Chapter 2

The Microprocessor Architecture

prepared by

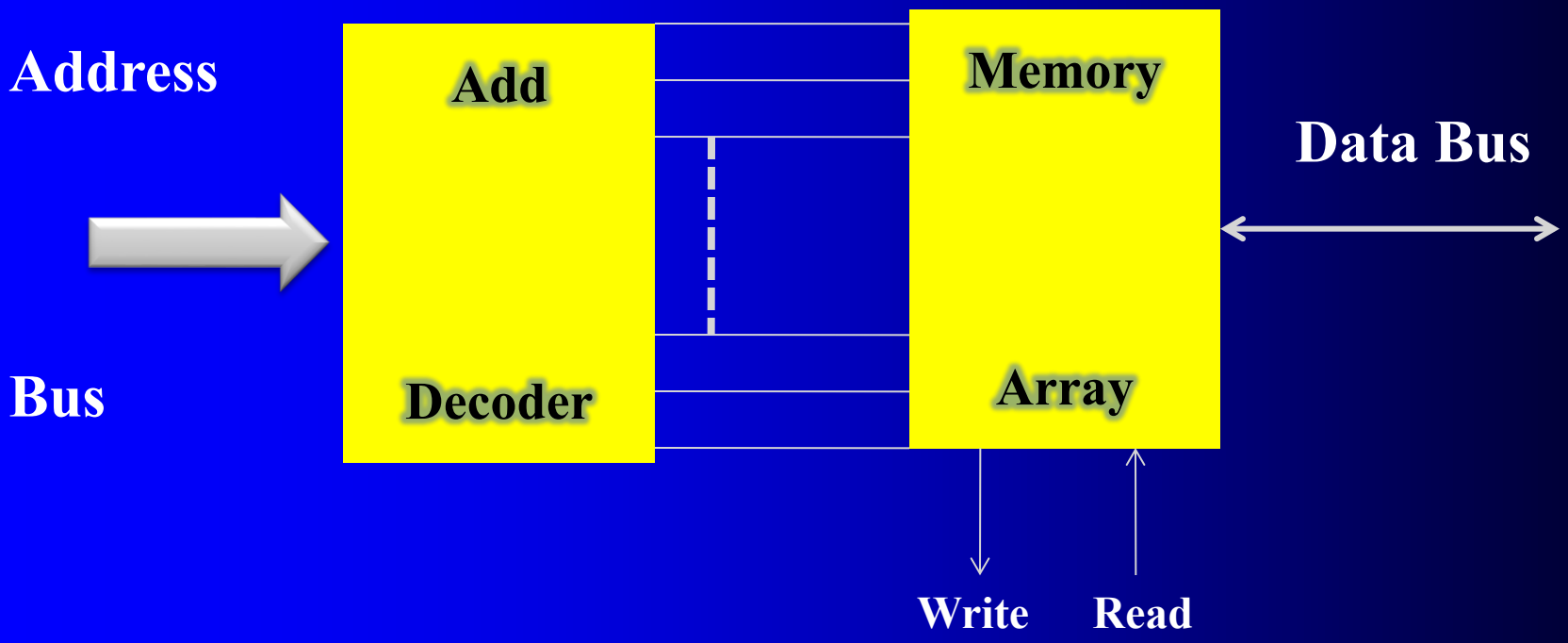
Dr. Mohamed E. NASSAR

Chapter Overview

- Memory Architcture and Basic operation of 8088, 8086
- Internal Microprocessor Architecture
- The Programming Model
- Real Mode Memory Addressing
- Segments and Offsets
- Default Segment and Offset Registers

