

The 80186 CPU is a high performance 8 MHz processor and can address one mega byte (1 Mb) of physical memory. It is more powerful than 8086 and completely software compatible with 8086. In 80186, most of the 8086 instructions execute in a fewer clock cycles, because of the hardware enhancement in the execution unit and the bus interface unit. The 80186 bus interface unit and execution unit operate in a similar manner to that of the 8086 processor units with advanced hardware capability. The integration of various peripherals inside the processor chip decreases the complexity of the subsystem designed using 80186 and hence it can be termed as a microcontroller.

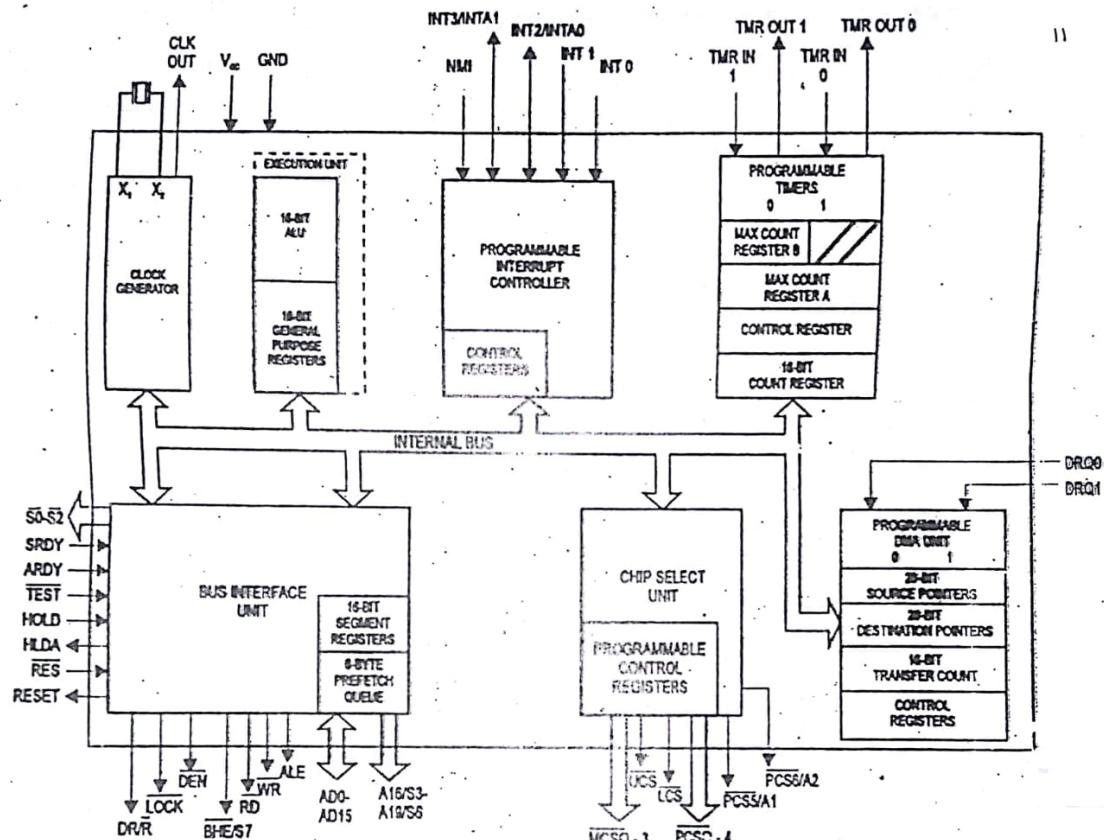


Figure 9.2 Intel 80186 CPU Internal block diagram (Intel Corp.)

Software Enhancements: The 80186 microprocessor has 16 additional instructions as compared to the 8086 microprocessor. They are,

ENTER	Enter a Procedure
LEAVE	Leave a Procedure
BOUND	Check for an array index validity
PUSHA	Push all registers onto the stack
POPA	Pop all registers from the stack
PUSH immediate	Push immediate numeric value onto the stack
INS	Input string byte or word
OUTS	Output string byte or word
IMUL dest, source, immediate	dest ← source * immediate

These instructions simplify assembly programming and facilitate the implementation of high level programming with improved performance.

