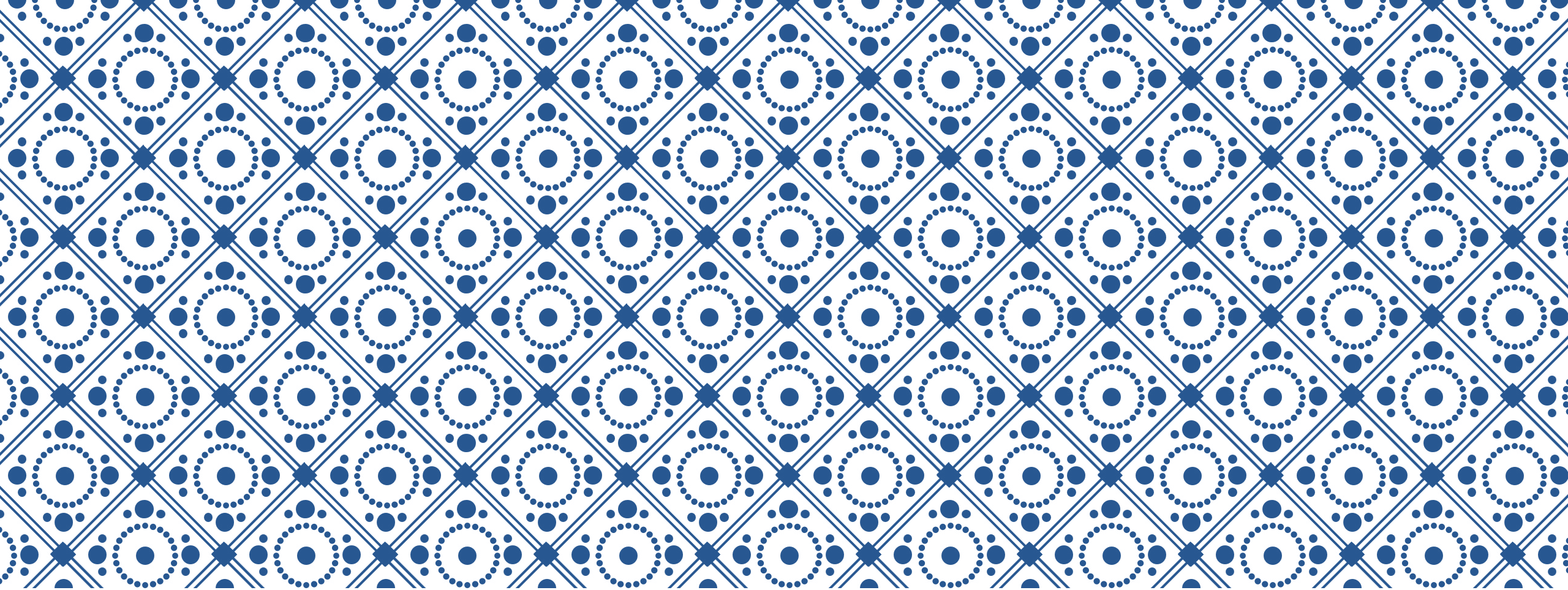


UNIT-8 8086,80186, 80286 PROCESSOR

Aniruddh Fataniya

8086 Microprocessor

INTRODUCTION

Intel introduced its first 4-bit microprocessor in 1971 and its 8-bit microprocessor 8008 in 1972

They could not survive as general purpose microprocessor due to design and performance limitations

First general purpose 8-bit microprocessor 8080 was introduced in 1974 by intel

First step towards the advanced microprocessor

8085 followed 8080, with few more added features to its architecture

INTRODUCTION

The main limitations of the 8-bit microprocessors were

- Low speed
- Low memory addressing capability
- Limited number of general purpose registers
- Less powerful instruction set

All these limitations pushed designers to build more powerful processors in terms of

- advanced architecture
- more processing capability
- large memory capability and
- more powerful instruction set

The result was 8086

REGISTER ORGANIZATION

8086 has powerful set of registers known as

- General purpose registers
- Special purpose registers

All of them are of 16-bits

GENERAL PURPOSE REGISTERS

The **general purpose registers** can be used as either 8-bit registers or 16-bit registers

They may be used either used for

- Holding data, variables and intermediate results temporarily or
- Other purposes like a counter or for storing offset address for some particular addressing modes

SPECIAL PURPOSE REGISTERS

The **special purpose registers** are used as

- Segment registers, pointers, index registers or
- as offset storage registers for particular addressing modes

GENERAL DATA REGISTERS

AX, BX, CX and DX are the general purpose 16-bit registers

AX is used as accumulator, with the lower 8-bits of AX designed as AL and higher 8-bits as AH

AL can be used as 8-bit accumulator for 8-bit operations

This is the most important register having multiple functions

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

GENERAL DATA REGISTERS

BX is used as an offset storage for forming physical address in case of certain addressing modes

CX is useful as a default counter in case of string and loop instructions

DX is a general purpose register which may be used as an implicit operand or destination in case of a few instructions

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

SEGMENT REGISTERS

Unlike 8085, the 8086 addresses a segmented memory

The complete 1 MB memory, which the 8086 addresses is divided into 16 logical segments

Each segment contains 64 KB of memory

There are four segment registers:

- **Code segment register (CS) -** addressing a memory location in the code segment of the memory, where executable program is stored
- **Data segment register (DS) -** points to the data segment of he memory, where the data is stored
- **Extra segment register (ES) -** another data segment
- **Stack segment register (SS) -** addressing stack segment of memory which is used to store stack data

SEGMENT REGISTERS

The CPU uses stack for temporarily storing the important data

While addressing any location in the memory bank, the physical address is calculated from two parts

- Segment address
- Offset

The segment registers address contain 16-bit base addresses, related to different segments

Any of the pointers and index registers or BX may contain the offset of the location to be addressed

SEGMENT REGISTERS

The advantage:

Instead of maintaining 20-bit register for a physical address, the processor just maintains two 16-bit registers which are within the word length capacity of the machine

Thus CS, DS, SS, ES segment registers, respectively, contain the segment address for code, data, stack and extra segments of memory

These all segments are logical segments

They may or may not be physically separated

POINTERS AND INDEX REGISTERS

The pointers contain offset within the particular segments

The pointers IP, BP and SP usually contain offsets within the code (IP), and stack (BP & SP) segments

The index registers are used as general purpose registers as well as for offset storage in case of indexed, based indexed and relative indexed addressing modes

The register SI is generally used to store the offset of source data in the data segment

DI is used to store the offset of destination in data or extra segment

The index registers are particularly useful for string manipulation

FLAG REGISTER

Indicates results of computations in the ALU

It also contains some flag bits to control the CPU operations

8086 Architecture

8086 ARCHITECTURE

8086 provides a number of improvements over 8085

- It supports a 16-bit ALU
- A set of 16-bit registers
- Provides segmented memory addressing capability
- A rich instruction set
- Powerful interrupt structure
- Etc…

8086 ARCHITECTURE

The complete architecture can be divided into two parts

- Bus Interface Unit (BIU)
- Execution Unit (EU)

The BIU contains the circuit for physical address calculations and a predecoding instruction byte queue (6 bytes long)

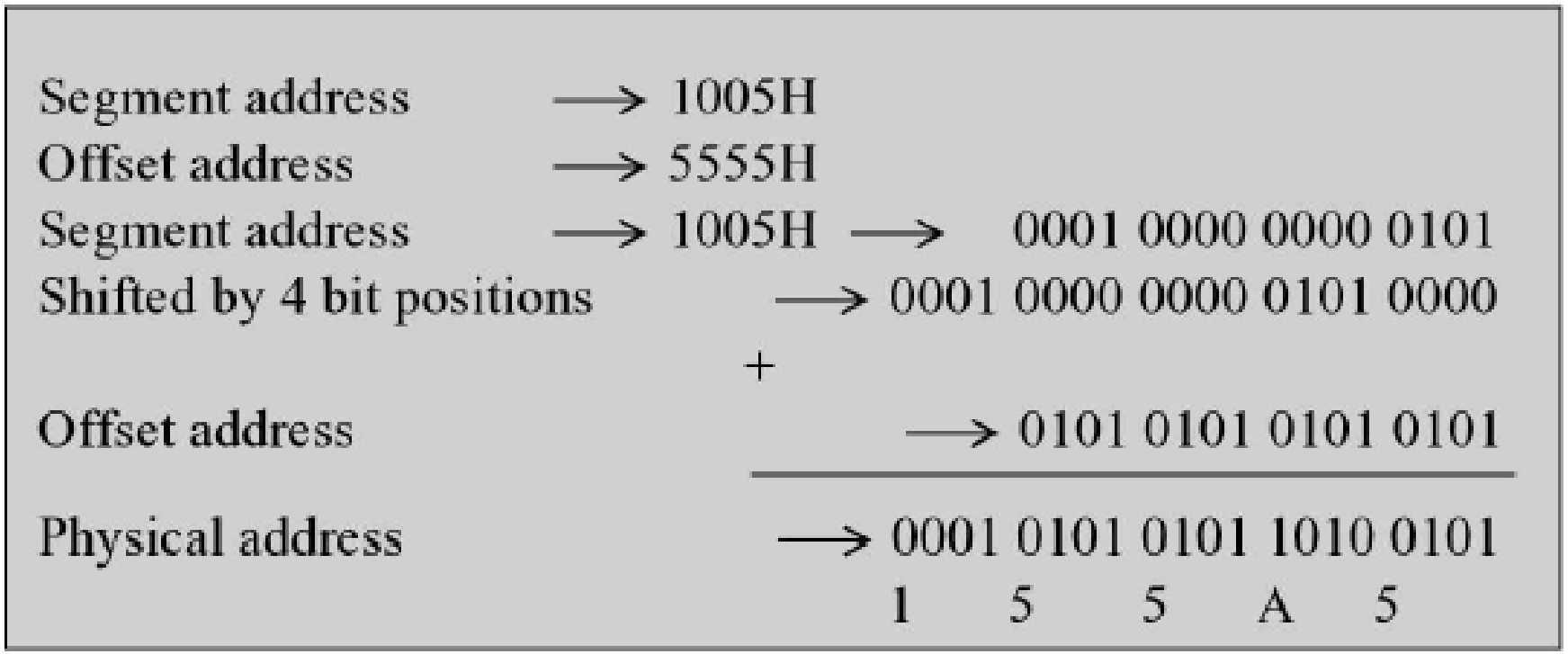
This unit is responsible for establishing communications with external devices and peripherals including memory via bus