Memory Architecture and Basic operation of 8088 / 8086

Memory Arch



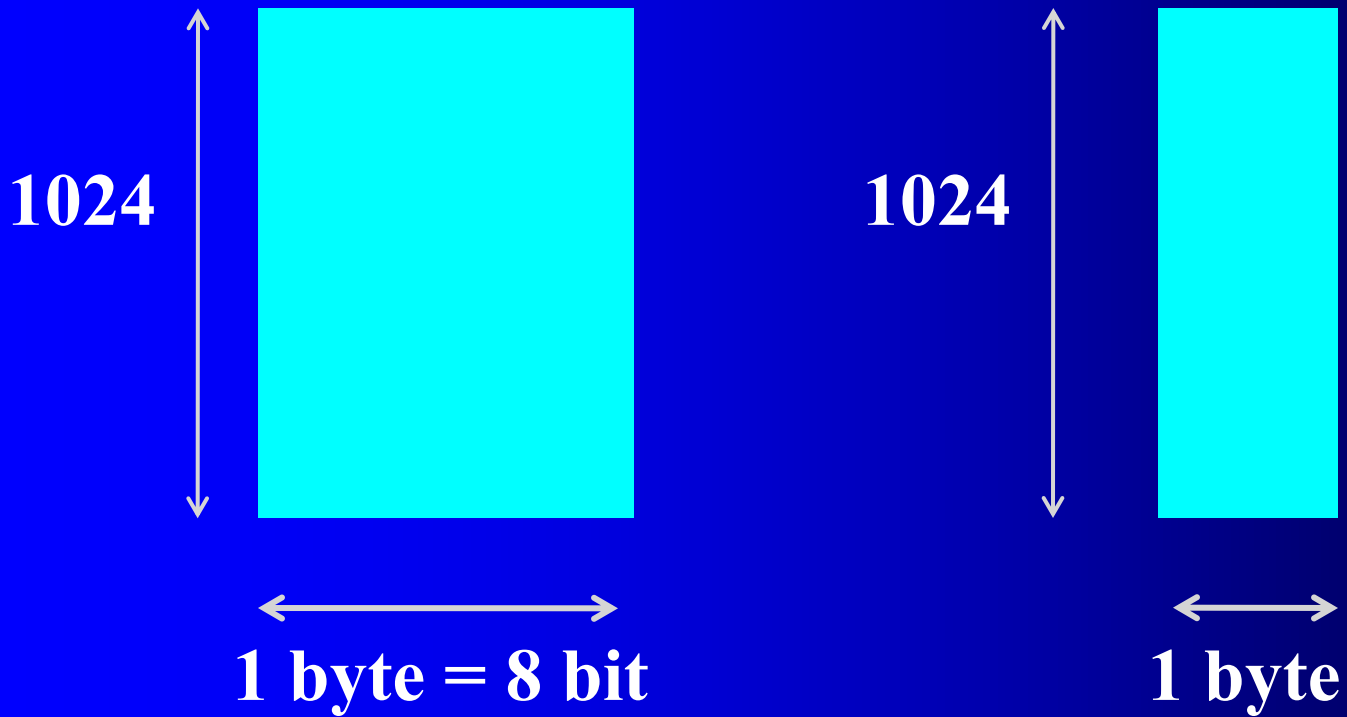
The capacity of a memory is the total n of Data units can be stored.

Ex: size of Memory = 1K byte
it means 2^{10} Location
each Location ⟶ Byte

} 2^{10} bytes

1K byte = 1024 byte
1K Hs = 1000 Hs

1K bit = 1024 bit



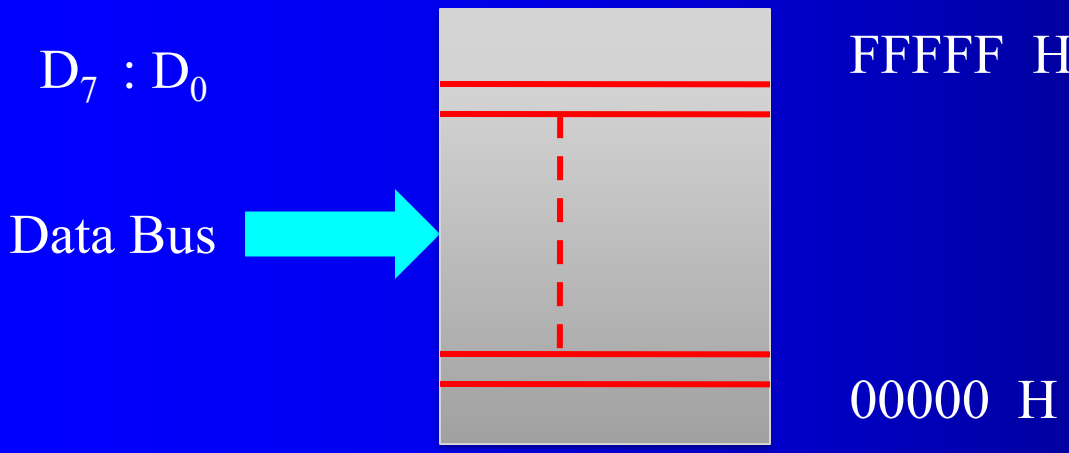
Basic Memory Arch : All MPs Support byte operation.

[1] For 8088 MP

Data Bus = 8 bit

Memory = 1 M byte → 20 line in Add Bus

.. 8 bit .. Only one bank of memory.