9.3 INTEL 80286 PROCESSOR

The Intel 80286 is a 16-bit, high performance microprocessor which supports multitasking. It has 24 address lines and is capable of directly addressing 16 Mb (2^{24}) of physical memory. It has memory management capability with four levels of memory protection. It supports implementation of virtual memory and operating system. The assignment of privilege to the software module allows the processor to enforce the hierarchical inter-relationship among the software modules at run time. The code segments which are more privileged than those of the operating system tasks, takes full control of the 80286 hardware. It prevents the application programs from performing I/O, responding to interrupts, and calling the system services directly except through pre-established entry points. The 8086, 80186, and 80286 CPU family contains same basic set of registers, instructions, and the addressing modes.

The overall organisation of the 80286 microprocessor is shown in Figure 9.3.

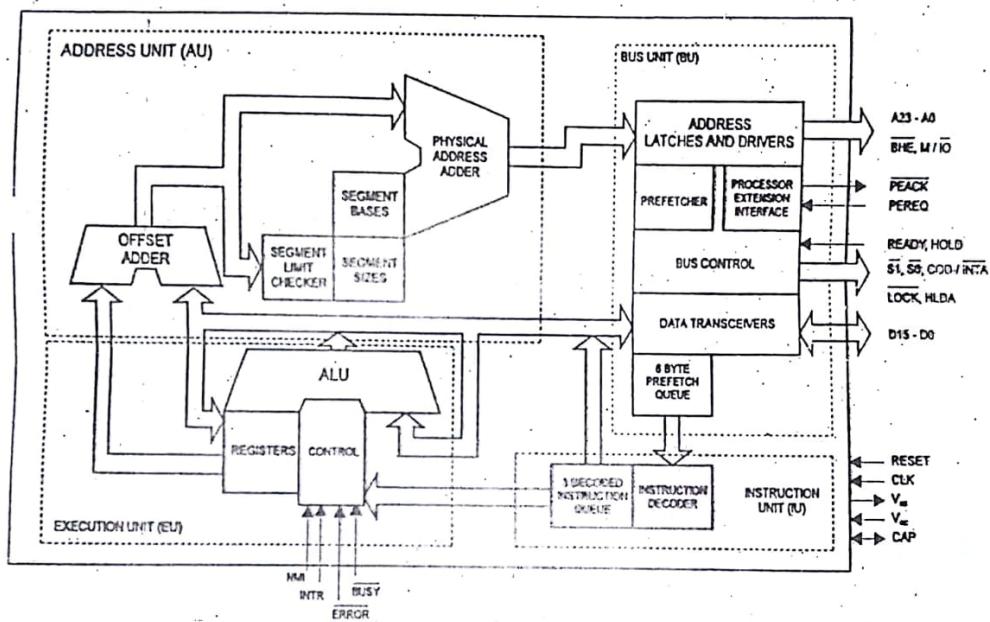


Figure 9.3 Intel 80286 CPU Internal block diagram (Intel Corp.)

Functional Units: The 80286 processor consists of four functional units. They are,

- Bus Interface Unit (BIU)
- Instruction Unit (IU)
- Execution Unit (EU)
- Address Unit (AU)

All the four units work in parallel within the CPU.

Bus Interface Unit (BIU): The memory or peripheral device Read/Write operations are performed by the bus unit. The BIU prefetches the instructions and places them in a 6-byte instruction buffer while executing the current instruction.

Instruction Unit (IU): The instruction unit decodes the instructions prefetched by the BIU and maintains a queue of three decoded instructions for execution.

Execution Unit (EU): The execution unit executes the instructions from the queue of three decoded instructions.

Operating Modes

The 80386 CPU can operate in three modes. They are,

- Real Mode : 8086 compatible mode
- Protected Mode : 80286 compatible protected mode
- Virtual Mode : 8086 environment created within protected mode

The processor executes 32-bit code segment in the protected mode and delivers full 32-bit addressing. The three operating modes are shown in Figure 9.5.

