear addresses into physical addresses.
TS (Task Switched, bit 3): The processor sets TS with every task switch and tests TS when interpreting coprocessor instructions.
CR2 is used for handling page faults when PG is set. The processor stores in CR2 the linear address that triggers the fault.
CR3 is used when PG is set. CR3 enables the processor to locate the page table directory for the current task.

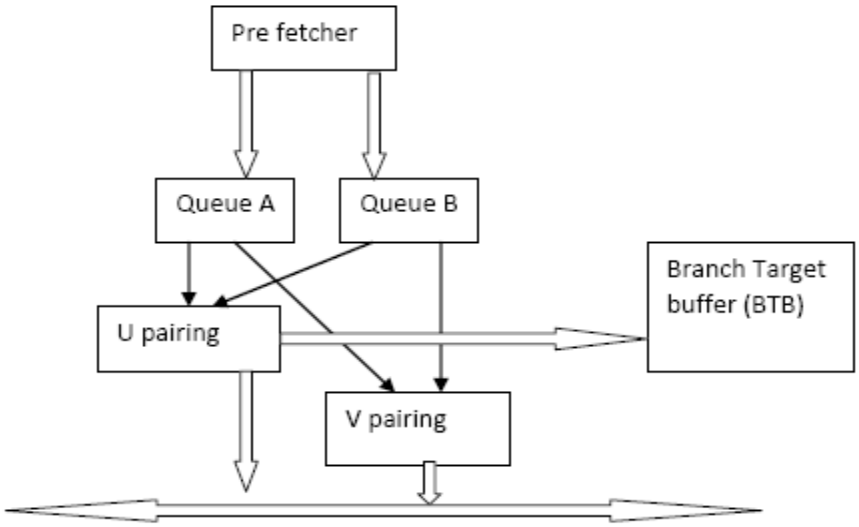


b. Explain branch prediction in Pentium Processor.
[3 marks Description and 1 Marks for Diagram]

Answer
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 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

Branch Prediction Logic:-



c. Explain any four floating point exception in Pentium processor
(Any four floating point exceptions -1 marks each)

Answer
The Pentium provides six floating point exceptions
1. Invalid operation (#I)
   - Stack overflow or underflow (#IS)
   - Invalid arithmetic operation (#IA)
2. Divide-by-zero (#Z).
3. Demoralized operand (#D)
4. Numeric overflow (#O)
5. Numeric underflow (#U)
6. Inexact result (precision)(#P).

Each of the six exception classes have a corresponding flag bit in the FPU status word and a mask bit in the FPU control word.

Denormal: The denormal exception occurs if one or more of the operands is a denormal number. This exception is never regarded as an error.

Divide-by-Zero Exception: A divide-by-zero exception occurs for a floating-point division operation if the divisor is zero and the dividend is finite and non-zero. It also occurs for other operations in which the operands are finite and the correct answer is infinite.
When the divide by zero exception is masked, the result is +/-infinity. The following specific cases cause a zero-divide exception:

- LOG(0.0)
- LOG10(0.0)

Overflow Exception: An overflow exception occurs if the rounded result of a floating-point operation contains an exponent larger than the numeric processing unit can represent. A calculation with an infinite input number is not sufficient to cause an exception.
When the overflow exception is masked, the calculated result is +/-infinity or the +/- largest represent able normal number depending on rounding mode. When the exception is not masked, a result with an accurate significant and a wrapped exponent is available to an exception handler.
Underflow Exception: The underflow exception occurs if the rounded result has an exponent that is too small to be represented using the floating-point format of the result.
If the underflow exception is masked, the result is represented by the smallest normal number, a denormal number, or zero. When the exception is not masked, a result with an accurate significant and a wrapped exponent is available to an exception handler
Inexact Exception: The inexact exception occurs if the rounded result of an operation is not equal to the unrounded result.
It is important that the inexact exception remain masked at all times because many of the numeric library procedures return with an undefined inexact exception flag. If the inexact exception is masked, no special action is performed. When this exception is not masked, the rounded result is available to an exception handler.

d. Describe hybrid architecture of microprocessor in brief.
(Explanation of hybrid architecture: 4Marks, any relevant points other than this shall be considered)

Answer

1. CISC processors are based on hybrid ISC-RISC architecture. Such hybrid architecture uses a decoder to convert CISC instructions into RISC instructions before execution.
2. These are then processes by a RISC core which performs a few basic instructions very quickly. Also RISC core allows performance enhancing features such as branch prediction and pipelining.
3. These have only been possible in RISC designs, since fixed length instructions are required for such features to work. Example Pentium and Athlon family of processor.