8086 addresses a segmented memory

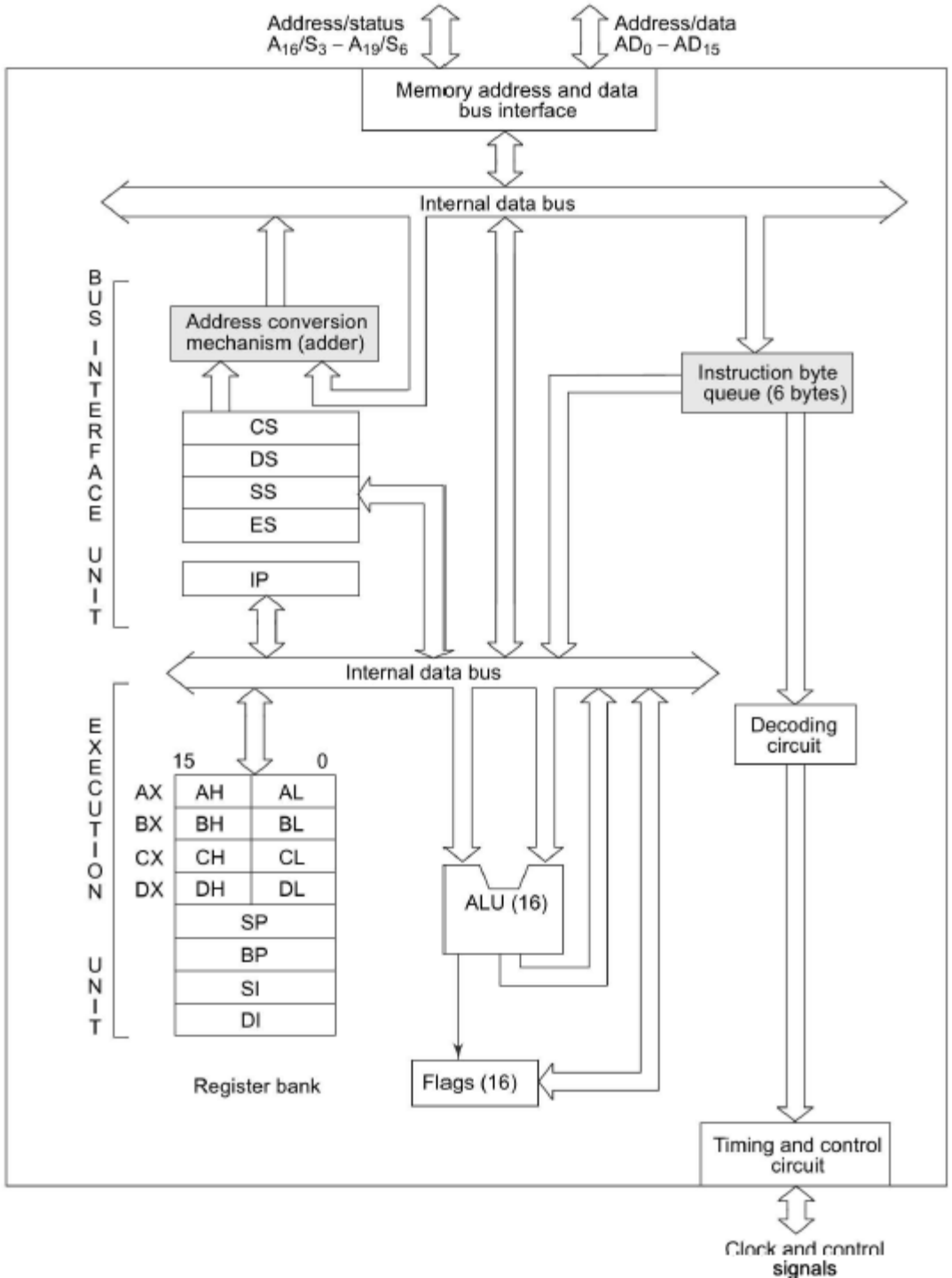
The complete physical address which is 20-bits long is generated using segment and offset registers, each 16-bits long

8086 ARCHITECTURE

Physical address calculation



8086 Architecture



8086 ARCHITECTURE

The segment addressed by the segment value 1005H can have offset values from 0000H to FFFFH within it, i.e. maximum 64K locations may be accommodate

The segment register indicates the base address of a particular segment

Offset address indicates the distance of the required memory location in the segment from the base address

The bus interface has a separate adder to perform this procedure for obtaining a physical address while addressing memory

8086 ARCHITECTURE

The segment address value is to be taken from an appropriate segment register depending upon whether code, data, or stack are to be accessed

The offset may be the content of IP, BX, SI, DI, SP, BP or an intermediate 16-bit value, depending upon the addressing mode

In case of 8085, once the opcode is fetched and decoded, the external bus remains free for some time, while the processor internally executes the instruction

This time slot is utilized in 8086 to achieve the overlapped fetch and execution cycle

8086 ARCHITECTURE

While the fetch instruction is executed internally, the external bus is used to fetch the machine code of the next instruction and arrange it in a queue called predecoded instruction byte queue

It is a 6 bytes long, first-in-first-out structure

The instruction from queue are taken for decoding sequentially

Once a byte is decoded, the queue is rearranged by pushing it out and the queue status is checked for the possibility of the next opcode fetch cycle

8086 ARCHITECTURE

While the opcode is fetched by the BIU, the EU executes the previously decoded instruction concurrently