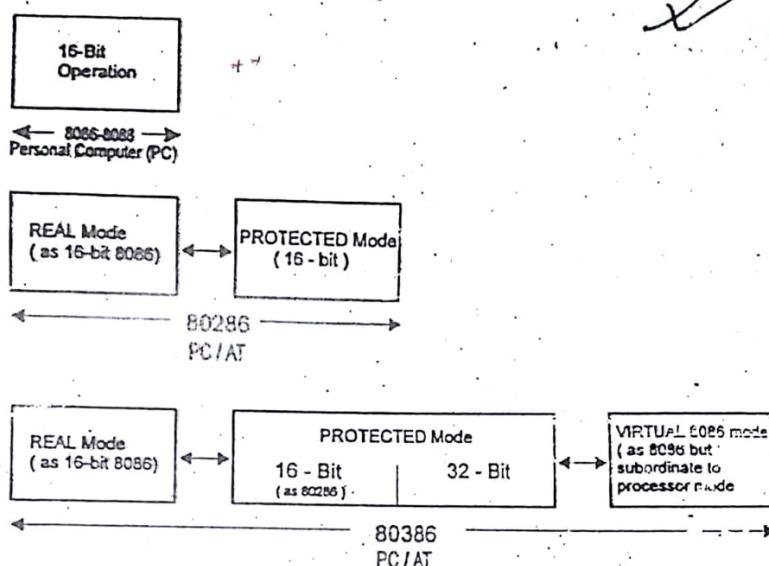


Figure 9.5 8086, 80286, and 80386 CPU Operating Mode Development

The 80386 processor always starts execution in the REAL mode which runs in the 80386 compatible mode. The 8086 microprocessor is a 16-bit processor supporting only REAL mode and can address 1 Mb (2^{20}) of physical memory. The 80386's 8086 compatible mode allows the 8086 software to run without modification at the speed of 80386 CPU.

The 80286 CPU is an advanced 16-bit processor, which operates in both the REAL and the PROTECTED mode and can address 16 Mb of physical memory. The addressing capability of 80286 is sixteen times greater as compared to the 8086 processor. The computer system built with 8086/8038 is called as Personal Computer (PC). The PCs built with 80286 or higher version processors are called as PC-AT. The 80386's 32-bit operation in the PROTECTED mode can address 4 Gb of physical memory. The 80386 supports one more operating mode known as subordinate VIRTUAL mode. It is fully compatible with 8086 processor and links the existing software.

The complete operation of the 32-bit hardware is provided in the PROTECTED mode. In the PROTECTED mode (either in 16-bit or 32-bit), the 80386 can enable its paging unit for full support of the VIRTUAL memory. It relieves the programmer from the limitation of addressing of the physical memory. Simultaneously, the protection hardware is activated, hence the name, PROTECTED mode. The on-chip protection ensures the greater system reliability as indicated by the policies enforced by the protection hardware. The protection features ensure that the integrity of operating system is maintained from the interference of the user programs. It also protects the user from other users. The VIRTUAL 8086 mode runs the existing software under the control of a 32-bit PROTECTED-mode operating system.

Address Unit (AU): The address unit is responsible for computation of the address of the memory or I/O devices which is sent by the bus interface unit for read and write operation.

Operating Modes

The 80286 processor operates in two modes: Real mode and Protected Virtual mode. Initially, the CPU will be in real mode. It can be switched to the protected mode by setting the protection enable (PE) bit of a new system register, in the machine status word.

Real Mode: In real mode, the CPU of 80286 operates as high performance 8086 compatible processor. It emulates 8086 as a high performance CPU and will be able to address 1 Mb of physical memory.

Protected Virtual Mode: In virtual address mode, the memory management unit of 80286 can address the physical memory of 16 Mb directly and can support 1 Gb (Giga Byte) of virtual memory. The virtual memory segments are swapped in from/to disk storage with to/from 16 Mb of physical memory. Protected virtual mode is useful for implementing the multiuser and multitasking system. The CPU supports a four-level memory protection such that the user programs cannot modify or interfere with the other user's data and code. It can protect the operating system from the user programs.

9.4 INTEL 80386 PROCESSOR

The Intel 80386 is the first 32-bit CPU in x86 family of processors. It has 32-bit data bus and 32-bit nonmultiplexed address bus. It is capable of directly addressing 4 gigabyte (2^{32}) of physical memory and supports 64 terabyte (2^{45}) of virtual memory. The overall organisation of 80386 microprocessor is shown in Figure 9.4.

