Intel 80386 MP architecture:

The 80386 is a high performance 32-bit microprocessor designed to drive the most advanced computer-based applications. The 80386 forms the basis for a high-performance 32-bit system. The 80386 incorporates multitasking support, memory management, pipelined architecture, address translation caches, and a high-speed bus interface all on one chip.

Pipelined architecture enables the 80386 to perform instruction fetching, decoding, execution, and memory management functions in parallel.

The internal architecture of the 80386 consists of six functional units that operate in parallel. Fetching, decoding, execution, memory management, and bus accesses for several instructions are performed simultaneously. This parallel operation is called pipelined instruction processing. With pipelining, each instruction is performed in stages, and the processing of several instructions at different stages may overlap as illustrated in Figure 1. The six-stage pipelined processing of the 80386 results in higher performance and an enhanced throughput rate over non-pipelined processors. The six functional units of the 80386 are identified as follows:

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

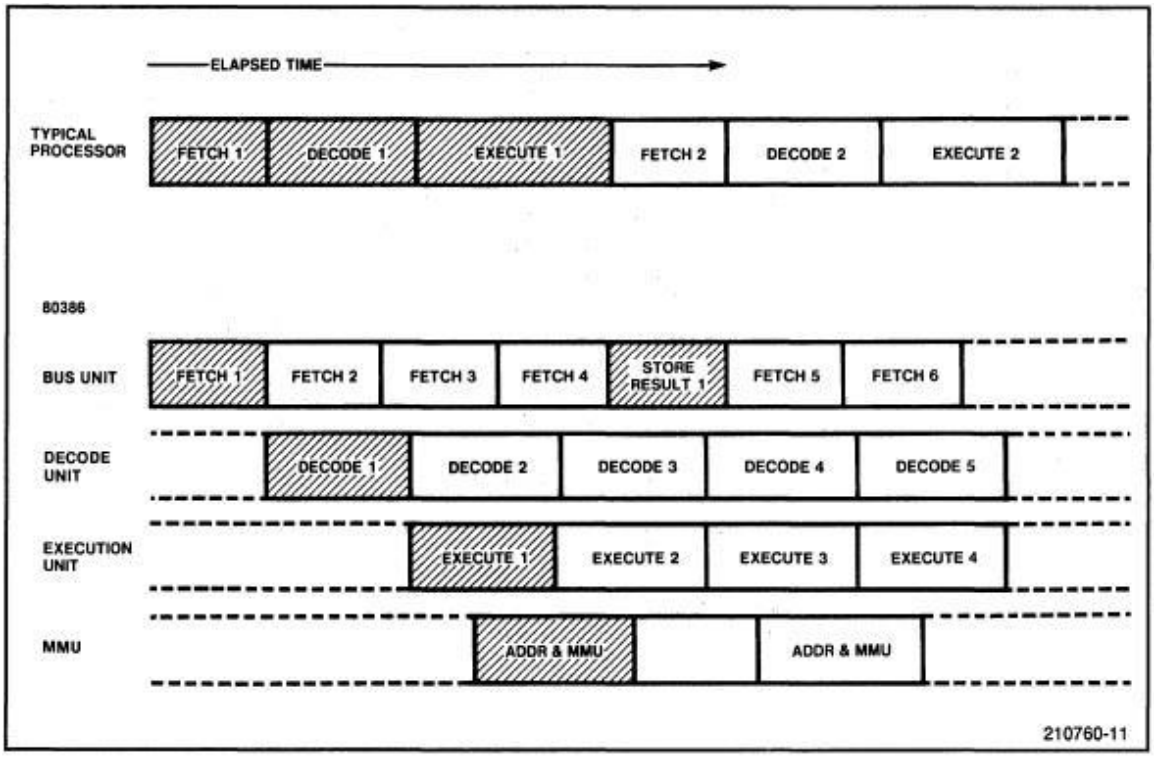


Figure (1): **Instruction Pipelining**

The Execution Unit in turn consists of three subunits:

• Control Unit

• Data Unit

• Protection Test Unit

Figure 2 shows the organization of these units.

1- BUS INTERFACE UNIT

The Bus Interface Unit provides the interface between the 80386 and its environment. It accepts internal requests for code fetches (from the Code Prefetch Unit) and data transfers (from the Execution Unit), and prioritizes the requests. At the same time, it generates or processes the signals to perform the current bus cycle. These signals include the address, data, and control outputs for accessing external memory and I/O. The Bus Interface Unit also controls the interface to external bus masters and coprocessors.