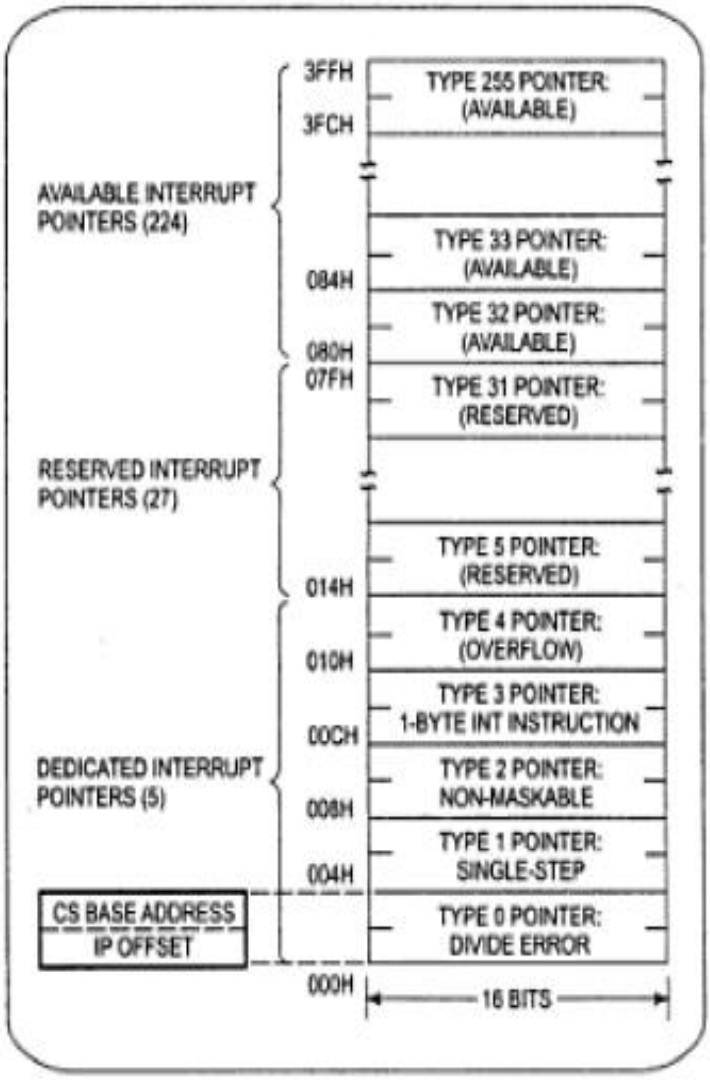
4. These processors are compatible with software developed for their CISC predecessors, yet they perform competitively against processors based on RISC design.
5. A CISC-RISC hybrid continues to consume a lot of power and is not best candidates for mobile and embedded applications.
9. Apart from having RISC core the number of general purpose registers in CISC processor has also grown and allow more instructions to be processes simultaneously.
6. 9.Intel Pentium III with SSE technology has an additional set of eight 128 bit vector registers for running SIMD (single instruction multiple data) instructions.
7. 10.The future successor to Pentium series Intel itanium IA-64 will even raise the bar further by implementing 128 general purpose registers.
8. 11.Many modern RISC processor support more instructions than old CISC designs. Example Motorola G4 processor used in power Macs and eMacs

e. Describe Interrupt Vector Table(IVT) of X86 processor with suitable diagram
(Diagram 2 marks, Description 2 marks)