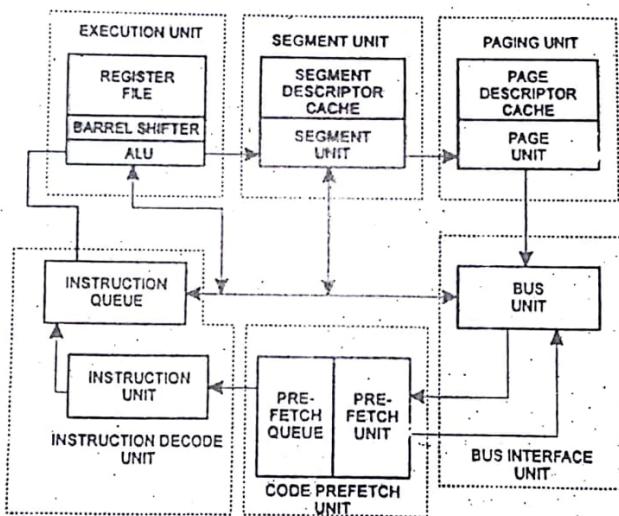


Figure 9.4 Intel 80386 CPU Internal block diagram (Intel Corp.)

The 80386 consists of six functional units. They are,

- Bus Interface Unit
- Execution Unit
- Segment Unit
- Paging Unit
- Instruction Decode Unit
- Code Prefetch Unit

Programmable Hardware Registers

The 80386 CPU has four 32-bit general purpose registers, two 32-bit index registers, two 32-bit pointer registers, six 16-bit segment registers, a 32-bit instruction pointer, a 32-bit flag register, six debug registers, and a 32-bit status register. The 32-bit registers of 80386 are designated by prefixing the letter 'E' to the 8086 registers (Eg. AX as EAX, etc.). The 16-bit 8086 compatible registers use only the first 16-bits of 80386's 32-bit registers. Only 16-bits of the 32-bit registers are used in 16-bit operation while running in the REAL or VIRTUAL mode.

Commercial Versions

The 80386 CPU is commercially available in two versions, the 80386 SX and the 80386 DX. The SX and DX versions have 32-bit internal architecture, however, they differ in bit width of the data bus.

80386 DX Version: The DX version uses a 32-bit wide data bus whereas the SX uses 16-bit data wide bus. The DX version is more powerful than the SX version. It has 32-bit nonmultiplexed address lines and can address 4 Gb (2^{32}) of physical memory and 64 Tb (2^{46}) of virtual memory. It is encased in a 132-pin grid array package and is available in speeds of 25, 33, and 40 MHz.

80386 SX Version: The SX version is less powerful than the DX version. It can be interfaced with 16-bit I/O devices. It has 24 address lines and can address 16 Mb (2^{24}) of physical memory and 64 Tb (2^{44}) of virtual memory. It is encased in a 100-pin plastic quad flatpack. The computers built with SX version are cheaper than those built with DX version.

9.5 INTEL 80486 PROCESSOR

The Intel 80486 is an advanced, evolutionary high performance, 32-bit CMOS, CISC microprocessor. The overall organisation of 80486 microprocessor is shown in Figure 9.6.

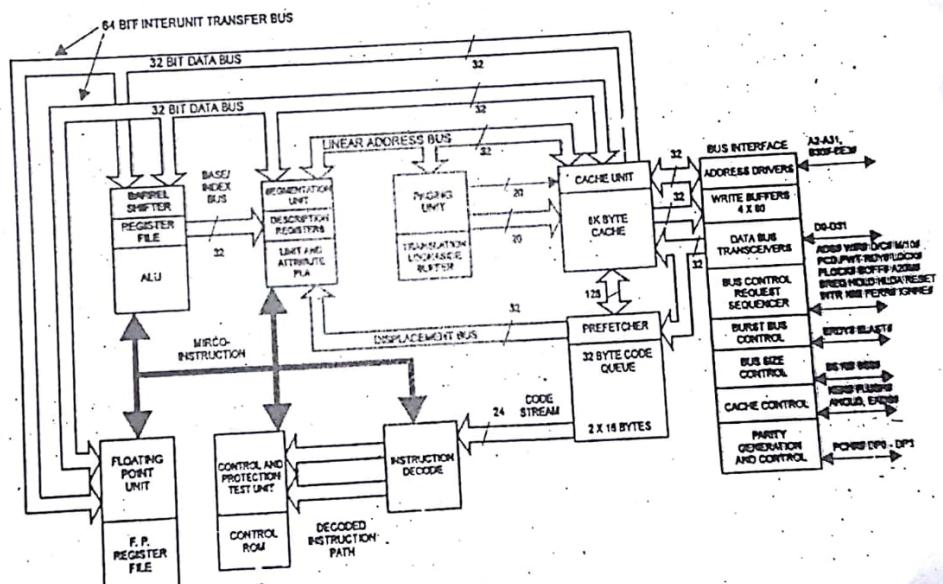


Figure 9.6 Intel 80486 Processor block diagram (Intel Corp.)