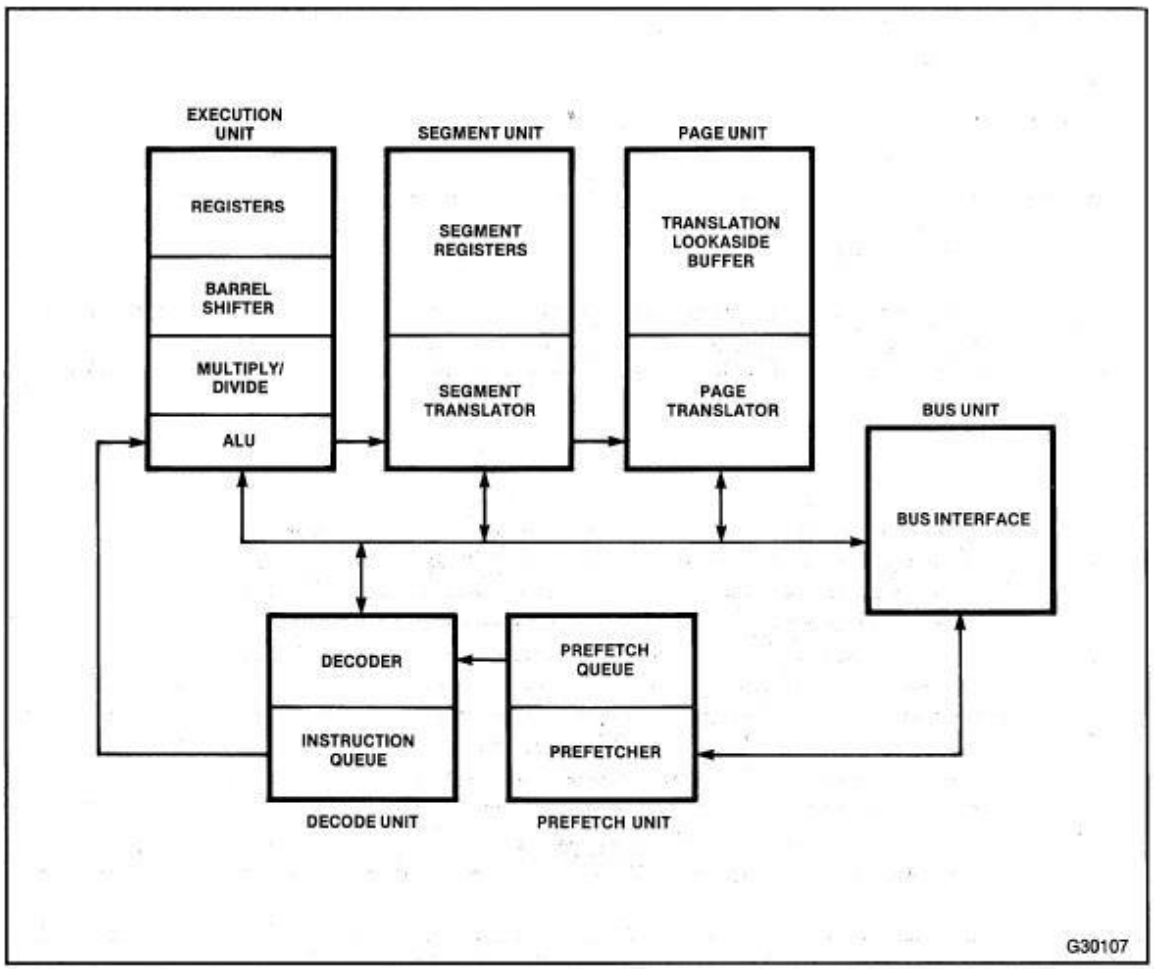


Figure (2): Intel 80386 Functional Units

2. Code Prefetch Unit

- The prefetch unit performs a mechanism known as an instruction stream queue.
- This queue permits a prefetch up to 16 bytes of instruction code.
- Whenever the queue is not full, and the execution unit is not asking the bus unit to read or write data from the memory, the prefetch queue supplies addresses to the bus interface unit and signals it to look ahead in the program by fetching the next sequential instructions.
- Prefetched instructions are held in the FIFO queue for use by the instruction decoder.

- Whenever bytes are loaded into the input end of the queue, they are automatically shifted up through the FIFO to the empty location near the output.

3. Instruction Decode Unit

- The decode unit accesses the output end of the prefetch unit's instruction queue.
- It reads the machine-code instructions from the output side of the prefetch queue and decodes them into microcode instruction format used by the execution unit, thus it off-loads the responsibility for the instruction decoding from the execution unit.
- The instruction queue, a part of the decode unit permits three fully decoded instructions to be held waiting for use by the execution unit.
- Thus it improves the performance of the CPU.

4. The Execution Unit

The Execution Unit executes the instructions from the Instruction Queue and therefore communicates with all other units required to complete the instruction. The functions of its three subunits are as follows:

- The Control Unit contains microcode and special parallel hardware that speeds multiply, divide, and effective address calculation.
- The Data Unit contains the ALU, a file of eight 32-bit general-purpose registers, and a 64-bit barrel shifter (which performs multiple bit shifts in one clock). The Data Unit performs data operations requested by the Control Unit.
- The Protection Test Unit checks for segmentation violations under the control of the microcode. To speed up the execution of memory reference instructions, the Execution Unit partially overlaps the execution of any memory reference instruction with the previous instruction. Because memory reference instructions are frequent, a performance gain of approximately nine percent is achieved.

5. Segmentation Unit

- The segment and the Page units provide the memory management and protection services for the 80386.
- They take the responsibility for address generation, address translation and segment checking from the bus interface unit and thereby further boosting the performance of the CPU.
- The segmentation unit has the ability to convert logical address into linear address.
- It contains dedicated hardware for performing high speed address calculations, logical to linear address translation and protection checks.

6. Paging Unit

- The page unit works only in protected mode.
- It contains the translation look aside buffer that stores recently used page directory and page table entries.
- When paging is enabled, the linear address produced by the segment unit is used as the input to the paging unit.
- Here the linear address is translated into the physical address of the memory or I/O location to be accessed. Thus physical memory is the output to the bus interface unit.

Versions of 80386

80386DX

- The first member in 80386 family
- This CPU could work with 16-bit and 32-bit external buses.
- Comprises of both 32-bit internal registers and 32-bit external bus.