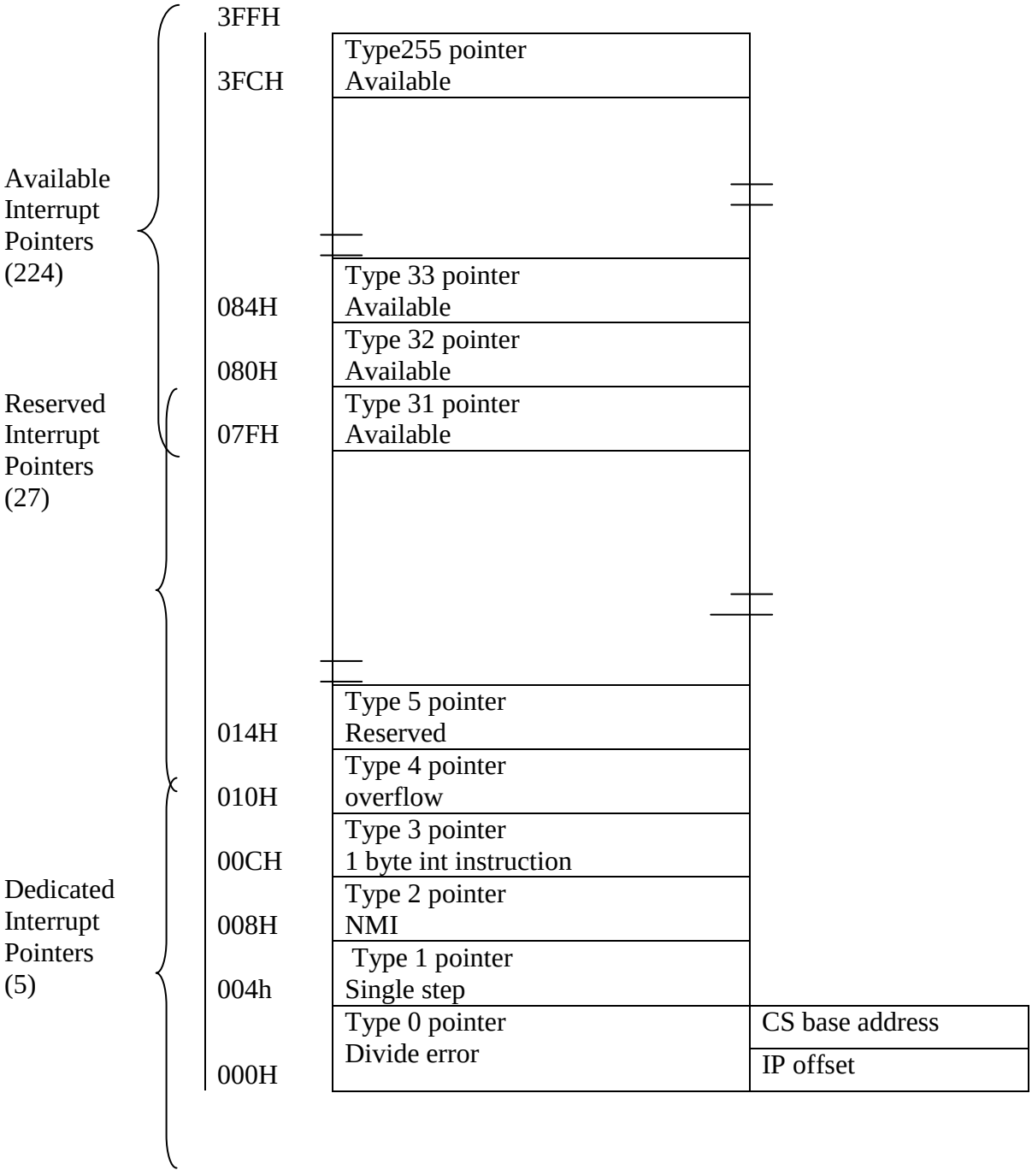
Answer
Figure shows the 256 interrupt vectors are arranged in the table in memory. Note that the instruction pointer value is put in as the low word of the vector, and the code segment register is put in as the high word of the vector. Each double word interrupt vector is identified by number from 0 to 255. Intel calls this number the type of interrupt.

- The lowest five types are dedicated to specific interrupts, such as the divide – by – zero interrupt, the single step interrupt, and the non maskable interrupt.
- Interrupts types 5 to 31 are reserved by intel for using more complex microprocessor, such as the 80286, 80386, and 80486.
- The upper 224 interrupts types, from 32 to 255, are available for use of hardware and software interrupts

In the figure the vector for each interrupt types requires four memory location. Therefore, when the 8086 represent to a particular type interrupt, it automatically multiplies the type by 4 to produce the desired address in vector table. It then goes to the address in the table to get the starting address of the interrupt – service procedure.