* Scalar processor

* The complement one instruction in one clock cycle replacing Pipeling

++ Repte Pipeling

++ Cache memory => 8 kB

++ Math Co-processor / unit (FP unit X FP memory)

80486 DX Version: The DX version uses 32-bit wide data bus, whereas the SX uses a 16-bit wide data bus. The DX version is more powerful than the SX version. It is available in the speeds of 33, 50, 66 and 90 MHz. It has a built-in floating point coprocessor.

80486 SX Version: The SX version is less powerful than the DX version. It can be interfaced with the 16-bit I/O devices. It is available in speeds of 20, 25, and 33 MHz. The SX version does not have built-in numeric coprocessor.

9.6 INTEL PENTIUM PROCESSOR

The Pentium is a 32-bit, superscalar architecture processor with a 64-bit bus, but at some stages inside the chip it has a 256-bit bus and has the ability to run almost two instructions per clock cycle. It has some critical and frequently used instructions hardwired rather than micro-coded which accounts for the CISC-RISC design. It employs superscalar integer pipelines, branch prediction, and highly pipelined floating point unit to achieve the highest x86 performance level while preserving the binary code compatibility with x86 architecture though it was the first one to break the x86 design philosophy (CISC).

Organisation

The overall organisation of the Pentium microprocessor is shown in Figure 9.7. It has a core execution unit, two integer pipelines, a floating point pipeline with a dedicated adder, multiplier, and divider. The separate on-chip instruction code and data caches meet the memory demands of the execution units, with a branch target buffer augmenting the instruction cache for dynamic branch prediction. The external interface includes separate addresses and a 64-bit data bus.

Integer Pipeline

The integer pipeline of the Pentium processor is simple as compared to the 80486 central processing unit. The pipeline has five stages (prefetch, first decode, second decode, execute, and write back) with the following functions.

Prefetch (PF): The prefetch stage prefetches (reading in advance) code from the instruction cache and aligns the code.

First Decode (D1): In this stage, the CPU decodes the instruction to generate a control word. A single control word executes directly, while the complex instruction requires microcoded control sequence.

Second Decode (D2): The CPU decodes the control word from D1 stage for the use in the execute stage. In addition, the CPU generates address for data reference in the memory.

Execute (E): The CPU either accesses the data cache or computes the result in the arithmetic and logic unit (ALU) in this stage.

Write back (WB): In the last stage, the CPU updates the registers and flags with the instruction results. All exception (triggered by some failure, e.g. divide by zero) conditions must be resolved before an instruction can advance to WB stage.

Comparing with the integer pipeline of the 80486 CPU, the Pentium processor integrates the additional hardware in several stages to speed up instruction execution. For example, the 80486 CPU requires two clock cycles to decode several instructions, but the Pentium takes one clock cycle and executes shift and multiply instruction faster. The Pentium processor substantially enhances the superscalar execution (executing more than one instruction at a time), branch prediction, and cache organisation.

Super-Scalar Execution: The Pentium CPU has a superscalar organisation that enables two instructions to execute in parallel. The ALU functions have been replicated in independent pipelines, called U and V (the pipeline names U and V are selected because U and V were the first successive letters of the alphabet and neither of them were the initials of a functional unit in design partitioning). In the prefetch (PF) and D1 stages described above, the CPU can fetch and decode two instructions in parallel and issue them to the U and V pipelines. The U and V have separate ALUs. Additionally, for complex instructions the CPU in D1 can generate microcode sequences that control both the U and V pipelines. When a jump instruction is issued to the U pipeline, the CPU in D1 never issues any instruction to the V-pipe, thereby eliminating control dependencies.

Problems of Pipeline

Resource Dependency: It occurs when two instructions require a signal from the functional unit or a data unit.