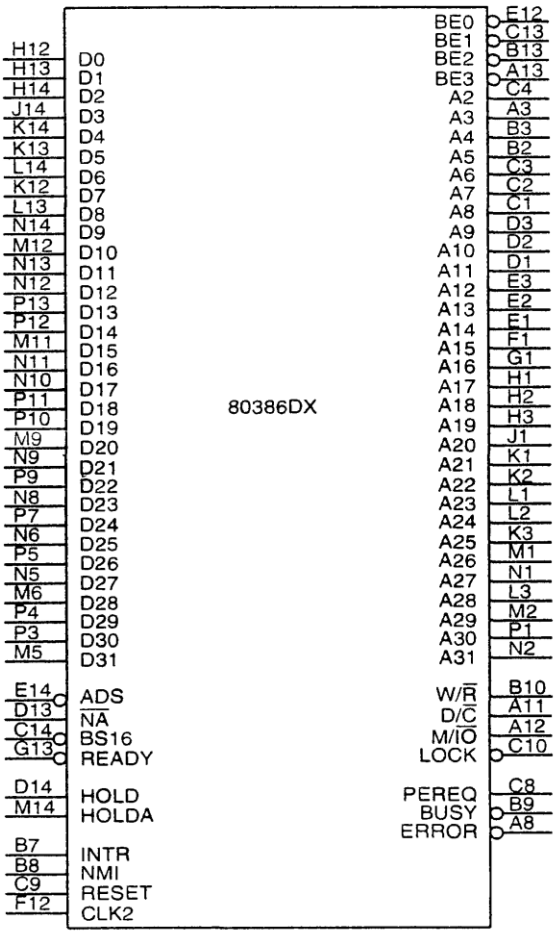
80386SX

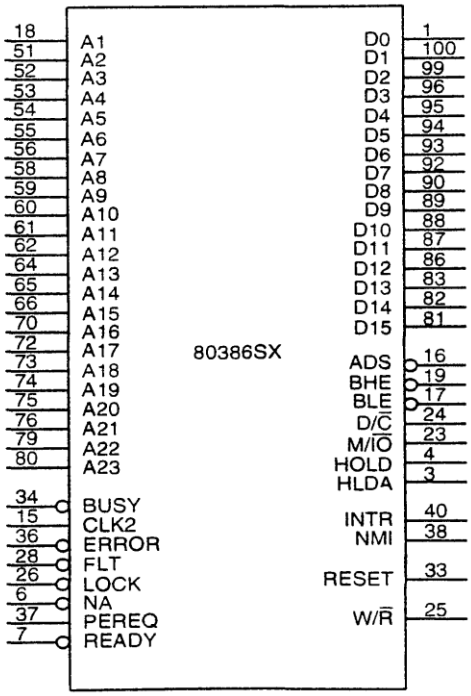
- Low cost version of the 80386.

- This processor had 16 bit external data bus, 32-bit internal registers and 24-bit external address bus.

80386SL

- Low-power microprocessor with power management features, with 16-bit external data bus and 24-bit external address bus.
- The processor included ISA bus controller, memory controller and Cache controller.





Datatypes of 80386 MP

The following data types are directly supported and thus implemented by one or more 80386 machine instructions; these data types are:

- Bit (boolean value), bit field (group of up to 32 bits) and bit string (up to 4Gb in length).
- 8-bit integer (byte), either signed (range -128...127) or unsigned (range 0...255).
- 16-bit integer, either signed (range -32,768...32,767) or unsigned (range 0...65,535).
- 32-bit integer, either signed (range -231...231-1) or unsigned (range 0...232-1).
- 64-bit integer, either signed (range -263...263-1) or unsigned (range 0...264-1).
- Offset, a 16 or 32-bit displacement referring to a memory location (using any addressing mode).
- Pointer, a 16-bit selector together with a 16 or 32 bit offset.
- Character (8-bit character code).
- String, a sequence of 8, 16 or 32-bit words (up to 4 GB in length).
- BCD, decimal digits (0...9) represented by unpacked bytes.
- Packed BCD, two BCD digits in one byte (range 0...99).

Intel 80386 MP Processing Modes

The processing mode of the 80386 determines the features that are accessible. The 80386 has three processing modes:

1. Real-Address Mode.
2. Virtual 8086 Mode.
3. Protected Mode.

- Real-address mode (often called just "real mode") is the mode of the processor immediately after RESET. In real mode the 80386 appears to

programmers as a fast 8086 with some new instructions. Most applications of the 80386 will use real mode for initialization only.

- Virtual 8086 mode (also called V86 mode) is a dynamic mode in the sense that the processor can switch repeatedly and rapidly between V86 mode and protected mode. The CPU enters V86 mode from protected mode to execute an 8086 program, then leaves V86 mode and enters protected mode to continue executing a native 80386 program.
- Protected mode is the natural 32-bit environment of the 80386 processor. In this mode all instructions and features are available.

Register Organization

- Eight 32 - bit general purpose registers which may be used as either 8 bit or 16 bit registers.
- A 32 - bit register known as an extended register, is represented by the register name with prefix E.
- Example: A 32 bit register corresponding to AX is EAX, similarly BX is EBX etc.
- The 16 bit registers BP, SP, SI and DI in 8086 are now available with their extended size of 32 bit and are names as EBP, ESP, ESI and EDI.
- AX represents the lower 16 bit of the 32 bit register EAX. BP, SP, SI, DI represents the lower 16 bit of their 32 bit counterparts, and can be used as independent 16 bit registers.
- The six segment registers available in 80386 are CS, SS, DS, ES, FS and GS.
- The CS and SS are the code and the stack segment registers respectively, while DS, ES, FS, GS are 4 data segment registers.
- A 16 bit instruction pointer IP is available along with 32 bit counterpart EIP.