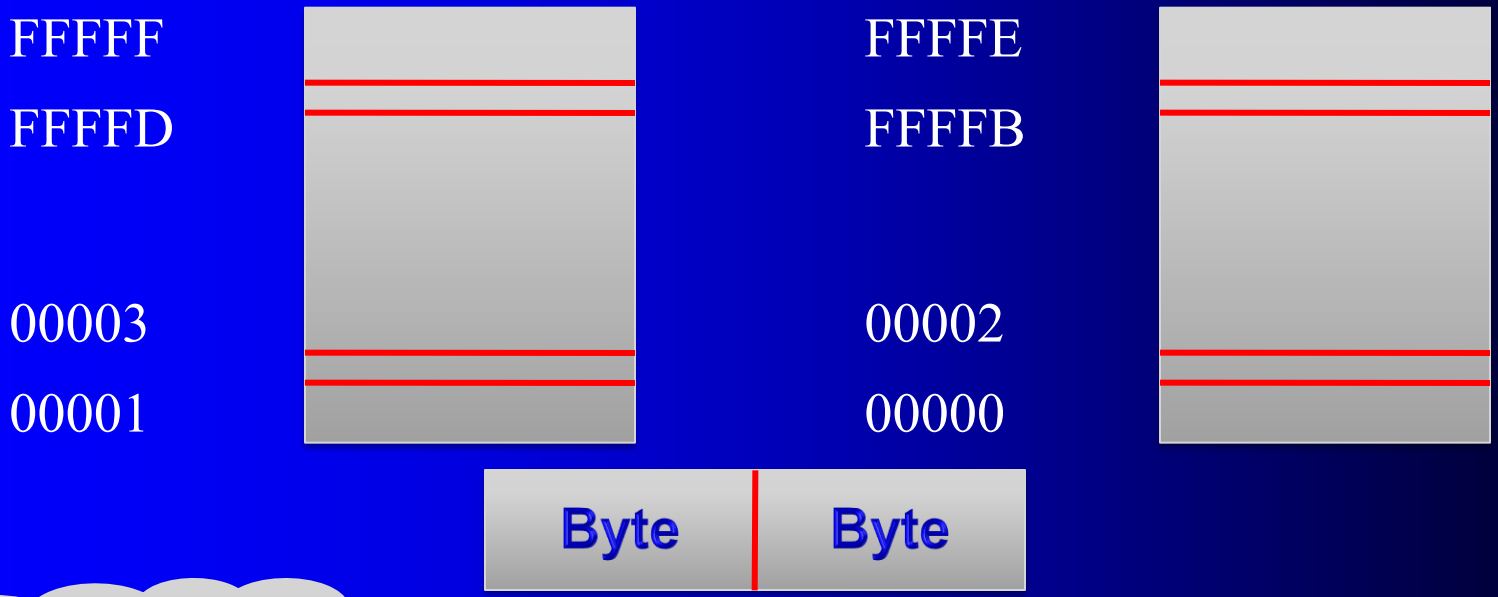
Basic Memory Arch : All MPs Support byte operation.

[1] For 8086 MP

Data Bus = 16 bit

It can fetch 16 bit in one instruction But it must Support Byte operation (we need two Banks)

.. 8 bit .. Only one bank of memory.



Basic operation of 8086/8088 :

- The MP executes a program (List of instruction) by repeatedly cycling From (3) steps:
  1) fetch an instruction from memory.
  2) Decode the instruction (read it)
  3) Execute the instruction (do what it says)
- 8085/8080 didn't use pipeline technique.
  They couldn't fetch on instruction while it is executing another.
- 8088/8086 MP could fetch a new instruction while executing another ⟶ This is called "Pipelining".
- To make the CPU faster, there are two methods:
  1) increase the operating freq.
  2) Change the internal Architecture.

8080/8085 No Pipe lining

CPU	Fetch	Read	Execute	Fetch	Write	Execute
BUS	Busy	Busy	……	Busy	Busy	……

8086/8088 With Pipe lining

EU			Execute		Execute
BIU	Fetch	Read	Fetch	Write	Fetch
BUS	Busy	Busy	Busy	Busy	Busy

In 8088/8086 it Provides two separate internal units:

[1] Execution Unit (EU)

- **EU decode of executes the instruction.**
- **It Contains ALU (Arithmetic Logic Unit) That perform Logic and Arithmetic operations**

[2] Bus Interface unite (BIU):

- BIU interfaces with the bus and fetches instructions, Read operands and writes results.
- BIU contains a buffer called “instruction pre fetch queue” to store instructions waiting for execution.
- This Buffer size is 6 bytes in 8086 (2 byte wide of 3 locations deep)

For 8088

Pre fetch queue buffer is 4 bytes
(1 byte wide of 4 Locations deep)