Data Dependency: It occurs when one instruction writes the result that is read or written by another instruction. Consider an example which illustrates this concept.

```
I1: MOV AX, BX ; ax ← bx  
I2: MOV DX, AX ; dx ← ax
```

The instructions I1 and I2 cannot be executed in parallel because the result generated by I1 which is in AX register is referred as a source operand in the instruction I2. Hence, I1 and I2 cannot be executed in parallel.

Control Dependency: It occurs when the result of one instruction determines whether another instruction has to be executed or not.

Example:

```
if( condition ) then  
I1: a ← 100  
else  
I2: a ← 200
```

The instructions I1 and I2 cannot be executed in parallel because they have to be executed based on the condition, as they are updating the same memory location.

Branch Prediction

This is a technique to predict the most likely set of instructions to be executed, and "prefetch" to make them available to the pipelines as and when they are referred. The execution pipelines are supplied with instructions at a rapid rate which improves the overall performance. The branch predictions are not always correct, but a good design ensures a hit ratio of 90 percent that can improve performance of the processor substantially.

The Pentium employs a Branch Target Buffer (BTB), which is an associative memory used to improve the performance if it takes the branch instruction.

When a branch instruction is first taken, the CPU makes an entry in the BTB to associate the branch instruction address with its destination address and to initialize the history used in the prediction algorithm. Using these information, the Pentium CPU executes correctly predicted branch without any delay.

Cache Organisation

The Pentium processor uses separate code and data caches as the superscalar design and branch prediction demand more bandwidth than a unified cache (both code and data in same cache). First, efficient branch prediction requires that the destination of a branch be accessed simultaneously with data reference of the previous instruction executing in the pipeline. Second, the parallel execution of data memory reference requires simultaneous access for loads and stores. Third, in the context of the overall Pentium microprocessor design, handling self-modifying code for separate code and data cache is only marginally more complex than for a unified cache.

Floating-Point Pipeline

The floating point unit supports common functions such as the computationally expensive divide function with hard-wired implementation, speeding up across the board. Pentium floating point pipeline consists of eight stages. They are Prefetch, First Decode, Second Decode, Operand fetch, First execute, Second execute, Write float, Error reporting. The eight stage pipeline in floating point unit (FPU) allows a single cycle throughput, for most of the basic floating point instructions such as floating add, subtract, multiply, and compare.