Intel 80386 registers

$^{3}_{1}$ ... $^{1}_{5}$ ... $^{0}_{7}$ ... $^{0}_{0}$ *(bit position)*

Main registers *(8/16/32 bits)*

EAX	AX	AL	Accumulator register
EBX	BX	BL	Base register
ECX	CX	CL	Count register
EDX	DX	DL	Data register

Index registers *(16/32 bits)*

ESI	SI	Source Index
EDI	DI	Destination Index
EBP	BP	Base Pointer
ESP	SP	Stack Pointer

Program counter *(16/32 bits)*

EIP	IP	Instruction Pointer

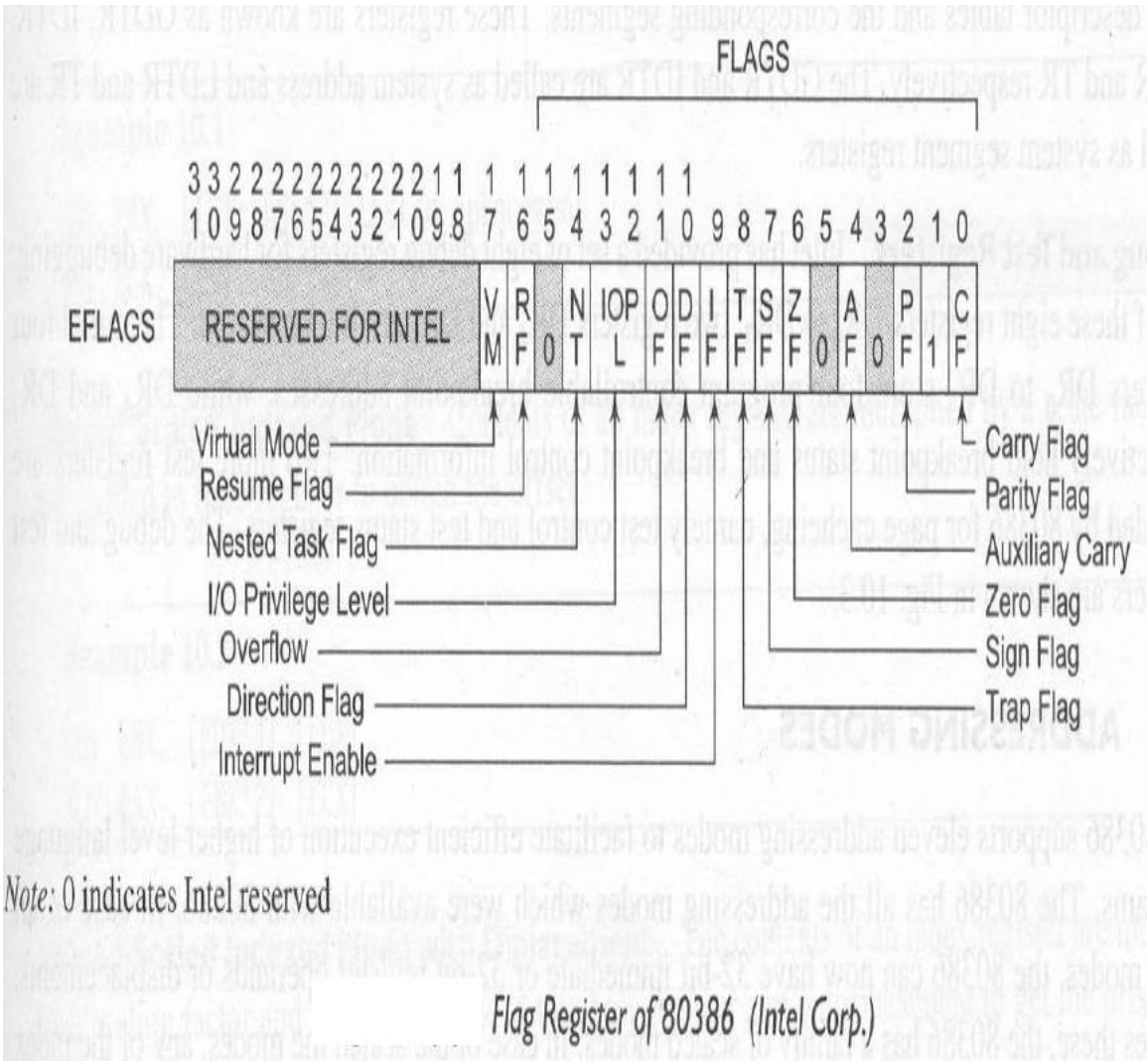
Segment selectors *(16 bits)*

	CS	Code Segment
	DS	Data Segment
	ES	ExtraSegment
	FS	F Segment
	GS	G Segment
	SS	Stack Segment

Status register

$^{1}_{7}$ $^{1}_{6}$ $^{1}_{5}$ $^{1}_{4}$ $^{1}_{3}$ $^{1}_{2}$ $^{1}_{1}$ $^{1}_{0}$ $^{0}_{9}$ $^{0}_{8}$ $^{0}_{7}$ $^{0}_{6}$ $^{0}_{5}$ $^{0}_{4}$ $^{0}_{3}$ $^{0}_{2}$ $^{0}_{1}$ $^{0}_{0}$ *(bit position)*

V	R	0	N	IOPL	O	D	I	T	S	Z	0	A	0	P	1	C	EFlags



Flag Register of 80386 (Intel Corp.)

Flags

1. **C (Carry)** –It holds the carry after calculations.
2. **P (Parity)** –Parity is a logic **0** for odd parity and a logic **1** for even parity. Parity is a count of ones in a number expressed as even or odd.
3. **A (Auxiliary carry)** – Carry occurs bits positions 3 and 5 of the results.
4. **Z (Zero)** – The zero flag shows that the result of an arithmetic or logical operation is zero. When Z = 1, the result is zero. When Z = 0, the result was non-zero.
5. **T (Trap)** – Enables trapping through an on-chip debugging facility.

6. **S (Sign)** – The sign flag holds the arithmetic sign after an arithmetic or a logical operation. If S =1 the sign bit is set and the result is negative. If S = 0, the sign bit is not set and the result is positive.
7. **I (Interrupt)** – The interrupt flag controls the operations of the INTR (Interrupt request) input pint. If I =1, the INTR pin is enabled; if I =0, the INTR pin is disabled.
8. **VM (virtual mode)** – If this flag set, the 80386 enters the virtual mode within the protected mode. In this mode, if any privileged instruction is executed, an exception 13 is generated.
9. **RF (resume)** – The resume flag is used with debugging to control the resumption of execution after the next instruction.
10. **NT (nested task)** – The nested task flag is used to indicate that the current task is nested within another task in protected mode operation. This flag is when the task I nested by software.
11. **IOPL (I/O Privilege level)** – IOPL is used in protected mode operation to select the privilege level for I /O devices. IF the current privilege level is higher or more trusted than the IOPL, I/O executed without hindrance. If the IOPL is lower than the current privilege level, an interrupt occurs.