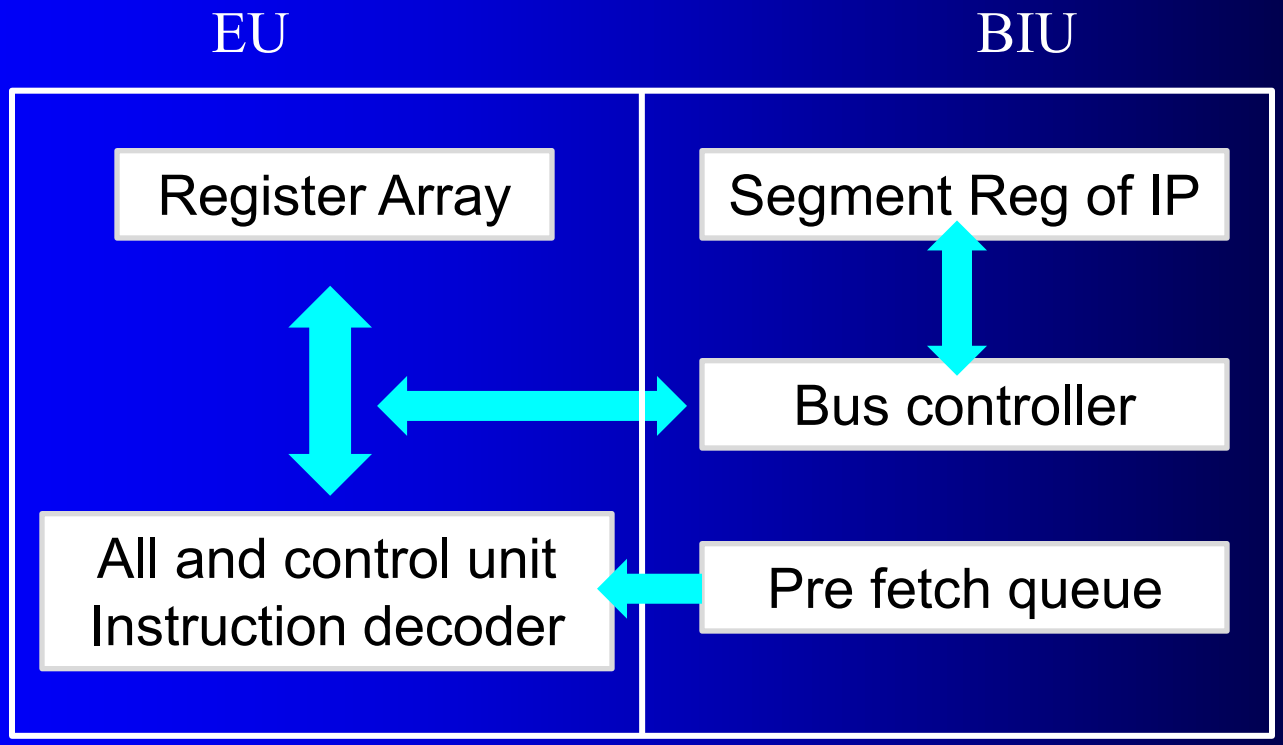
Note

Pre fetch queue is FIFO Memory
(First In First out)

- While EU is executing instructions, BIU Looks ahead and fetch more instructions for memory .
  - This is called “Pip lining” or “Pre fetching”

The main purposed of the BIU are :

1- keep pre fetch queue filled with instructions.
2- generate and accept system control signals.
3- Provide the system with memory address or I/O port number.
4- act as window bet n EU and memory for data.



Internal. Arch. Of 8088/8086

Instruction Decoder

1- Receives the instructions from Pre-fetch queue.

2- Decodes the instrucions to be executed.

Introduction to the INTEL 8086 microprocessor

- The processors 8086 to 80286 contains 16-bit internal architecture , 80386 up to Pentium pro has 32-bit internal architecture.

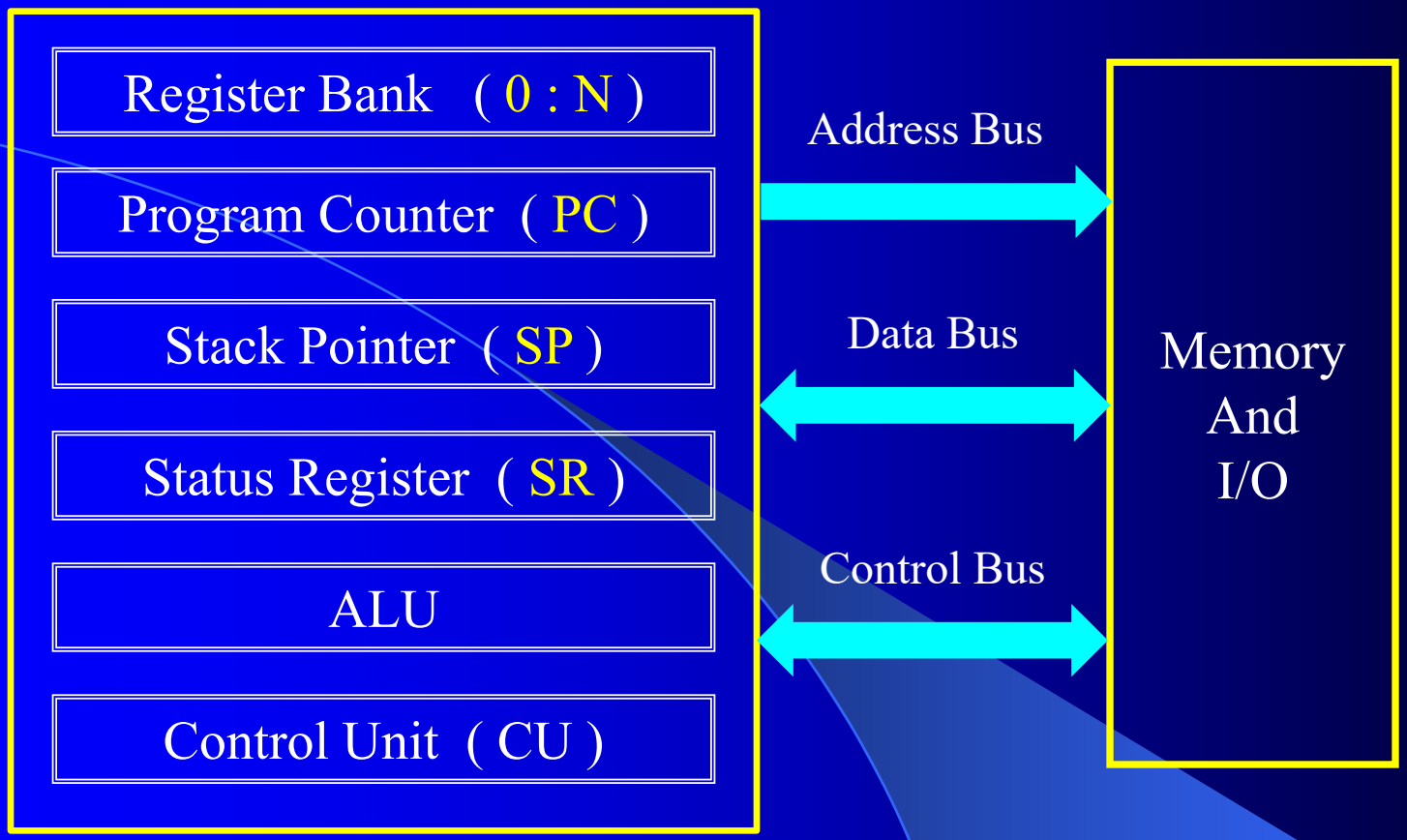
- The word 16-bit means that its arithmetic logical unit, internal registers, and most of its instructions are designed to work with 16-bit binary words.