3FFH
TYPE 255 POINTER:
(AVAILABLE)
3FCH
AVAILABLE INTERRUPT
POINTERS (224)
TYPE 33 POINTER:
(AVAILABLE)
084H
TYPE 32 POINTER:
(AVAILABLE)
080H
07FH
TYPE 31 POINTER:
(RESERVED)
RESERVED INTERRUPT
POINTERS (27)
TYPE 5 POINTER:
(RESERVED)
014H
TYPE 4 POINTER:
(OVERFLOW)
010H
TYPE 3 POINTER:
1-BYTE INT INSTRUCTION
00CH
DEDICATED INTERRUPT
POINTERS (5)
TYPE 2 POINTER:
NON-MASKABLE
008H
TYPE 1 POINTER:
SINGLE-STEP
004H
CS BASE ADDRESS
IP OFFSET
TYPE 0 POINTER:
DIVIDE ERROR
000H
16 BITS

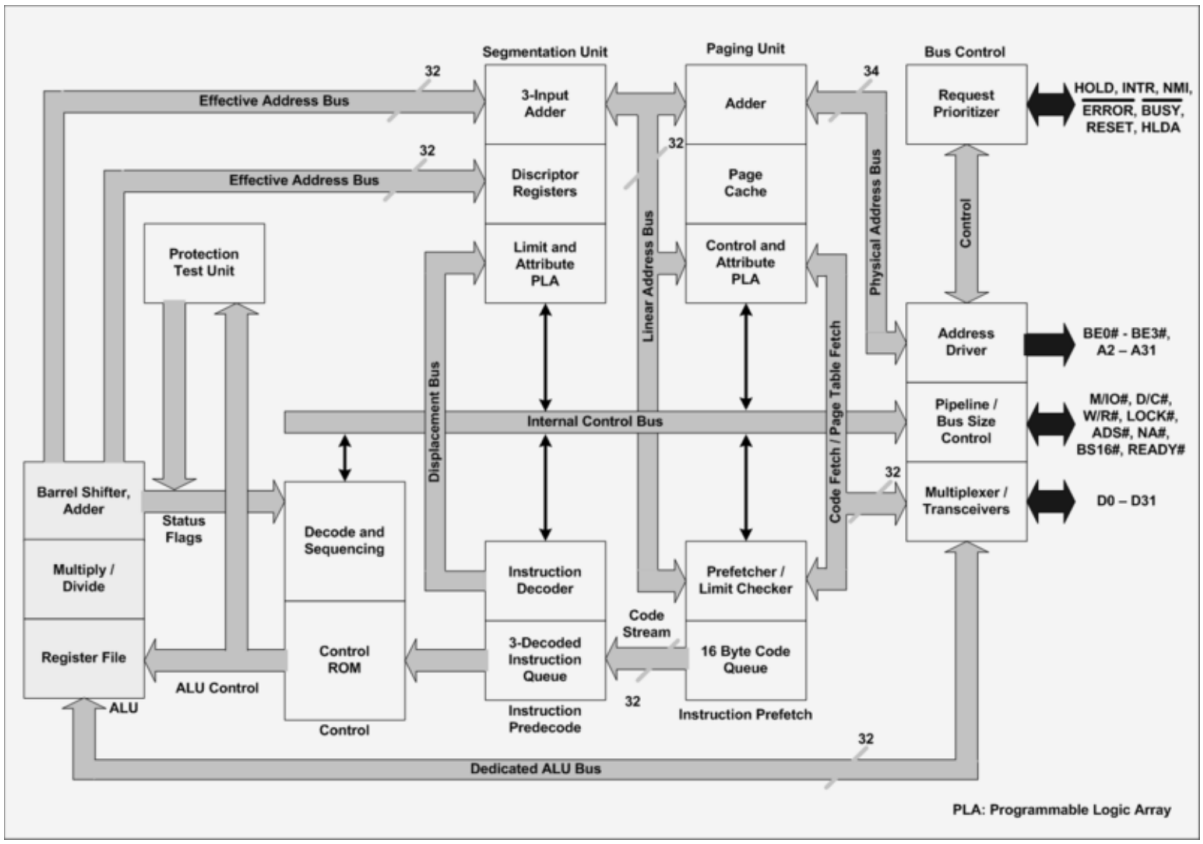
OR


3FFH
Type255 pointer
Available
3FCH
Available Interrupt Pointers (224)
Type 33 pointer
Available
084H
Type 32 pointer
Available
080H
Reserved Interrupt Pointers (27)
Type 31 pointer
Available
07FH
Type 5 pointer
Reserved
014H
Type 4 pointer
overflow
010H
Type 3 pointer
1 byte int instruction
00CH
Dedicated Interrupt Pointers (5)
Type 2 pointer
NMI
008H
Type 1 pointer
Single step
004h
Type 0 pointer
Divide error
CS base address
IP offset
000H

Q4 .Attempt any TWO of the following: 16M

a. Draw the architecture of 80386 and explain any two units in detail.
(Diagram 4 marks, any two units of 80386 2 marks each)

Answer



The internal architecture of 80386 can be divided into 3 sections such as
1. Central processing unit (CPU)
2. Memory management unit(MMU)
3. Bus interface unit(BIU)

The Central processing unit consists of: **Execution unit & Instruction unit**
Instruction unit has Instruction pre-fetcher and instruction pre-decode unit