Intel 80486 MP architecture:

- The 80486 is the next evolutionary step up from 80386.
- One of the most obvious features include in an 80386 is a built-in math coprocessor. This coprocessor is essentially the same as the 80387-processor used with 80386, but being integrated on the chip allows it to execute math instruction about three times as fast as 80386/387 combination
- 80486 is an 8kbyte code and data cache.

- The memory system of 80486 is identical to 80386 microprocessors. The 80486 cantinas 4Gbyte of memory beginning at location 00000000H and ending FFFFFFFFH.
- The major change to the memory system is internal to 80486 in the form of 8Kbyte cache memory which speeds the execution of instruction and the acquisition of data.
- Another addition in memory system is the parity checker/generator built into the 80486 microprocessors.
- The 80486 contains the same memory-management system as the 80386. This includes a paging unit to allow any 4Kbyte block of physical memory to be assigned 4 Kbyte block of linear memory. The only difference between 80386 and 80486 memory management system is paging.
- The 80486-paging system can disable caching for section of translation memory pages, while the 80386 could not.

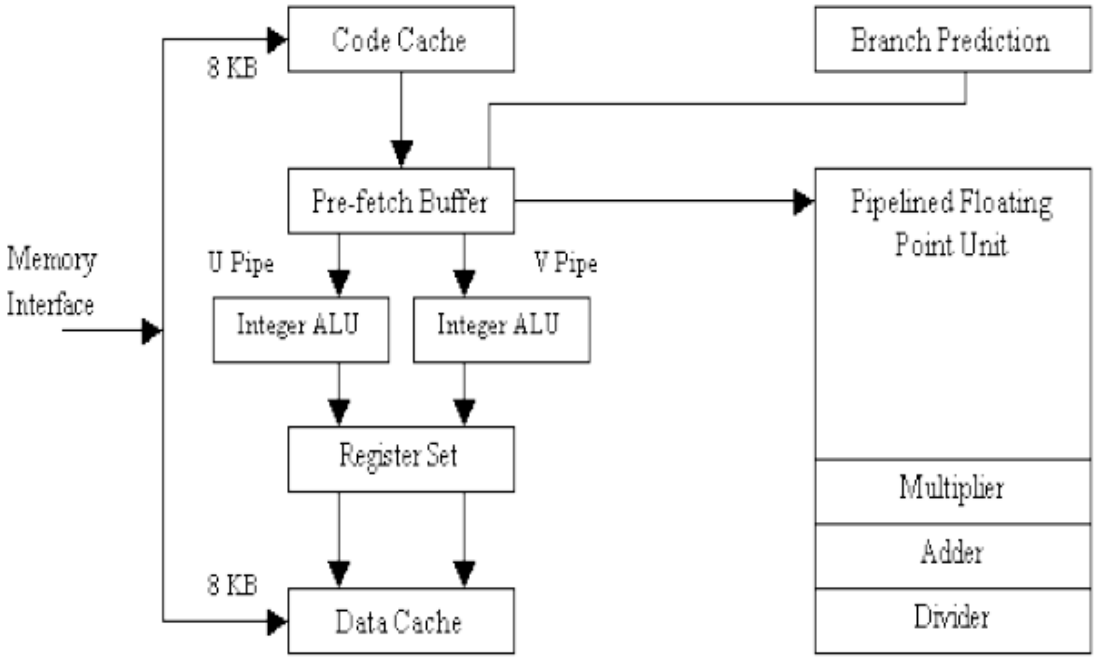


Figure: Intel 80486 MP architecture.

Parity Checker / Generator: Parity is often used to determine if data are correctly read from a memory location.

- Parity is generated by the 80486 during each write cycle. Parity is generated as even parity and a parity bit is provided for each byte of memory. The parity check bits appear on pins DP0-DP3, which are also parity inputs as well as parity outputs.
- These are typically stored in memory during each write cycle and read from memory during each read cycle.
- On a read, the microprocessor checks parity and generates a parity check error, if it occurs on the PCHK# pin. A parity error causes no change in processing unless the user applies the PCHK signal to an interrupt input.

EFLAG Register of the 80486:

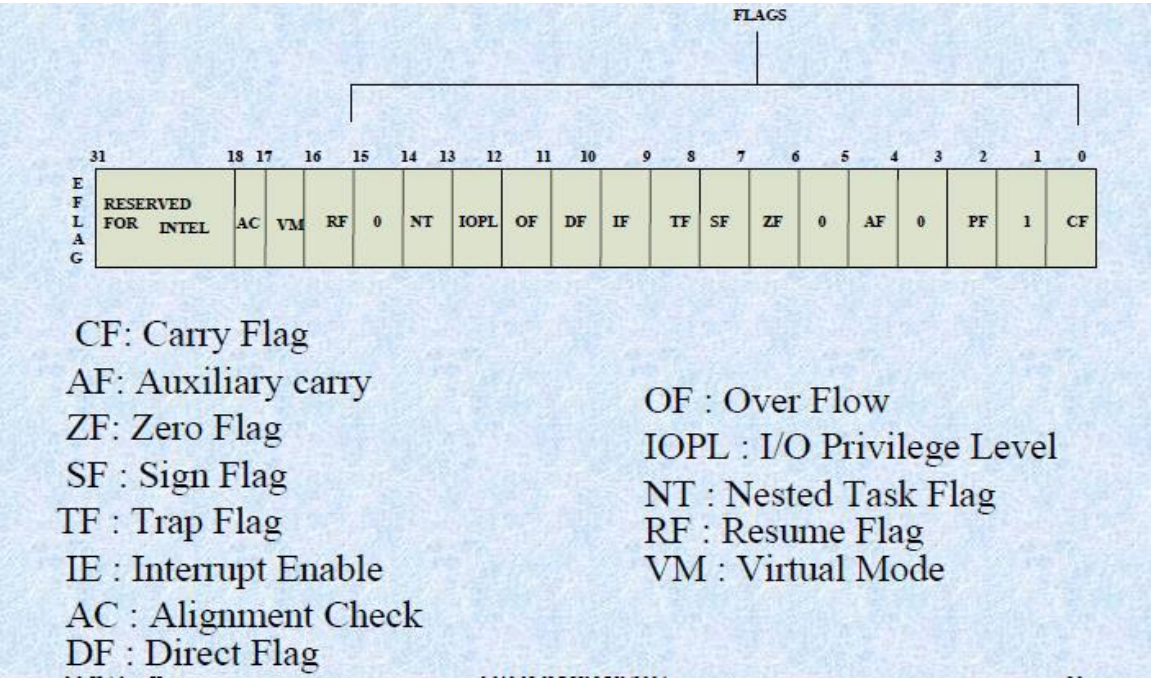


Figure: flag register of Intel 80486 MP.

The extended flag register EFLAG is illustrated in the figure above. The only new flag bit is the AC alignment check, used to indicate that the microprocessor has accessed a word at an odd address or a double word boundary.

The Floating-Point Unit:

The Intel Architecture Floating-Point Unit (FPU) provides high-performance floating-point processing capabilities. It supports the real, integer, and BCD-integer data types and the floating-point processing algorithms and exception hand ling architecture defined in the IEEE 754 and 854 Standards for Floating- Point Arithmetic. The FPU executes instructions from the processor's normal instruction stream and greatly improves the efficiency of Intel Architecture processors in handling the types of high-precision floating-point processing operations commonly found in scientific, engineering, and business, applications.

The architecture of the Intel Architecture FPU has evolved in parallel with the architecture of early Intel Architecture processors. The first Intel Math Coprocessors (the Intel 8087, Intel 287, and Intel 387) were companion processors to the Intel 8086/8088, Intel 286, and Intel386 processors·, respectively, and were designed to improve and extend the numeric processing capability of the Intel Architecture. The Intel486 DX processor for the first time integrated the CPU and the FPU architectures on one chip. The Pentium processors FPU offered the same architecture as the Intel486 DX processor's FPU, but with improved performance. The Pentium Pro processors FPU further extended the floating-point processing capability of Intel Architecture family of processors and added several new instructions to improve processing throughput. Throughout this evolution, compatibility among the various generations of FPUs and math coprocessors has been maintained. For example, the Pentium Pro processors FPU is fully compatible with the Pentium and Intel486 DX processor's FPUs.

Intel Pentium Pro Processor

1)The Pentium microprocessors have advanced superscalar, the *superscalar factor* (the maximum number of instructions that can be completed in a clock cycle) is three in the Pentium Pro processor, compared to two in Pentium processor. And the 2) data path width inside the Pentium Pro is 64-bits, double that of the Pentium. 3) Dynamic branch prediction is implemented in Pentium Pro that is similar to the Pentium processor. Figure (1) shows functional block diagram of the Pentium Pro processor micro architecture.

Referring to the diagram of Pentium Pro processor, we can divide the architecture into four processing units and the memory subsystem as follows:

- Memory subsystem: This consists of, system bus, L2 cache, bus interface unit, instruction cache (L1), data cache unit (L1), memory interface unit, and memory reorder buffer.
- Fetch/Decode unit: This unit comprises of instruction fetch unit, branch target buffer, instruction decoder, microcode sequencer, and register alias table.
- Instruction pool: This is made up of the reorder buffer.
- Dispatch/Execute unit: This has a reservation station, two integer units, two floating-point units, and two address generation units.
- Retire unit: This consists of the retire unit and retirement register file.

These processing units are discussed in little more detail in the following subsections.