- The 8086 has a 16-bit data bus, so it can read data form or write data to memory and ports either 16-bits or 8-bits at a time.

- The 8086 has a 20-bit address bus, so it can address any one of 2^{20} or 1,048,576 memory locations.

- Each of the 1,048,576 memory addresses of the 8086 represents a byte-wide location.

Internal Microprocessor Architecture



Conceptual Microprocessor Organization

Internal Microprocessor Architecture

(1) Registers Bank: It is a 16 bit registers store digital data during perform the program instruction

(2) Program Counter: Acts as a counter contain address information about the memory location, that the μP obtains program instructions each time the μP access memory PC to increment memory address.

(3) Stack Pointer : When the μP have to suspend the operation and take up different operation (sub rotation or interrupt) in this case processors stores internal information of present operation in certain part of memory called stack memory. The stack pointer address information in stack memory.

Internal Microprocessor Architecture

(4) Status Register: Contains information status of operation such as, when there was a carry in addition operation, which examined by processor to make proper decision.

(5) Arithmetic Logic Unit :Perform all arithmetic and logic operation such as OR, NOR, AND, NAND, Add, Mul, Sub,…..

(6) Control Unite : Coordinate all the operation of the processor. It contains an instruction request (IR) which decodes and fetched instruction and generate appropriate timing and control signals to process the information .

Internal Microprocessor Architecture

(7) Address Bus :It is unidirectional bus from the microprocessor to the memory or I/O, it contains address information that being put out by the processor

(8) Data Bus: It is bidirectional bus. The information flows between the processor and the external memory or I/O.

(9) Control Bus : It control information between the processor and the memory or I/O such as :

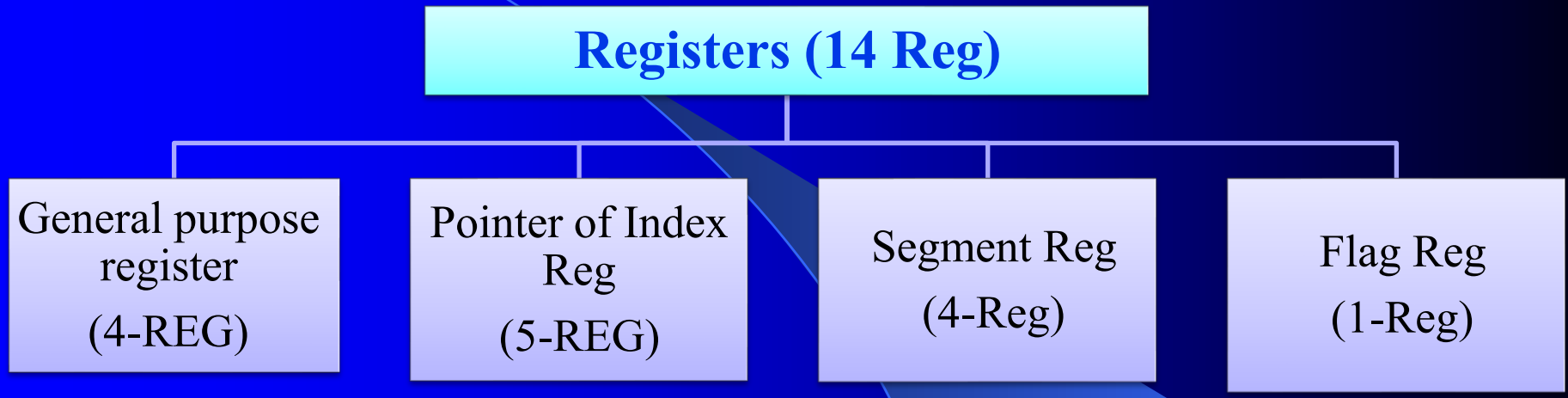
$R/\overline{W}$: Read - Write.