Register Stack Manipulation

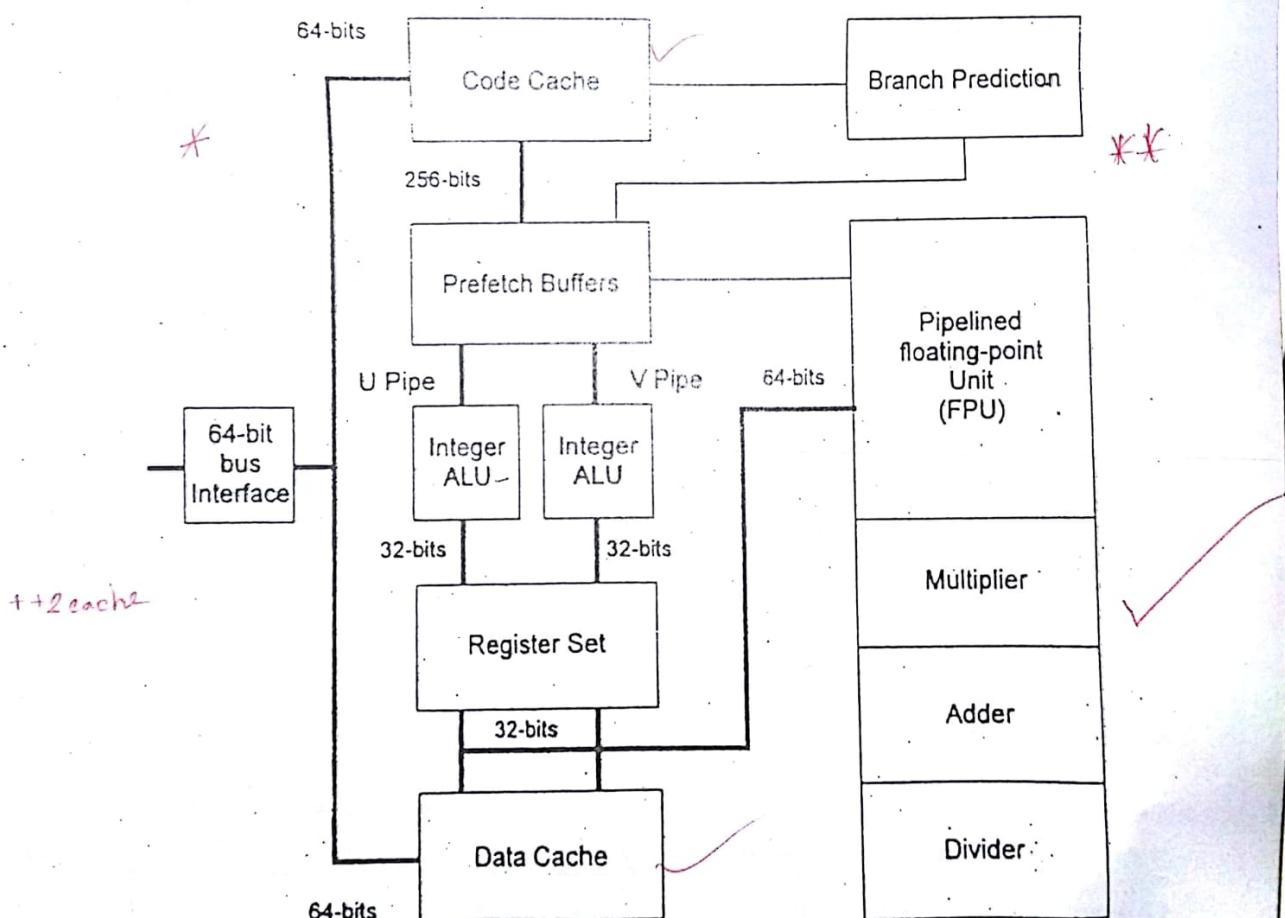
The x86 floating point instruction set uses the register file as a stack of eight registers in which the top of stack (TOS) acts as an accumulator of the results. To improve the floating point performance by optimizing the use of floating point register file, Pentium's FPU can execute the FXCH instruction in parallel with any of the basic floating point operation.

Instruction issue algorithm takes control dependency into consideration and issues the instruction as per the algorithm listed below.

Algorithm: Pentium Instruction Issue

1. Decode two consecutive instructions: I1 (current instruction), I2 (next instruction).
2. If the following conditions are true
 - a) I1 is a simple instruction
 - b) I2 is a simple instruction
 - c) Instruction I1 is not jump instruction
 - d) Destination of I1 \neq source of I2
 - e) Destination of I1 \neq Destination of I2
 then
 - Issue I1 to U-pipeline and I2 to V-pipeline
 - else
 - Issue I1 to U-pipeline
3. End

Several techniques are used to resolve dependencies between the instructions that might be executed in parallel. Various dependencies associated with the instruction are Resource, Data, Control, and Duplication.



Pentium is not only a Super Pipelined Processor.

Problems of Pipeline

Resource Dependency: It occurs when two instructions require a signal from the functional unit or a data unit.

Data Dependency: It occurs when one instruction writes the result that is read or written by another instruction. Consider an example which illustrates this concept.

```
I1: MOV AX, BX ; ax ← bx  
I2: MOV DX, AX ; dx ← ax
```

The instructions I1 and I2 cannot be executed in parallel because the result generated by I1 which is in AX register is referred as a source operand in the instruction I2. Hence, I1 and I2 cannot be executed in parallel.

Control Dependency: It occurs when the result of one instruction determines whether another instruction has to be executed or not.

Example:

```
if( condition ) then  
I1: a ← 100  
else  
I2: a ← 200
```

The instructions I1 and I2 cannot be executed in parallel because they have to be executed based on the condition, as they are updating the same memory location.

Branch Prediction

This is a technique to predict the most likely set of instructions to be executed, and "prefetch" to make them available to the pipelines as and when they are referred. The execution pipelines are supplied with instructions at a rapid rate which improves the overall performance. The branch predictions are not always correct, but a good design ensures a hit ratio of 90 percent that can improve performance of the processor substantially.

The Pentium employs a Branch Target Buffer (BTB), which is an associative memory used to improve the performance if it takes the branch instruction.

