UNIT-9 80386 PROCESSOR

Aniruddh Fataniya

INTRODUCTION

80386 Microprocessor is a 32-bit processor that holds the ability to carry out 32-bit operations in one cycle.

It has a data and address bus of 32-bit each. Thus has the ability to address 4 GB (or 232) of physical memory

Multitasking and **protection capability** are the two key characteristics of the 80386 microprocessor

80386 has internal dedicated hardware that permits multitasking.

INTRODUCTION

We know 8086 is a 16-bit microprocessor and 80286 was an advancement of 8086 with some additional characteristics.

But with the advent of technology intel introduced a 32-bit microprocessor whose processing speed was twice that of the 80286 microprocessor.

This was an 80386 microprocessor that was designed by Intel in October 1985 and was an upgraded version of the 80286 microprocessor.

FEATURES

As it is a 32-bit microprocessor. Thus has a 32-bit ALU.

80386 has a data bus of 32-bit.

It holds an address bus of 32 bit.

It supports physical memory addressability of 4 GB and virtual memory addressability of 64 TB.

80386 supports a variety of operating clock frequencies, which are 16 MHz, 20 MHz, 25 MHz, and 33 MHz.

It offers 3 stage pipeline: fetch, decode and execute. As it supports simultaneous fetching, decoding, and execution inside the system.

OPERATING MODES OF 80386

80386 supports 3 operating modes:

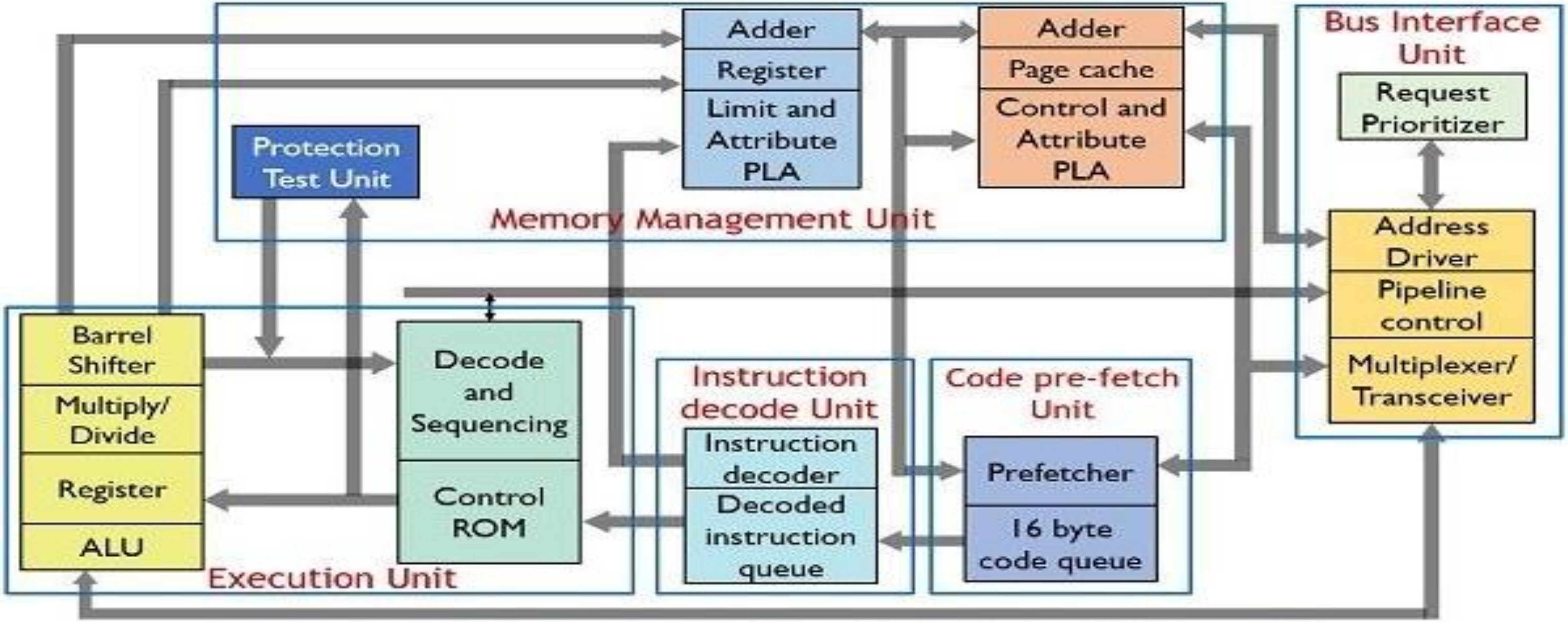
- real,
- protected, and
- virtual real mode

OPERATING MODES OF 80386

In the **protected mode,** 80386 microprocessor operates in a similar way like 80286 but offers higher memory addressing ability.