The BIU along with the EU thus forms a pipeline

The BIU, thus manages the complete interface of execution unit with memory and I/O devices under the control of the timing and control unit

8086 ARCHITECTURE

The EU contains the register set of 8086 except segment register and IP

It has a 16-bit ALU, able to perform arithmetic and logical operations

The 16-bit flag register reflects the results of the execution by ALU

The decoding unit decodes the opcode bytes issued from the instruction byte queue

The timing and control unit derives the necessary control signals to execute the instruction opcode received from the queue

The EU may pass the results to the BIU for storing them in memory

MEMORY SEGMENTATION

The memory in an 8086/8088 based system is organized as segmented memory

In this scheme, the complete physically available memory may be divided into a number of logical segments

Each segment is 64KB in size and is addressed by one of the segment registers

The 16-bit contents of the segment register actually points to the starting location of a particular segment

To address a specific memory location within a segment, offset address is required

The offset address is 16-bit long so the maximum value can be FFFFH, and maximum size of any segment is 64K locations

MEMORY SEGMENTATION

The CPI8086 is able to address 1MB of physical memory

The complete 1MB can be divided into 16 segments, each of 64KB

The addresses may be assigned as 0000H to F000H respectively

The offset values are from 0000H to FFFFH, so the physical addresses range from 00000H to FFFFFH

In this case, the segments are called as **non-overlapping segments**

In some cases segments may be **overlapping**

MEMORY SEGMENTATION

Suppose a segment starts at a particular address and its maximum size can be 64KB

If another segment starts before its 64KB locations of the first segment, the two segments are said to be overlapping segments

The area of memory from the start of the second segment to the possible end of the first segment is called overlapped segment area

The locations lying in the overlapped area may be addressed by the same physical address generated from two different sets of the segment and offset address i.e. CS1 + IP1 = CS2 + IP2 ('+' indicates physical address calculation)