The Instruction pre-fetcher fetches the 16 instruction bytes ahead of time and stores them into the 16 byte instruction pre-fetch queue(16 byte code).This speeds up the program execution process.
The instruction pre-decode unit has the instruction decoder and 3 decoded instruction queue.

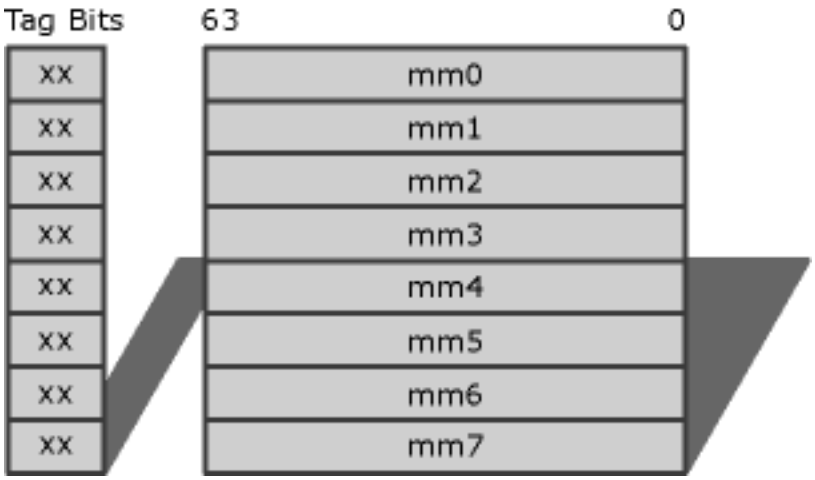
The instruction decoder decodes 3 instructions ahead of time and stores them in the 3 decoded instruction queue.
Execution unit has ALU and control unit.
The control unit stores the control signals in the control ROM, which are generated at the time of decoding .The decode and sequencing unit decodes 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.

The memory management unit has **segmentation unit** and **paging unit.**
The segmentation unit allows the use of two address components such as segment base address and offset address to calculate the 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
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.
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 data buffers (mux / transceivers) interface the internal data bus with the system data bus.

b. Describe Intel MMX architecture with register set and new data types.
(MMX registers set explanation : 3 marks ,Diagram 2 marks, description of data types 3 marks)

Answer
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.
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 single64 bit quantity.

c. Write interrupt processing sequence of X86 processors in detail.
(Description of sequence 6 marks, diagram 2 marks)