$DT/\overline{R}$: Data transfer-reception

$M/\overline{IO}$: Memory Input-output

Register

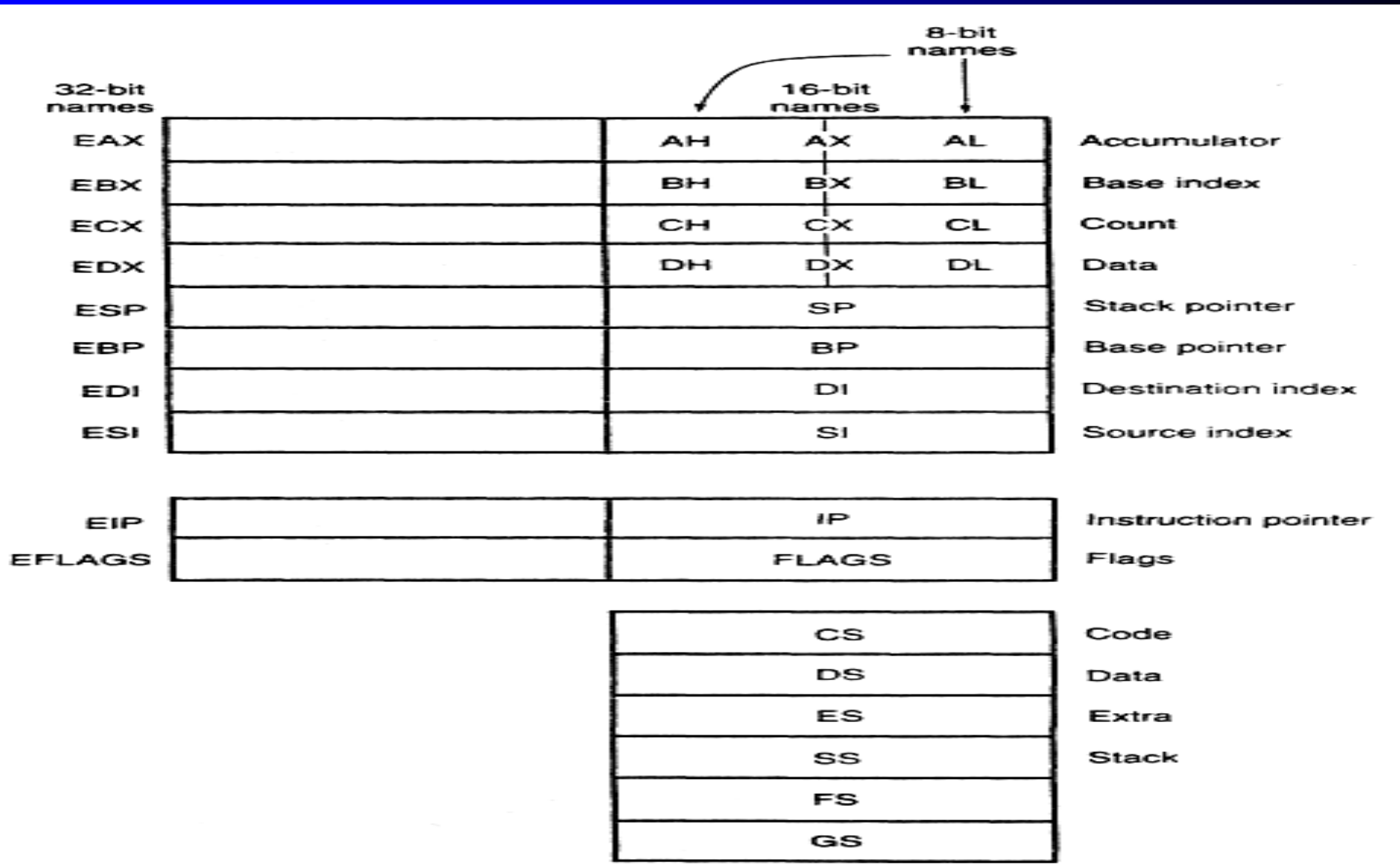
- A register is a digital with two Basic fn:
  - Temporary data storage.
  - Shifting capabilities.
- In CPU register each is **16 bit** wide.



[1] General Purpose Register (4-Reg).

- Each can be used for general purpose or for special f,**n**
- Advantage of Storing data in Reg internally in Mp than storing in Memory is (Reg are much more faster)

Internal Microprocessor Architecture



Multipurpose Registers

These registers hold various data size (bytes, words, or doublewords) and are used for almost any purpose as dictated by a program

AX	AX (accumulator) a 16-bit register (AX), or of two 8-bit registers (AH and AL). The accumulator is used for instructions such as multiplication, division, and some of adjustment instruction, I/O devices and In translation (X LAT).
BX	BX (base index) sometimes holds the offset address of a location in the memory, and used by Translate instruction (X LAT).
CX	CX (count) is a general-purpose register that also holds the counter value in Loop, and string operation.