MEMORY SEGMENTATION

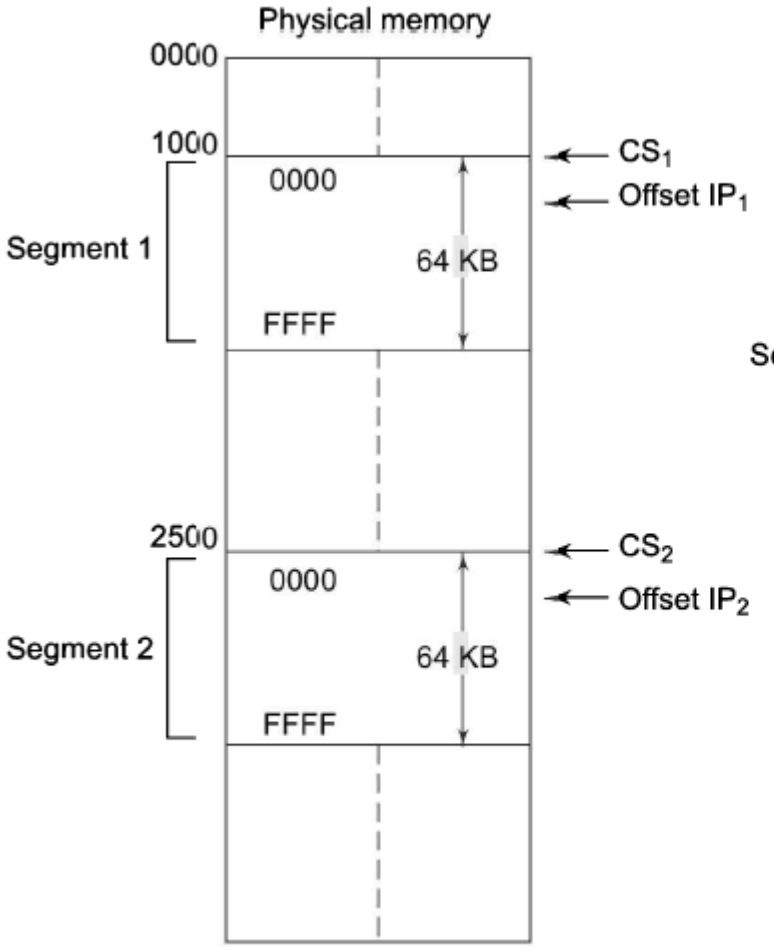


Fig. 1.3(a) *Non-overlapping Segments*

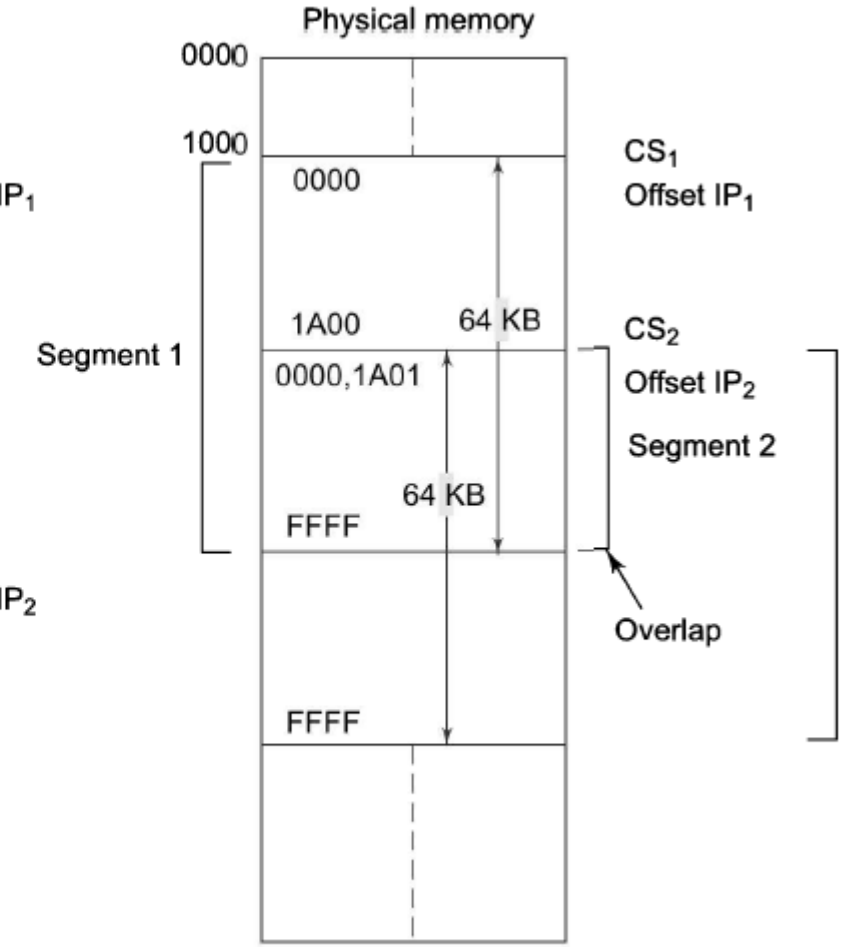


Fig. 1.3(b) *Overlapping Segments*

ADVANTAGES OF SEGMENTATION

It allows to extend the address ability of the processor, i.e. segmentation allows the use of 16 bit registers to give an addressing capability of 1 Megabytes. Without segmentation, it would require 20 bit registers.

Allows placing of code, data and stack portions of the same program in different parts (segments) of the memory, for data and code protection

Permits a program and/or its data to be put into different areas of memory each time the program is executed

FLAG REGISTER

8086 ha a 16-bit flag register which is divided into two parts

- Condition code or status flags
- Machine control flags

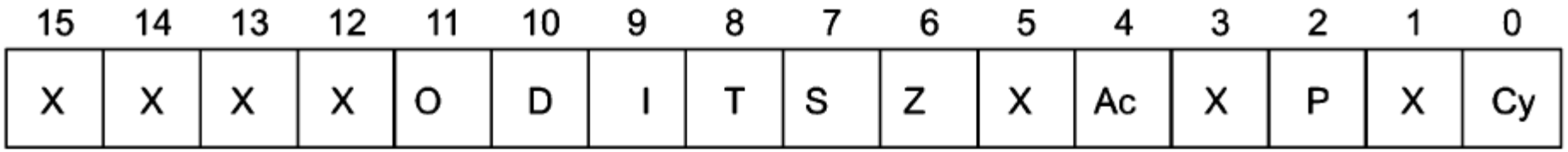
The **condition code flag register** is lower byte of the 16-bit flag register along with the overflow flag

This part of the flag reflects the results of the operations performed by ALU

The **control flag register** is the higher byte of the flag register of 8086

It contains three flags, direction flag, interrupt flag and trap flag

FLAG REGISTER



O — Overflow flag
D — Direction flag
I — Interrupt flag
T — Trap flag
S — Sign flag
Z — Zero flag
Ac — Auxiliary carry flag
P — Parity flag
Cy — Carry flag
X — Not used

FLAG REGISTER

O-Overflow Flag

- Set if overflow occurs, i.e. if the result of a signed operation is large enough to be accommodated in a designation register

D-Direction Flag

- Used by string manipulation instructions
- If this flag bit is '0', the string is processed from the lowest address to the highest address, i.e., auto incrementing mode
- Otherwise, the string is processed from the highest address towards the lowest address, i.e., auto decrementing mode

FLAG REGISTER

I-Interrupt Flag

- This flag is for interrupts
- If interrupt flag is set (1), the microprocessor will recognize interrupt requests from the peripherals.
- If interrupt flag is reset (0), the microprocessor will not recognize any interrupt requests and will ignore them.

T-Trap Flag

- This flag is used for on-chip debugging. Setting trap flag puts the microprocessor into single step mode for debugging. In single stepping, the microprocessor executes a instruction and enters into single step ISR (Interrupt Service Routine).
- If trap flag is set (1), the CPU automatically generates an internal interrupt after each instruction, allowing a program to be inspected as it executes instruction by instruction.
- If trap flag is reset (0), no function is performed.