Answer

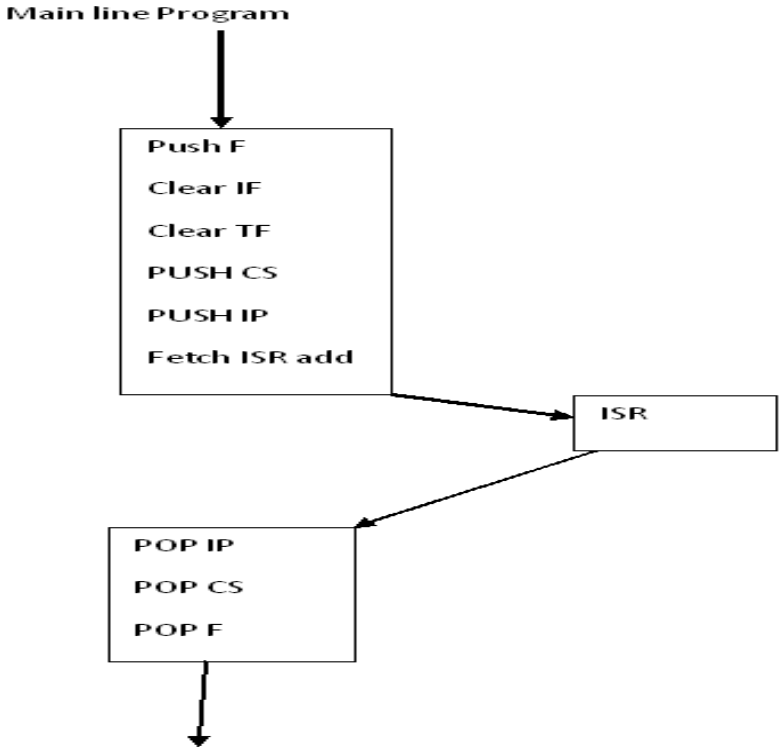
Interrupt processing sequence is as given below: When INT n instruction is executed:
1. The processor pushes flag register on stack then the contents of CS And IP register on stack
2. It clears two flags TF (trap flag) and IE (Interrupt enable flag).
3. Number of interrupt is used to find correct address 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.



Q5. **Attempt any FOUR of the following** **16M**

a. Describe the segment descriptor cache register with suitable diagram, in 80386 microprocessor.
(Diagram of segment descriptor cache register of 80386, 2 marks and description 2 marks)

Answer
Segment descriptor cache registers:

- 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.
- 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.

Segment registers ←16 BITS→		Descriptors registers (loaded automatically) Physical base address and segment limit (64 bits)		Segment attributes (9 bits)
Selector	CS			
Selector	SS			
Selector	DS			
Selector	ES			
Selector	FS			
Selector	GS			
Visible to users		**←32 bits base address→**	**←-----32 bits limit addr.→**	**Invisible to users**

b. List any eight special features of Pentium Pro-processor.
[Any 8 points, ½ Mark each]

Answer

1. The Pentium Pro processor has 36 address lines
2. The Pentium Pro processor has an additional 256/512 KB L2 cache memory on chip.
3. On chip L2 cache speeds processing and reduces the number of components in a system.
4. The L2 cache is connected to BIU, BIU generates memory addresses and control signals and passes or fetches data or instructions either to L1 data cache or L1 instruction cache.
5. The Instruction Fetch and Decode Unit (IFDU), contains three separate: instruction decoders that decode three instructions simultaneously
6. It also includes Branch Prediction Logic.
7. It predicts if the branch will be taken or not for a conditional jump instruction.
8. The instruction are then put into the instruction pool.
9. The instruction pool is a memory accessible with its content.
10. The execute unit consists of three units namely two integer execution unit and one floating point unit , two integer and one floating instruction can be executed simultaneously
11. Pentium Pro also has one jump execution unit (address generation unit).
12. The scheduling is performed by reservation station (RS) which can schedule up to five events for execution and process four simultaneously

c. Explain any two design issues of RISC processor.
(Any two design issues: 2 marks each)

Answer

- **Register Window:**
  1. The reduced hardware requirements of RISC processors leave additional space available on the chip for the system designer. RISC CPUs generally use this space to include a large number of registers (> 100 occasionally).
  2. 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.
  3. 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.
  4. Most RISC CPUs have some global registers which are always accessible. The remaining registers are windowed so that only a subset of the registers are accessible at any specific time.
  5. To understand how register windows work, we consider the windowing scheme used by the Sun SPARC processor.
  6. 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.

7. Of these 32 registers, 8 are global registers that are always accessible. The remaining 24 registers are contained in the register window.
8. The register window overlap. The overlap consists of 8 registers in SPARC CPU. Notice that the organization of the windows are supposed to be circular and not linear; meaning that the last window overlaps with the first window.
9. 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.