In **virtual mode,** the overall memory of 80386 can be divided into various virtual machines. And all of them acts as a separate computer with 8086 microprocessor. This mode is also called virtual 8086 mode or V86 mode.

The other one is the **virtual real mode,** this mode allows the system to execute multiple programs in the protected memory. And in case a program at a particular memory gets crashed then it will not cause any adverse effect on the other part of the memory.



Architecture of 80386 Microprocessor

ARCHITECTURE

it has 5 functional units which are as follows:

Bus Interface Unit

Code Pre-Fetch Unit

Instruction Decode Unit

Execution Unit

Memory Management Unit

BUS INTERFACE UNIT

The bus interface unit or BIU holds a 32-bit bidirectional data bus as well as a 32-bit address bus.

Whenever a need for instruction or a data fetch is generated by the system then the BIU generates signals (according to the priority) for activating the data and address bus in order to fetch the data from the desired address.

The BIU connects the peripheral devices through the memory unit and also controls the interfacing of external buses with the coprocessors.

CODE PRE-FETCH UNIT

This unit fetches the instructions stored in the memory by making use of system buses.

Whenever the system generates a need for instruction then the code prefetch unit fetches that instruction from the memory and stores it in a 16-byte prefetch queue.

So to speed up the operation this unit fetches the instructions in advance and the queue stores these instructions.

The sequence in which the instructions are fetched and gets stored in the queue depends on the order they exist in the memory.

INSTRUCTION DECODE UNIT

This unit decodes the instructions stored in the pre-fetch queue

Basically the decoder changes the machine language code into assembly language and transfers it to the processor for further execution

EXECUTION UNIT

The decoded instructions are stored in the decoded instruction queue

So, these instructions are provided to the execution unit in order to execute the instructions

The execution unit controls the execution of the decoded instructions

This unit has a 32-bit ALU, that performs the operation over 32-bit data in one cycle

Also, it consists of 8 general purpose as well as 8 special purpose registers

These are used for data handling and calculation of offset address

MEMORY MANAGEMENT UNIT

This unit has two separate units within it. These are

- Segmentation Unit and
- Paging Unit

MEMORY MANAGEMENT UNIT: SEGMENTATION UNIT

The segmentation unit plays a vital role in the 80836 microprocessor.

It offers a protection mechanism in order to protect the code or data present in the memory from application programs.

It gives 4 level protection to the data or code present in the memory.

Every information in the memory is assigned a privilege level from PL0 to PL3.

Here, ***PL0 holds the highest priority*** and PL3 holds the lowest priority

MEMORY MANAGEMENT UNIT: SEGMENTATION UNIT

Suppose a file (either data or code) is needed to be accessed is stored in the memory at PL0.

Then only those programs which are working at PL0 would be able to access that file.

While other programs will not be able to access the same.

Also, if a file is present at PL1, then programs of PL0 and PL1 both can access it.

As PL0 has a higher priority than PL1.

So, for protection purposes, the main part of the OS is stored in PL0 while PL3 holds the user programs.

Providing protection to the data or code inside the system is the most advantageous factor that was first given by the 80386 microprocessor.

MEMORY MANAGEMENT UNIT: PAGING UNIT

operates only in protected mode

changes the linear address into a physical address

The segmentation unit controls the action of the paging unit, as the segmentation unit has the ability to convert the logical address into the linear address at the time of executing an instruction

Basically, it changes the overall task map into pages and each page has a size of **4KB**

This allows the handling of tasks in the form of pages rather than segments

MEMORY MANAGEMENT UNIT: PAGING UNIT

The paging unit supports multitasking

This is so because the physical memory is not required to hold the whole segment of any task

Despite this, only that part of the segment which is needed to be currently executed must be stored in that memory whose physical address is calculated by the paging unit

This resultantly reduces the memory requirement and hence this frees the memory for other tasks

Thus by this we get an effective way for managing the memory to support multitasking

Thank You