FLAG REGISTER

S-Sign Flag

- Set when the result of any computation is negative
- The sign flag equals the MSB of the result

Z-Zero Flag

- Set if the result of the computation or comparison performed by the previous instruction/instructions is zero

Ac-Auxiliary carry Flag

- Set if there is a carry from the lowest nibble during addition or borrow from the lowest nibble during subtraction

FLAG REGISTER

P-Parity Flag

- Set to 1 if lower byte of the result contains even number of 1s

Cy-Carry Flag

- Set when there is a carry out of MSB in case of addition or borrow in case of subtraction

80186 Microprocessor

80186

The 80186 doubles the performance of 8086

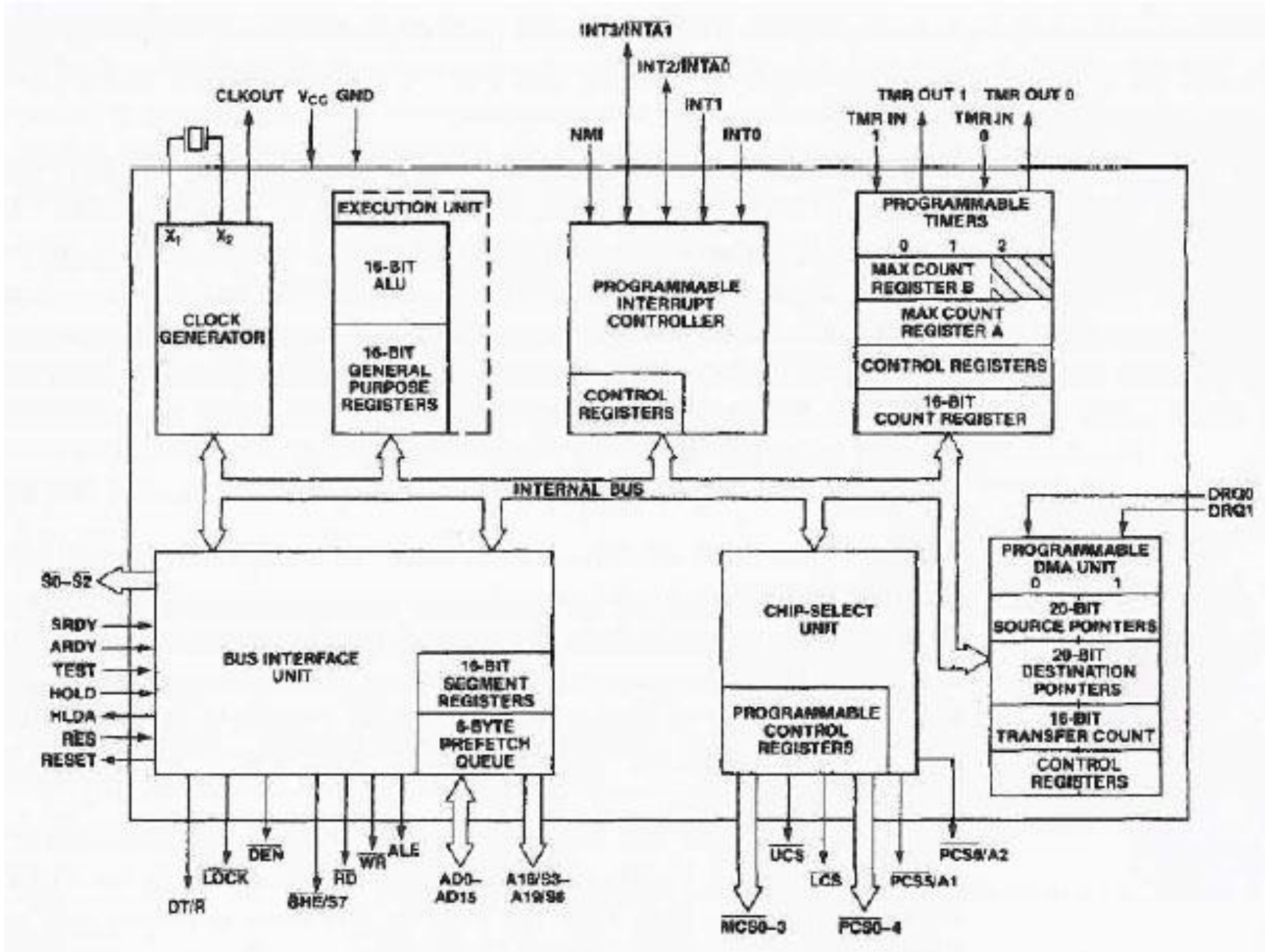
80186 = 8086 + several additional chips

Contains 16-bit data bus and 20-bit address bus

The total addressable memory size is 1MB

The instruction set is upward compatible with 8086/88 family

80186 Architecture



80186

The CPU is divided into seven independent functional parts

- The Bus Interface Unit (BIU)
- Execution Unit (EU)
- Clock Generator
- Programmable interrupt controller
- Programmable Chip Select Unit (CSU)
- Programmable DMA Unit
- Programmable counter/timers

BUS INTERFACE UNIT (BIU)

The 80186 has the same bus interface unit (BIU) and execution unit (EU) as the 8086

CLOCK GENERATOR

The 80186 has the built-in clock generator so that we can add external crystal

The system crystal determines the speed at which a CPU operates

This reduces the component count in a system

PROGRAMMABLE INTERRUPT CONTROLLER

It allows internal and external interrupts and controls up to two external 8259A PIC

Accepts interrupts only in master mode

PROGRAMMABLE CHIP SELECT UNIT

This built-in address decoder unit can be programmed to produce an active low chip select signal when the memory or port address in a specified range is sent out.