When a branch instruction is first taken, the CPU makes an entry in the BTB to associate the branch instruction address with its destination address and to initialize the history used in the prediction algorithm. Using these information, the Pentium CPU executes correctly predicted branch without any delay.

Cache Organisation

The Pentium processor uses separate code and data caches as the superscalar design and branch prediction demand more bandwidth than a unified cache (both code and data in same cache). First, efficient branch prediction requires that the destination of a branch be accessed simultaneously with data reference of the previous instruction executing in the pipeline. Second, the parallel execution of data memory reference requires simultaneous access for loads and stores. Third, in the context of the overall Pentium microprocessor design, handling of self-modifying code for separate code and data cache is only marginally more complex than for a unified cache.

Floating-Point Pipeline

The floating point unit supports common functions such as the computationally expensive divide function with hard-wired implementation, speeding up across the board. Pentium floating point pipeline consists of eight stages. They are Prefetch, First Decode, Second Decode, Operand fetch, First execute, Second execute, Write float, Error reporting. The eight stage pipeline in floating point unit (FPU) allows a single cycle throughput, for most of the basic floating point instructions such as floating add, subtract, multiply, and compare.

Register Stack Manipulation

The x86 floating point instruction set uses the register file as a stack of eight registers in which the top of stack (TOS) acts as an accumulator of the results. To improve the floating point performance by optimizing the use of floating point register file, Pentium's FPU can execute the FXCH instruction in parallel with any of the basic floating point operation.

- write instructions to transfer data under status check I/O and interrupt I/O.
- List operating modes of the 8254 timer and write instructions to set up the timer in the various modes.
- Explain the functions of the 8259A interrupt controller and its operation in the fully nested mode.
- Explain the process of the Direct Memory Access (DMA) and the functions of various elements of the 8237.

15.1 THE 8255A PROGRAMMABLE PERIPHERAL INTERFACE

The 8255A is a widely used, programmable, parallel I/O device. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It is flexible, versatile, and economical (when multiple I/O ports are required), but somewhat complex. It is an important general-purpose I/O device that can be used with almost any microprocessor.

The 8255A has 24 I/O pins that can be grouped primarily in two 8-bit parallel ports: A and B, with the remaining eight bits as port C. The eight bits of port C can be used as individual bits or be grouped in two 4-bit ports: C_{UPPER} (C_U) and C_{LOWER} (C_L), as in Figure 15.1(a). The functions of these ports are defined by writing a control word in the control register.

Figure 15.1(b) shows all the functions of the 8255A, classified according to two modes: the Bit Set/Reset (BSR) mode and the I/O mode. The BSR mode is used to set or reset the bits in port C. The I/O mode is further divided into three modes: Mode 0, Mode 1, and Mode 2. In Mode 0, all ports function as simple I/O ports. Mode 1 is a handshake mode whereby ports A and/or B use bits from port C as handshake signals. In the handshake mode, two types of I/O data transfer can be implemented: status check and interrupt. In Mode 2, port A can be set up for bidirectional data transfer using handshake signals from port C, and port B can be set up either in Mode 0 or Mode 1.

15.1.1 Block Diagram of the 8255A

The block diagram in Figure 15.2(a) shows two 8-bit ports (A and B), two 4-bit ports (C_U and C_L), the data bus buffer, and control logic. Figure 15.2(b) shows a simplified but expanded version of the internal structure, including a control register. This block diagram includes all the elements of a programmable device; port C performs functions similar to that of the status register in addition to providing handshake signals.

CONTROL LOGIC

The control section has six lines. Their functions and connections are as follows:

- RD (Read):** This control signal enables the Read operation. When the signal is low, the MPU reads data from a selected I/O port of the 8255A.
- WR (Write):** This control signal enables the Write operation. When the signal goes low, the MPU writes into a selected I/O port or the control register.

Pipelining: Save time in ratio
of 25:9

I.	PF	D ₁	D ₂	E	WB
I ₁	PF	D ₁	D ₂	E	WB
I ₂	PF	D ₁	D ₂	E	WB
I ₃	PF	D ₁	D ₂	I	WB
I ₄	PF	D ₁	D ₂	E	WB

Delayed branching:-

Q) What is Hardware Control?

2) What is Software Control

What are the advantages and disadvantages?