Digital Logic & Computer Architecture

Course Code	Course Name	Teaching scheme		
		Theory	Pract.	Tut.
CSC405	Microprocessor (MP)	03	02	—

Course Objectives & Outcomes

Pre-requisite: Digital Logic and Computer Architecture

Course Objectives	• To equip students with the fundamental knowledge and basic technical competence in the field of Microprocessors. • To emphasize on instruction set and logic to build assembly language programs • To prepare students for higher processor architectures and embedded systems

Course Objectives & Outcomes

Course Outcomes	A Learner will be able to Describe core concepts of 8086 microprocessor Interpret the instructions of 8086 and write assembly and Mixed language programs Identify the specifications of peripheral chip. Design 8086 based system using memory and peripheral chips. Appraise the architecture of advanced processors Understand hyperthreading technology

Syllabus

Module	Contents	Hours
1.	**The Intel Microprocessors 8086 Architecture** 8086CPU Architecture Programmer's Model Functional Pin Diagram Memory Segmentation Banking in 8086 Demultiplexing of Address/Data bus Functioning of 8086 in Minimum mode and Maximum mode Timing diagrams for Read and Write operations in minimum and maximum modeInterrupt structure and its servicing	08

Syllabus

Module	Contents	Hours
2.	**Instruction Set and Programming** Addressing Modes Instruction set-Data Transfer Instructions, String Instructions, Logical Instructions, Arithmetic Instructions, Transfer of Control Instructions, Processor Control Instructions Assembler Directives and Assembly Language Programming, Macros, Procedures	06

Syllabus

Module	Contents	Hours
3.	**Memory and Peripherals interfacing** Memory Interfacing - RAM and ROM Decoding Techniques – Partial and Absolute 8255-PPI-Block diagram, CWR, operating modes, interfacing with 8086. 8257-DMAC-Block diagram, DMA operations and transfer modes. Programmable Interrupt Controller 8259-Block Diagram, Interfacing the 8259 in single and cascaded mode	08

Syllabus

Module	Contents	Hours
4.	**Intel 80386DX Processor** Architecture of 80386 microprocessor 80386 registers–General purpose Registers, EFLAGS and Control registers Real mode, Protected mode, virtual 8086 mode 80386 memory management in Protected Mode – Descriptors and selectors, descriptor tables, the memory paging mechanism	07

Syllabus

Module	Contents	Hours
5.	**Pentium Processor** Pentium Architecture Superscalar Operation, Integer &Floating-Point Pipeline Stages, Branch Prediction Logic, Cache Organization and MESI protocol	07

Syllabus

Module	Contents	Hours
6.	**Pentium 4** Comparative study of 8086, 80386, Pentium I, Pentium II and Pentium III Pentium 4: Net burst micro architecture. Instruction translation look aside buffer and branch prediction Hyper threading technology and its use in Pentium 4	04

Assessment

Examination scheme					
Int. Assessment			End Sem	TW	Oral
PT1	PT2	Avg			
20	20	20	80	25	

Bharati Ingale (bharati_ingale@yahoo.com)

Textbooks

1. John Uffenbeck, “8086/8088 family: Design Programming and Interfacing”, PHI.
2. Yu-Cheng Liu, Glenn A. Gibson, “Microcomputer System: The 8086/8088 Family, Architecture, Programming and Design”, Prentice Hall
3. Walter A. Triebel, “The 80386DX Microprocessor: hardware, Software and Interfacing”, Prentice Hall
4. Tom Shanley and Don Anderson, “Pentium Processor System Architecture”, Addison- Wesley.
5. K. M. Bhurchandani and A. K. Ray, “Advanced Microprocessors and Peripherals”, McGraw Hill

References

1. Barry B. Brey, “Intel Microprocessors”, 8thEdition, Pearson Education India
2. Douglas Hall, “Microprocessor and Interfacing”, Tata McGraw Hill.
3. Intel Manual
4. Peter Abel, “IBM PC Assembly language and Programming”, 5th Edition, PHI
5. James Antonakons, “The Pentium Microprocessor”, Pearson Education

Study: Useful Links

1. https://swayam.gov.in/nd1_noc20_ee11/preview
2. https://nptel.ac.in/courses/108/105/108105102/
3. https://www.classcentral.com/course/swayam-microprocessors-and-microcontrollers-9894
4. https://www.mooc-list.com/tags/microprocessors