This built-in decoder is to select major blocks of memory

Also selects I/O

PROGRAMMABLE DMA UNIT

Direct memory access (DMA) is a feature of computer systems that allows certain hardware subsystems to access main system memory (random-access memory) independently of the central processing unit (CPU)

The DMA unit has two DMA request inputs

These inputs allow external devices such as disk controllers, CRT controllers etc. to request use of one of the DMA channels

For each DMA channel, the 80186 has a full 20-bit register to hold the source address, a 20-bit register to hold the destination address, and a 16-bit counter to keep track of how many words or bytes have been transferred

PROGRAMMABLE COUNTER/TIMERS

The timer section contains three fully programmable 16-bit timers

Timers 0 and 1 generate waveforms for external use and are driven by either the master clock of 80186 or by an external clock

They are also used to count external events

The third timer, timer 2, is internally connected to the processor clock

The output of timer 2 generates an interrupt after a specified number of clocks and can provide a clock to the other timers

80286 Microprocessor

80286

80286 Microprocessor is a 16-bit microprocessor that has the ability to execute 16-bit instruction at a time

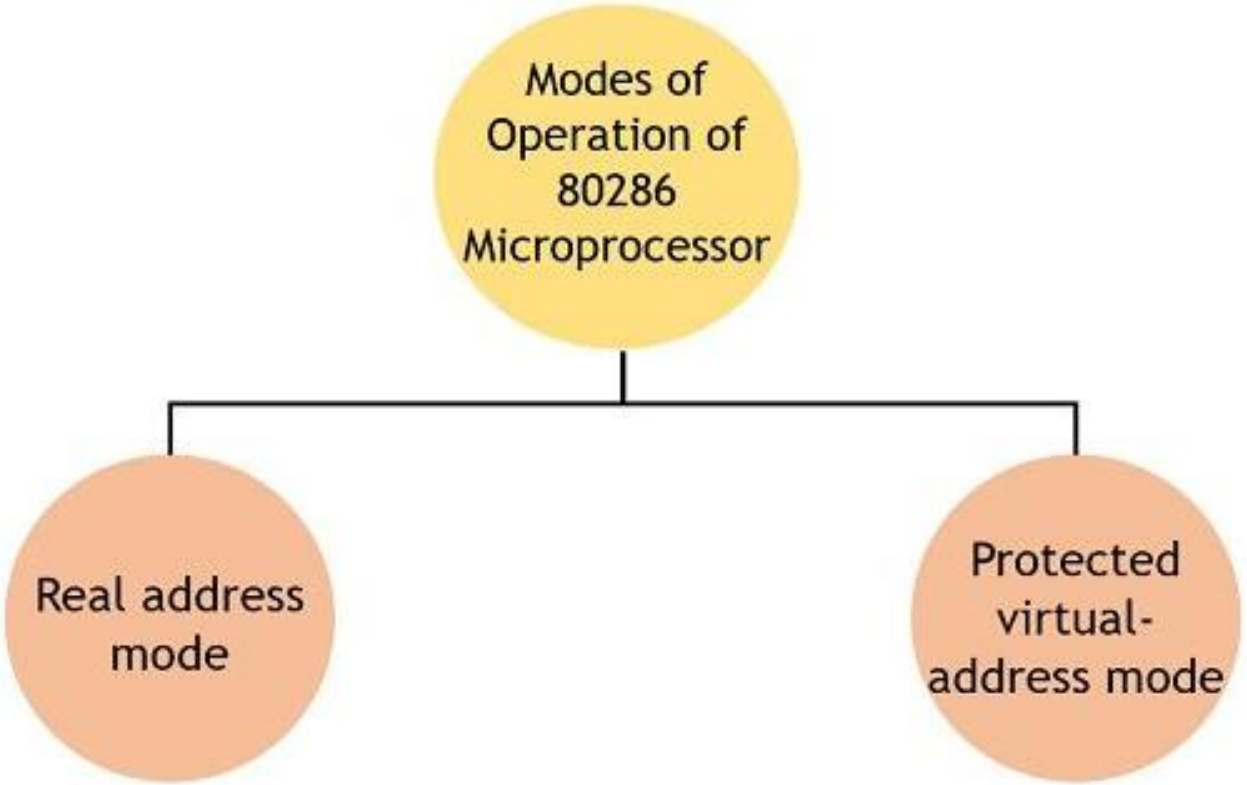
It has non-multiplexed data and address bus, which increases the operating speed of the system.

The size of data bus is 16-bit whereas the size of address bus is 24-bit.

It was invented in February **1982** by **Intel**

The performance per clock cycle of 80286 is almost twice when compared with 8086 or 8088

OPERATING MODES OF 80286 MICROPROCESSOR



OPERATING MODES OF 80286 MICROPROCESSOR

In **real address mode**, this microprocessor acts as a version of 8086 which is quite faster.

Also without any special modification, the instruction programmed for 8086 can be executed in 80286.

It offers memory addressability of 1 MB of physical memory.

OPERATING MODES OF 80286 MICROPROCESSOR

The **protected virtual-address mode** of 80286 supports multitasking because multiple programs can be executed using virtual memory.

This mode of 80286 offers memory addressability of 16 MB of physical memory along with 1 GB of virtual memory.

As using virtual memory, space for other programs can be saved

Sometimes bulky programs also do exist that cannot be stored in physical memory, so virtual memory is utilized in order to execute large programs

WHAT IS VIRTUAL MEMORY?