Multipurpose Registers

DX	DX (data) holds a part of the result from a multiplication or part of the divide end in division and used in I/O addressing used in string operation.
BP	BP (base pointer) points to a memory location for memory data transfers.
DI	DI (destination index) often addresses string destination data for the string instructions.
SI	SI (source index) often addresses source string data for the string instructions.

Special-purpose Registers.

IP	IP (instruction pointer), addresses the next instruction in a section of memory defined as a code segment. This register is IP (16 bits) when the microprocessor operates in the real mode and EIP (32 bits) when the 80386 and above operate in the protected mode.
SP	ESP (stack pointer) addresses an area of memory called the stack memory. The stack memory stores data through this pointer .
FLAGS	FLAGS indicate the condition of the microprocessor and control its operation. Next figure shows the flag registers of 8086 microprocessor.

8086 Flag Registers Format.

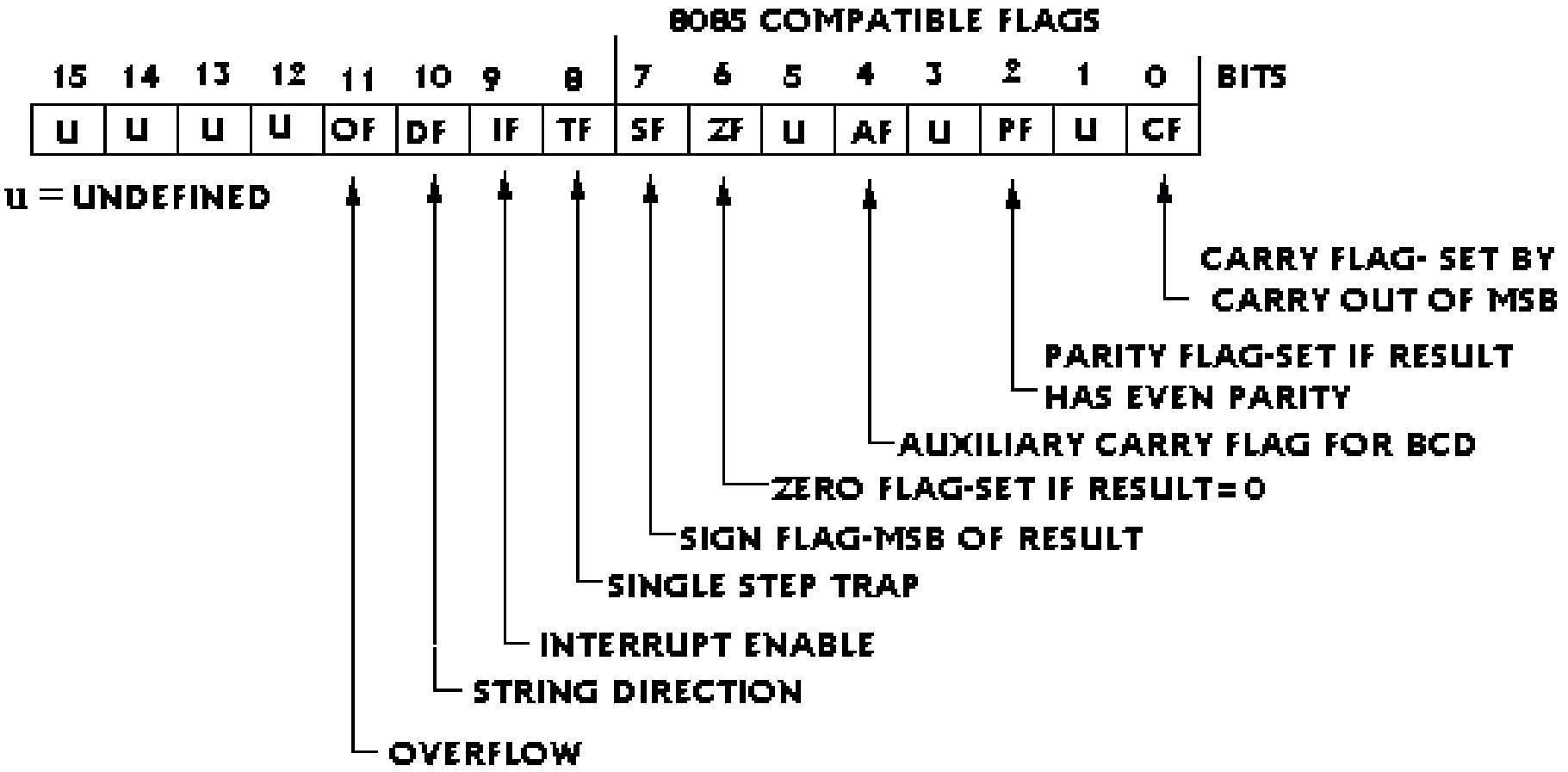
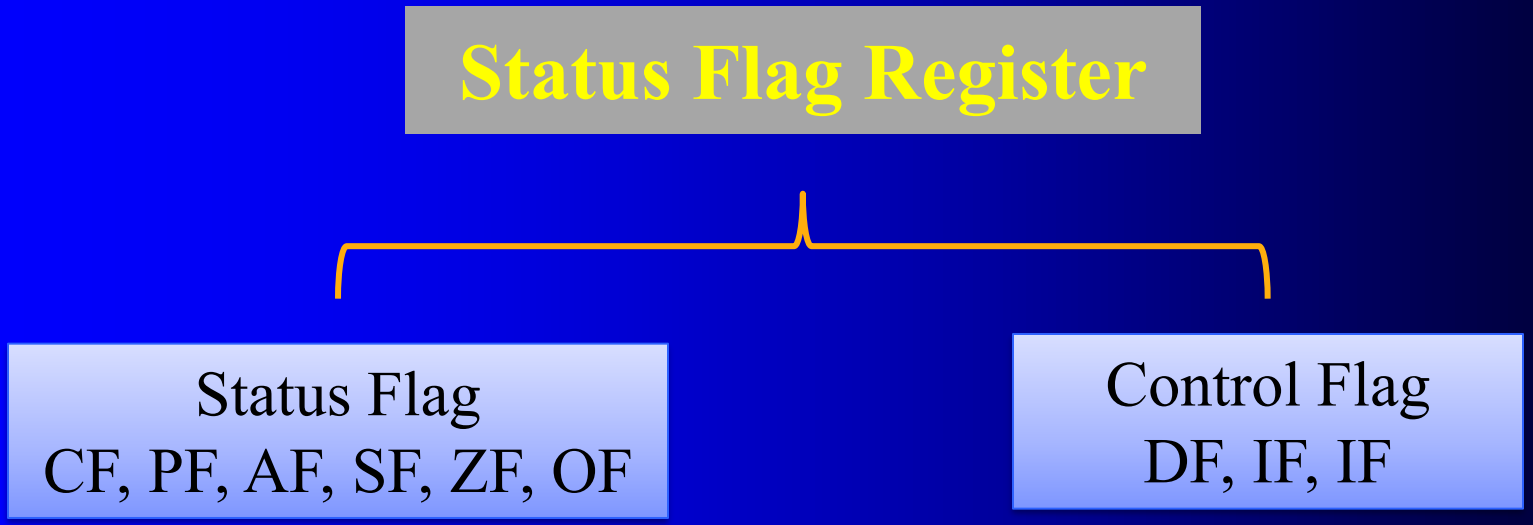