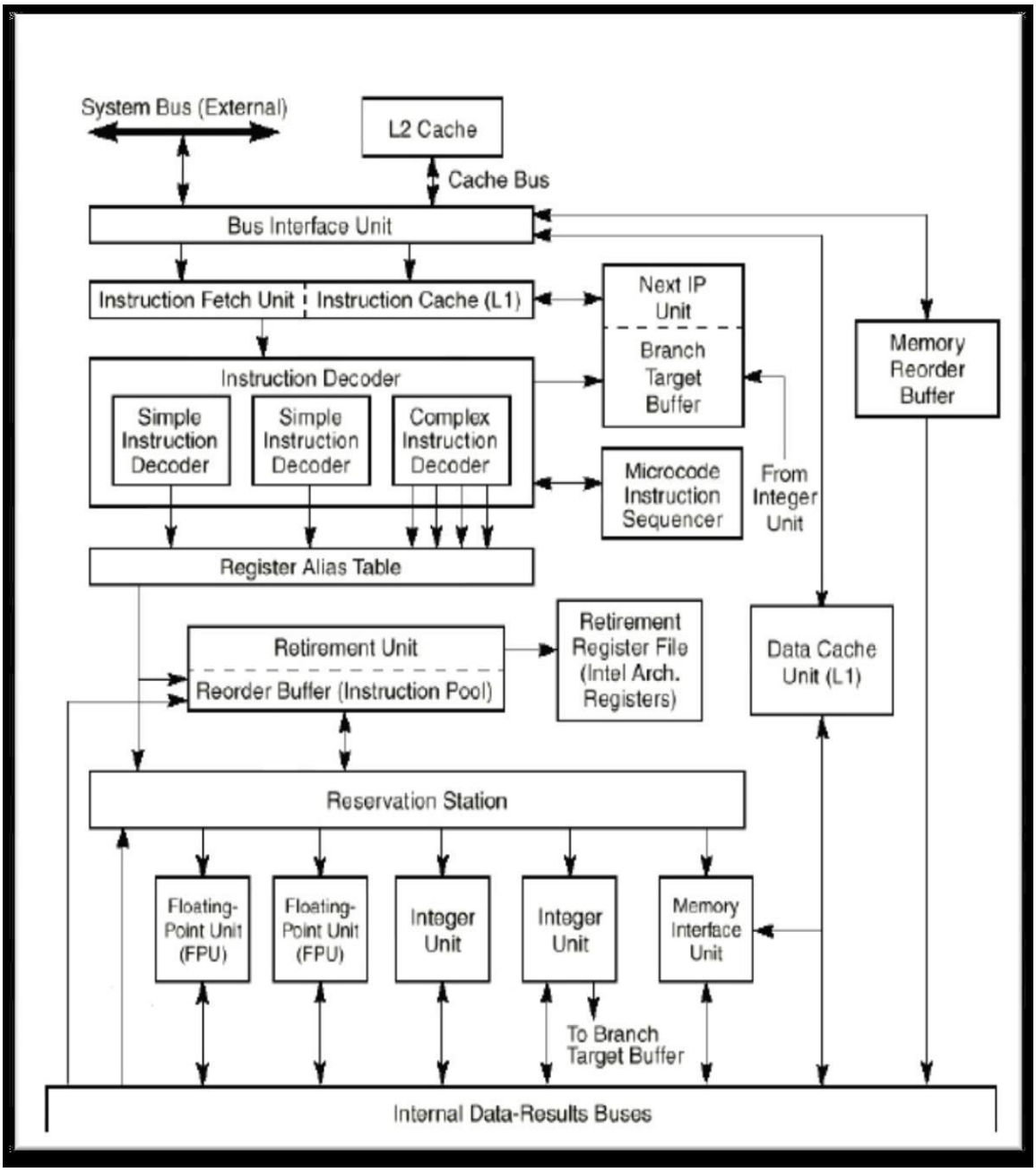


Figure (1): Functional block diagram of Pentium Pro processor.

Memory Subsystem

The memory subsystem for the Pentium Pro processor consists of main system memory, the primary cache (L1), and the secondary cache (L2). The bus interface unit accesses system memory through the external system bus. The external system bus is a 64-bit bus that handles each bus access as separate request and response

operations (transaction oriented bus). While the bus interface unit is waiting for a response to one bus request, it can issue numerous additional requests. The bus interface unit accesses the L2 cache through a 64-bit cache bus. This bus is also transactional oriented, supporting up to four concurrent cache accesses, and operates at the full clock speed of the processor. Bus interface unit gets access to the L1 caches is through internal buses. The L1 cache also operates at full clock speed. The 8-KByte L1 instruction cache is four-way set associative whereas the 8-KByte L1 data cache is two-way set associative, and dual-ported supporting one load and one store operation per cycle. Coherency between the caches and system memory are maintained using the MESI (modified, exclusive, shared, and invalid) cache protocol. Processor's execution units request memory through the memory interface unit and the memory order buffer.

These units have been designed to support a smooth flow of memory access requests through the cache and system memory hierarchy to prevent memory access blocking. The L1 data cache automatically forwards a cache miss on to the L2 cache. Memory requests to the L2 cache or system memory go through the memory reorder buffer. The memory reorder buffer functions as a scheduling and dispatch station. This unit keeps track of all memory requests and is able to reorder some requests to prevent blocks and improve throughput.

The Fetch/Decode Unit

The fetch/decode unit reads instructions from the L1 instruction cache and decodes them into a series of micro-operations (micro-ops). This micro-op stream is then sent to the instruction pool. From the instruction cache the instruction fetch unit fetches one 32-byte cache line per clock. It marks the beginning and end of the instructions in the cache lines and transmits 16 aligned bytes to the decoder. Basing on inputs from the branch target buffer, the interrupt status, and branch prediction

indications the instruction fetch unit computes the instruction pointer. The branch target buffer performs the branch prediction (branch prediction means that the microprocessor tries to predict whether the branch instruction will jump or not, based on a past history of the branch). The 512 entry branch target buffer looks many instructions ahead of the retirement program counter. The instruction decoder contains three parallel decoders:

- two simple-instruction decoders
- one complex instruction decoder.

Each decoder converts an instruction into one or more triadic micro-ops (two logical sources and one logical destination per micro-op). Micro-ops are primitive instructions that are executed by the processor's six parallel execution units. Many instructions are converted directly into single micro-ops by the simple instruction decoders, and some instructions are decoded into from one to four micro-ops. The more complex instructions are decoded into sequences of preprogrammed micro-ops obtained from the microcode instruction sequencer. The decoding of instruction prefixes and looping operations are handled by instruction decoders. The instruction decoder can generate up to six micro-ops per clock cycle. The processor provides 40 internal, general-purpose registers, which are used for the actual computations. These registers can handle both integer and floating point values.

The micro-ops from the instruction decoder are sent to the register alias table unit, where references to the logical architecture registers are converted into internal physical register references. Then the allocator in the register alias table unit adds status bits and flags to the micro-ops to prepare them for out-of-order execution and sends the resulting micro-ops to the instruction pool.

Instruction Pool (Reorder Buffer)

The reorder buffer is an array of content-addressable memory, arranged into 40 micro-op registers. It contains micro-ops that are waiting to be executed, as well as those that have already been executed but not yet committed to machine state. The dispatch/execute unit can execute instructions from the reorder buffer in any order.