- **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.
- **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.

- **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.

d. Distinguish between .com and .exe program in DOS. (Any four points)
(Any 4 differences between .com and .exe programs. 1 marks each)

Answer

Sr. No	.COM programs	.EXE Programs
1	.COM file does not contain any header	.EXE file contains header
2	.COM file cannot contain relocation items.	.EXE file may contain relocation items.
3	Maximum size is 64k minus 256 bytes. For PSP and 2 bytes for stack.	No limit on size; Can be of any size
4	Entry point is PSP:0100	Entry point is defined by END directive.
5	Stack size is 64K minus 256 bytes for PSP and size of executable data and code.	Stack size is defined in a program with STACK directive.
6	Size of file is exact size of program.	Size of file is size of program plus header (Multiple of 256 bytes)

e. Compare 80386 processor with Pentium processor (any four points)
(Any four points- 1 mark each)

Answer:

Sr. No	80386	Pentium
1	It is 32 bit processor with 32 bit data and address bus	It is 32 bit processor with 32 bit address bus and 64 bit data bus
2	It does not have superscalar and super pipelined architecture	It has superscalar super pipelined architecture
3	It uses co-processor 80387, for mathematical calculations	It has on-chip floating point unit
4	It has on-chip address translation cache, but cache is not available	It has two separate on chip cache for code and data memory
5	It has single ALU	It has two ALUs
6	It does not support Branch Prediction	It supports branch prediction using prefetch buffers and branch target buffer

Q6. Attempt any TWO of the following **16M**

a. Describe the functions of the following pins of 80386 processor.
1. $\overline{BE_3}$
2. PEREQ
3. $\overline{LOCK}$
4. $\overline{BUSY}$
(2 marks for function of each pin)

Answer

1. $\overline{\mathbf{BE_3}}$ **(Bus/byte enable signal):** The 32-bit Data bus supported by 80386 and the memory system of 80386 can be viewed as a 4-byte wide memory access mechanism. The four byte enable lines, $\overline{BE_3} - \overline{BE_0}$, may be used for enabling these four banks. Using $\overline{\boldsymbol{BE_3}}$ enable signal line, the CPU may transfer 4 bytes of data simultaneously.

Byte Enable Signal	**Data bus Signal**
BE_0	D_0-D_7 (Byte 0- least significant)
BE_1	D_8- D_{15} (Byte 1)
BE_2	D_{16}- D_{23} (Byte 2)
BE_3	D_{24}- D_{31} (Byte 3- Most Significant)

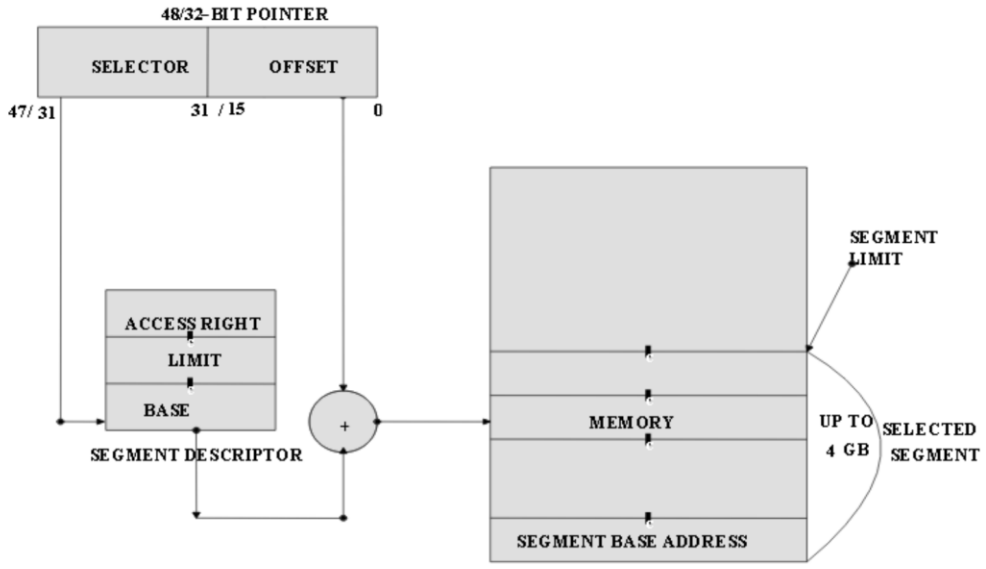
2. PEREQ: The processor extension request output signal indicates to the CPU to fetch a data word for the processor.
3. LOCK: The $\overline{\text{LOCK}}$ output pin enables the CPU to prevent the other bus masters (like coprocessor) from gaining the control of the system bus.
4. $\overline{\textbf{BUSY}}$**:** It signals a busy condition from a processor extension. When asserted this input indicates the coprocessor is still executing an instruction and is not yet able to accept another.