Fig.5 8086 flag register format


Status Flag Register
Status Flag
CF, PF, AF, SF, ZF, OF
Control Flag
DF, IF, IF

8086 Flag Registers

C (carry)	**Carry** holds the carry after addition or the borrow after subtraction.
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. For example : if a number contains three binary one bits, it has odd parity.
A Auxiliary Carry	**Auxiliary carry** holds the carry (half-carry) after addition or the borrow after subtraction between bits.
Z (zero)	**The zero flag** shows that the result of an arithmetic or logic operation is zero.

8086 Flag Registers

S (sign)	The sign flag holds the arithmetic sign of the result after an arithmetic or logic instruction executes. If S=1 the sign bit is set or negative, If S=0 the sign bit is cleared or positive.
T(trap)	The trap flag enables trapping through an on-chip debugging feature. If the T flag is enabled 1, the microprocessor interrupts the flow of the program on condition as indicated by the debug registers and control registers. If the T flag is a logic 0, the trapping feature is disabled and it can call central flag.
I (interrupt)	The interrupt flag controls the operation of the INTR (interrupt request) input pin. If I=1, the INTR pin is enabled, if I=0, the INTR pin is disabled. The state of the I flag bit is controlled by the STI (set I flag) and CLI (clear I flag) instructions and it can call central flag.

8086 Flag Registers

D(direction)	The direction flag selects either the increment or decrement mode and it call the central flag D = 0 auto increment the address. D = 1 auto decrement the address.
O(overflow)	Overflow occur when signed numbers are added or subtracted

Segment Registers

CS	The CS (code segment) is a section of memory that holds the code (programs and procedures) used by the microprocessor. The code segment register defines the starting address of the section of memory holding-code.
DS	The DS (data segment) is a section of memory that contains most data used by a program. Data are accessed in the data segment by an offset address or the contents of other registers that hold the offset address.
ES	The ES (extra segment) is an additional data segment that is used by some of the string instructions to hold destination data.
SS	The SS (stack segment) defines the area of memory used for the stack. The stack entry point is determined by the stack segment and stack pointer registers.
FS-GS	The FS and GS segments are supplemental segment registers to allow two additional memory segments for access by programs.

Real Mode Memory Addressing

- The 80286 and above operate in either the real or protected mode.
- Only the 8086 and 8088 operate exclusively in the real mode.
- **Real mode operation** allows the microprocessor to address only the first 1M byte of memory space even if it is the Pentium 4 microprocessor.
- Note that the first 1M byte of memory is called either **the real memory or conventional memory** system.
- The DOS operating system requires the microprocessor to operate in the real mode.
- Real mode operation allows application software written for the 8086/8088, which contains only 1M byte of memory, to function in the 80286 and above without changing the software.