Dispatch/Execute Unit

The dispatch/execute unit schedules and executes the micro-ops stored in the reorder buffer according to data dependencies and resource availability. The reservation station handles the scheduling and dispatching of micro-ops from the reorder buffer. The results of a micro-op execution are returned to the reorder buffer and stored along with the micro-op until it is retired. If two or more micro-ops of the same type are available at the same time, then the reorder buffer follows a FIFO algorithm to execute them. Two integer units, two floating-point units, and one memory-interface unit handle execution of micro-ops. Thus up to five micro-ops can be scheduled per clock. The two integer units can handle two integer micro-ops in parallel. One of integer units is designed to handle branch micro-ops. This unit detects branch mis-predictions and signals the branch target buffer to restart the pipeline. The memory interface unit handles the load and store micro-ops. The memory interface unit executes both a load and a store in parallel in one clock cycle. The floating-point execution units are similar to those found in the Pentium processor, few new floating-point instructions have been added to the Pentium Pro processor.

Retirement Unit

The retirement unit commits the results of speculatively executed (decided by branch prediction mechanism) micro-ops to permanent machine state and removes the micro-ops from the reorder buffer. The retirement unit continuously checks the

status of micro-ops in the reorder buffer, similar to the reservation buffer. It then retires completed micro-ops in their original program order, taking into accounts interrupts, exceptions, breakpoints, and branch mis-predictions. The retirement unit can retire three micro-ops per clock. In retiring a micro-op, it writes the results to the retirement register file and/or memory. The retirement register file contains the architecture registers (eight general-purpose registers and eight floating-point data registers). After the results have been committed to machine state, the micro-op is removed from the reorder buffer.

Instruction Set Architecture Features

To make a computer hardware work we must speak to the hardware in its language. The words of this machine language are called instructions, and the vocabulary is called an instruction set. The Pentium processor is a CISC (Complex-Instruction-Set-Computer) architecture, but it achieves high performance by using many organizational features of RISC (Reduced-Instruction-Set Computer) architecture.

All the Intel Architecture instructions divided into four major groups:

- integer
- MMX technology
- floating-point
- System instructions.

Integer Instructions

Integer instructions perform the integer arithmetic, logic, and program flow control operations that programmers commonly use to write application and system software to run on an Intel Architecture processor. The integer instructions include different types of instructions like, data transfer instructions (PUSH, POP, MOV etc.); binary arithmetic instructions (ADD-integer add, ADC -Add with carry, SUB-

Subtract, SBB-Subtract with borrow etc.); Decimal Arithmetic (DAA-Decimal adjust after addition, DAS-Decimal adjust after subtraction, etc.); Logic Instructions (AND, OR, XOR, NOT); Shift and Rotate Instructions (SAR-Shift arithmetic right, SHR-Shift logical right, etc.)

MMX™ Technology Instructions

The MMX instructions execute on those Intel Architecture processors that implement the Intel MMX technology. These instructions operate on packed-byte, packed-word, packed-doubleword, and quadword operands. All of the MMX technology instructions are grouped as MMX™ Conversion Instructions, MMX™ Packed Arithmetic Instructions, MMX™ Comparison Instructions, MMX™ Logic Instructions, MMX™ Shift and Rotate Instructions, or MMX™ State Management.

Floating-Point Instructions

The floating-point instructions are those that are executed by the processor's floating point unit (FPU). These instructions operate on floating-point (real), extended integer, and binary-coded decimal (BCD) operands. These instructions include different types like, Data Transfer (FLD-Load real, FST-Store real, etc.); Basic Arithmetic (FADD- Add real, FADDP-Add real and pop, etc.); Comparison (FCOM-Compare real, FCOMP Compare real and pop, etc.)

System Instructions

These instructions are used to control those functions of the processor that are provided to support for operating systems and executives.

Intel Pentium III Processor

The Pentium III is essentially a Pentium II running at higher speed, with two interesting and useful features:

- The processor serial number and
- Streaming SIMD Extensions (SSE).

The processor serial number (or chip ID) is a unique identifier 'burned' into the Pentium III processor that can be accessed over the internet, allowing e-commerce sites and others to know which machine is visiting a site or using a service. This has drag Intel to a major controversy. But Intel claims that processor serial number can add value to a wide range of applications in both business and consumer computing. The advantages that the processor serial number can provide are discussed below.

- **Security:** The e-commerce depends on the assurance that only the authorized people access the confidential information. Applications that take advantage of the processor serial number can use that as another element of identification thus increasing confidentiality. Similarly, processor serial number can strengthen the data security for the consumer web sites who wants to maintain a section open only to their family members or so. It can also be used in businesses for adding a level of validation to electronic signature approvals.
- **Manageability:** IT departments use various ways to track assets such as MAC address or BIOS's GUID. But Intel claims that all these could be erased, so less reliable. But, processor serial number can be reliably used as a once it is burn on the chip at the time of manufacture it can never be erased. So designing applications using chip ID can help IT customers to manage their resources more efficiently.
- **Information Management:** Companies can turn information into a competitive advantage if they can manage it effectively. Information related applications can use processor serial numbers to handle tasks ranging from finding multiple copies of virus-infected document, tracking change information, to delivering customized information to the end user.

The other significant feature of the Pentium processor is the Streaming SIMD Extensions (SSE). Usually, the processors are SISD meaning Single Instruction and

Single Data thus processing one data in one instruction. MMX and SSE, both share the concept of SIMD, they differ in the type of data they handle, and the way they are supported in the processor. MMX instructions are SIMD for integers, while SSE instructions are SIMD for single-precision floating-point number. MMX instructions operate on two 32-bit integers simultaneously, while SSE instructions operate on four 32- bit floats simultaneously. A major difference between MMX and SSE is that no new registers were defined for MMX, while eight new registers have been defined for SSE. The SSE can be used in 3D graphics applications.