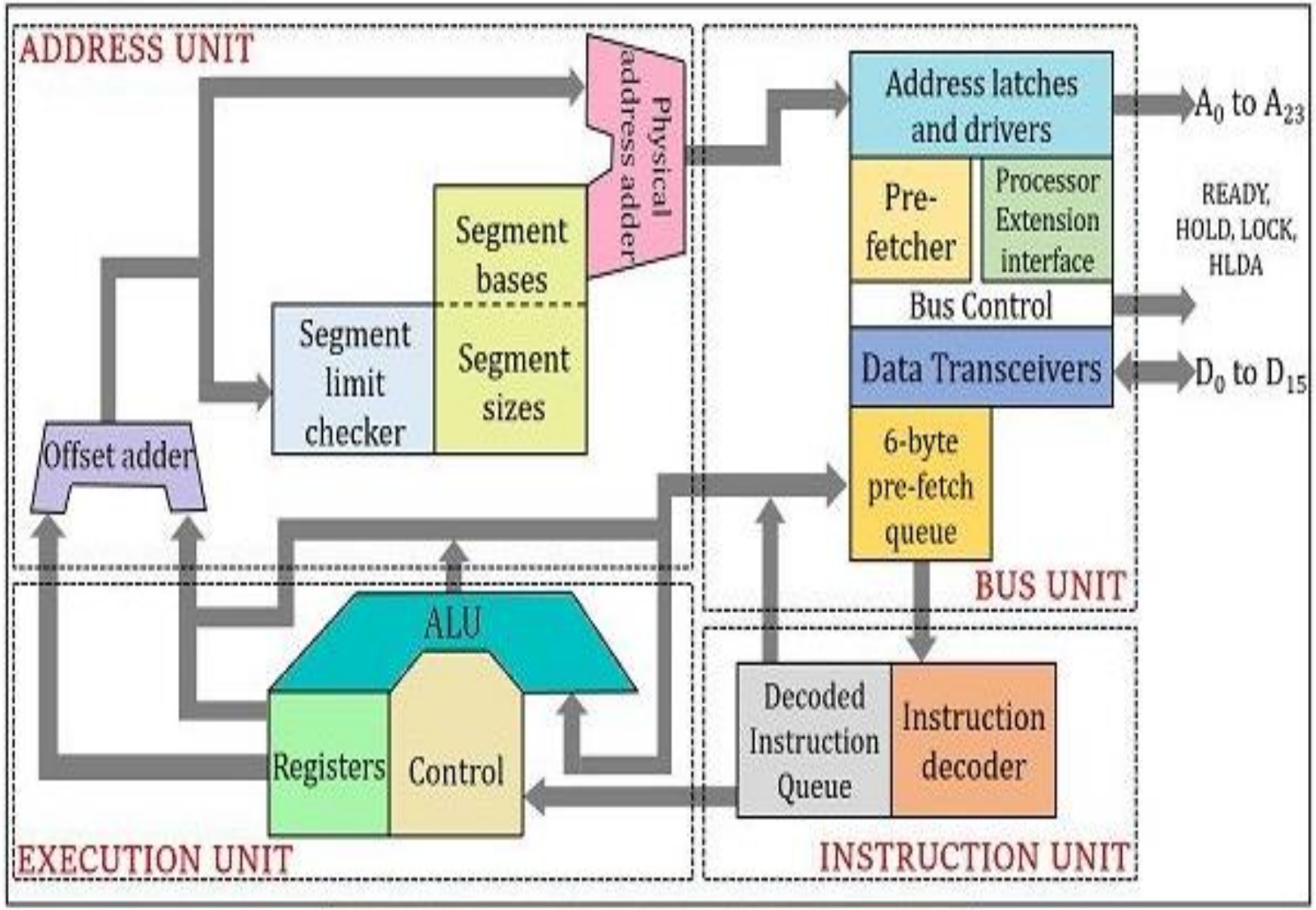
Virtual memory is that part of hard disk which can be utilized for storing large instructions inside the system.

This extra memory can be addressed by the computer other than the physical memory.

When there exists an instruction that is to be loaded in the memory but whose size is greater than the provided physical memory.

Then some part of hard disk is used in order to store that instruction, which is known as virtual memory.

80286 Architecture



Block Diagram of 80286 Microprocessor

The CPU, central processing unit of 80286 microprocessor, consists of 4 functional block:

- Address Unit
- Bus Unit
- Instruction Unit
- Execution Unit

Firstly, the physical address from where the data or instruction is to be fetched is calculated, by the **address unit**.

Once the physical address is calculated then the calculated address is handed over to the bus unit.

More specifically we can say, that the calculated address is loaded on the address bus of the bus unit.

This address specifies the memory location from where the data or instruction is to be fetched.

The fetching of data through the memory is done through the data bus.

For faster execution of instruction, the BU fetches the instructions in advanced from the memory and stores them in the queue.

This is done through the bus control module.

As we have discussed that the prefetched instructions are stored in a **6-byte instruction queue**. This instruction queue then further sends the instruction to the **instruction unit**

The instruction unit on receiving the instructions now starts decoding the instruction.

As instructions are stored in prefetched queue thus the decoder continuously decodes the fetched instructions and stores them into decoded instruction queue

Now after the instructions gets decoded then further these are needed to be executed.

So, the instructions from decoded instruction queue are fed to the **execution unit**.

The main component of EU is ALU i.e., arithmetic and logic unit that performs the arithmetic and logic operations over the operand according to the decoded instruction

Once the execution of the instruction is performed then the result of the operation i.e., the desired data is send to the register bank through the data bus

As we have already discussed that 80286 is just a modified version of 8086. The register set in 80286 is same as that of 8086 microprocessor

Thank You