Introduction to Microprocessor

- Intel introduced microprocessors in 1969
  - 4-bit microprocessor 4004
  - 8-bit microprocessors
    - 8080
    - 8085
  - 16-bit processors
    - 8086 introduced in 1979
      - 20-bit address bus, 16-bit data bus
    - 8088 is a less expensive version
      - Uses 8-bit data bus
    - Can address up to 4 segments of 64 KB
    - Referred to as the real mode

Bharati Ingale (bharati_ingale@yahoo.com)

Introduction to Microprocessor

- 80186
  - A faster version of 8086
  - 16-bit data bus and 20-bit address bus
  - Improved instruction set
- 80286 was introduced in 1982
  - 24-bit address bus
  - 16 MB address space
  - Enhanced with memory protection capabilities
  - Introduced protected mode
    - Segmentation in protected mode is different from the real mode
  - Backwards compatible

Introduction to Microprocessor

- 80386 was introduced 1985
  - First 32-bit processor
  - 32-bit data bus and 32-bit address bus
  - 4 GB address space
  - Segmentation can be turned off (flat model)
  - Introduced paging
- 80486 was introduced 1989
  - Improved version of 386
  - Combined coprocessor functions for performing floating-point arithmetic
  - Added parallel execution capability to instruction decode and execution units
    - Achieves scalar execution of 1 instruction/clock
  - Later versions introduced energy savings for laptops

Bharati Ingale (bharati_ingale@yahoo.com)

Introduction to Microprocessor

- Pentium (80586) was introduced in 1993
  - Similar to 486 but with 64-bit data bus
  - Wider internal datapaths
    - 128- and 256-bit wide
  - Added second execution pipeline
    - Superscalar performance
    - Two instructions/clock
  - Doubled on-chip L1 cache
    - 8 KB data
    - 8 KB instruction
  - Added branch prediction

Bharati Ingale (bharati_ingale@yahoo.com)

Introduction to Microprocessor

- Pentium Pro was introduced in 1995
  - Three-way superscalar
    - 3 instructions/clock
  - 36-bit address bus
    - 64 GB address space
  - Introduced dynamic execution
    - Out-of-order execution
    - Speculative execution
  - In addition to the L1 cache
    - Has 256 KB L2 cache

Bharati Ingale (bharati_ingale@yahoo.com)

Introduction to Microprocessor

- Pentium II was introduced in 1997
  - Introduced multimedia (MMX) instructions
  - Doubled on-chip L1 cache
    - 16 KB data
    - 16 KB instruction
  - Introduced comprehensive power management features
    - Sleep
    - Deep sleep
  - In addition to the L1 cache
    - Has 256 KB L2 cache
- Pentium III, Pentium IV,…

Bharati Ingale (bharati_ingale@yahoo.com)

Introduction to Microprocessor

- Itanium processor
  - RISC design
    - Previous designs were CISC
  - 64-bit processor
  - Uses 64-bit address bus
  - 128-bit data bus
  - Introduced several advanced features
    - Speculative execution
    - Predication to eliminate branches
    - Branch prediction

Bharati Ingale (bharati_ingale@yahoo.com)

Module 1

8086 Microprocessor

Architecture

Bharati Ingale (bharati_ingale@yahoo.com)

8086 Microprocessor : Topics

1)Features

2)Architecture: Memory, Internal registers

3)Addressing modes

4)Instruction Sets

5)Programming

Bharati Ingale (bharati_ingale@yahoo.com)

8086- Features

Features of 8086 Microprocessor

Basic Features:

1) 8086 is a **16 bit microprocessor** i.e

- Has 16 bit ALU that can perform 16 bit operation simultaneously.
- Has 16 bit internal registers
- Has 16 bit external data bus.
- It has **16-bit data bus**, so it can read data or write data to memory or I/O ports either 16 bits or 8 bits at a time.

8086- Features

2) 8086 comes with different versions.

Sr No	Versions of 8086 µP	Working Frequency
1	8086	5 MHz
2	8086-2	8 MHz
3	8086-1	10 MHz

8086- Features

3) It has **20 bit address line.**

- It can access memory up to 2^{20} memory locations

  i.e. 1048576 = 1Mbytes of memory .

  The address range is 00000 to FFFFF

- Words i.e. 16 bit numbers are stored in consecutive memory locations.

4) It can generate 16 bit address lines to access I/O devices

   hence can access 2^{16} = 65536 I/O ports. = 64K I/O locations

8086- Features

5) It has multiplexed address & data bus which reduces the number of pins buts slows down the transfer of data.

6) Operates on +5v supply.

7) It requires a single phase (single line) clock with 33 % duty cycle.(Clock is generated by separate peripheral chip 8284).