b. Describe address generation in PVAM mode of 80386 with suitable diagram.
(Diagram 4 Marks, Explanation 4 marks)

Answer

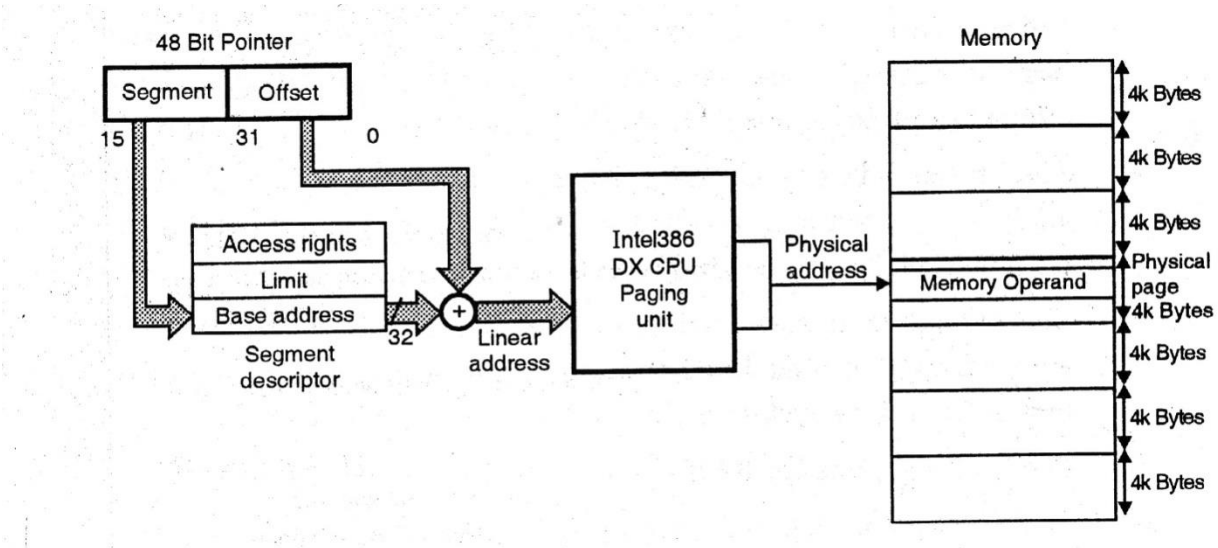
Address calculation in protected mode: Address generation in PVAM: In PVAM there are two components. A 16-bit selector which determines the linear base address of a segment and the base address is added to a 32-bit effective address to form a 32 bit linear address. The linear address is used as the 32-bit physical address or if paging is enabled the paging mechanism maps

the 32-bit linear address into a 32 physical address. The selector is used to specify an index into an OS defined table that contains the 32-bit base address of given segment. The physical address is formed by adding the base address obtained from the table to the offset. Paging provides additional memory management that operates only in PVAM. It provides a mean of managing large segments of memory. The paging mechanism translates the protected linear address from segmentation unit into a physical address.



Protected Mode Addressing Without Paging Unit

OR



c. Explain any two DOS functions of INT21H for file processing with suitable example.
(Description of any two DOS function with example : 4 marks each)

Answer

1. 3CH : to create file

This function creates a file with indicated attributes and opens the 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
Example: 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
Example: 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
Example: 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
CX = number of bytes to read
DS:DX -> buffer for data
Example: 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

Example: 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.
Example: 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
Example: 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
Example: 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
Example: mov ah, 57h; set/get file date and time int 21h; transfer to DOS