8086- Features

Special Features:

1) It is a pipelined processor:i.e it supports pipelining

- Fetching the next instruction while executing the current instruction is called **pipelining.**
- It uses 2 stage pipeling.
- Fetch stage: prefetches upto 6 bytes of instructions stores them in the queue
- Execute stage: executes these instructions.
- It improves the performance of the system i.e the operations are faster. **Higher Throughput (Speed)**

8086- Features

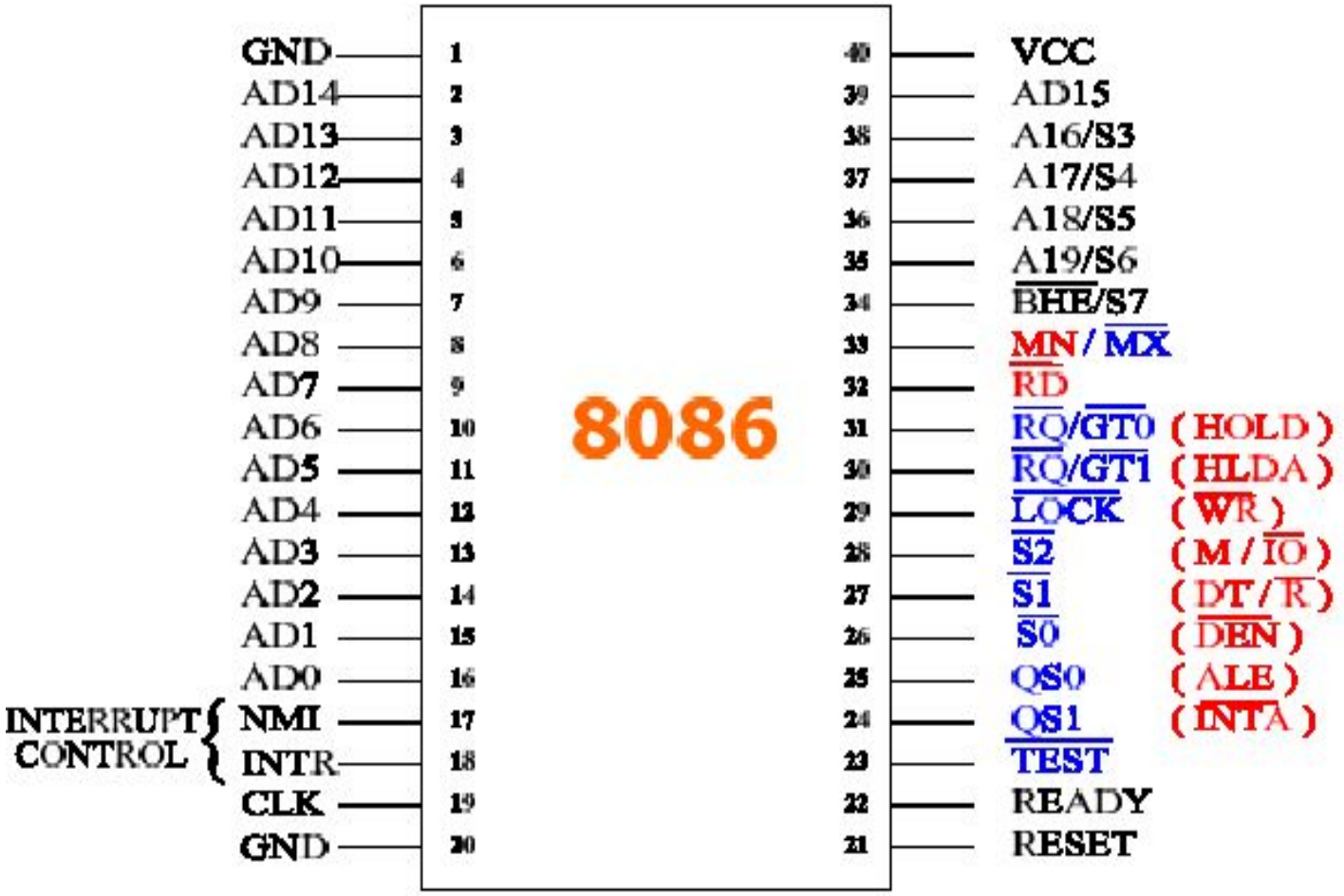
2) It operates in 2 operating modes
   - Minimum mode (single processor) &
   - Maximum Mode (Multi processors like 8087 – NDP etc)
3) It uses Memory Banks- 1MB is divided into 2 banks of 512KB – Lower and Higher Banks
4) It uses memory segmentation

8086- Features

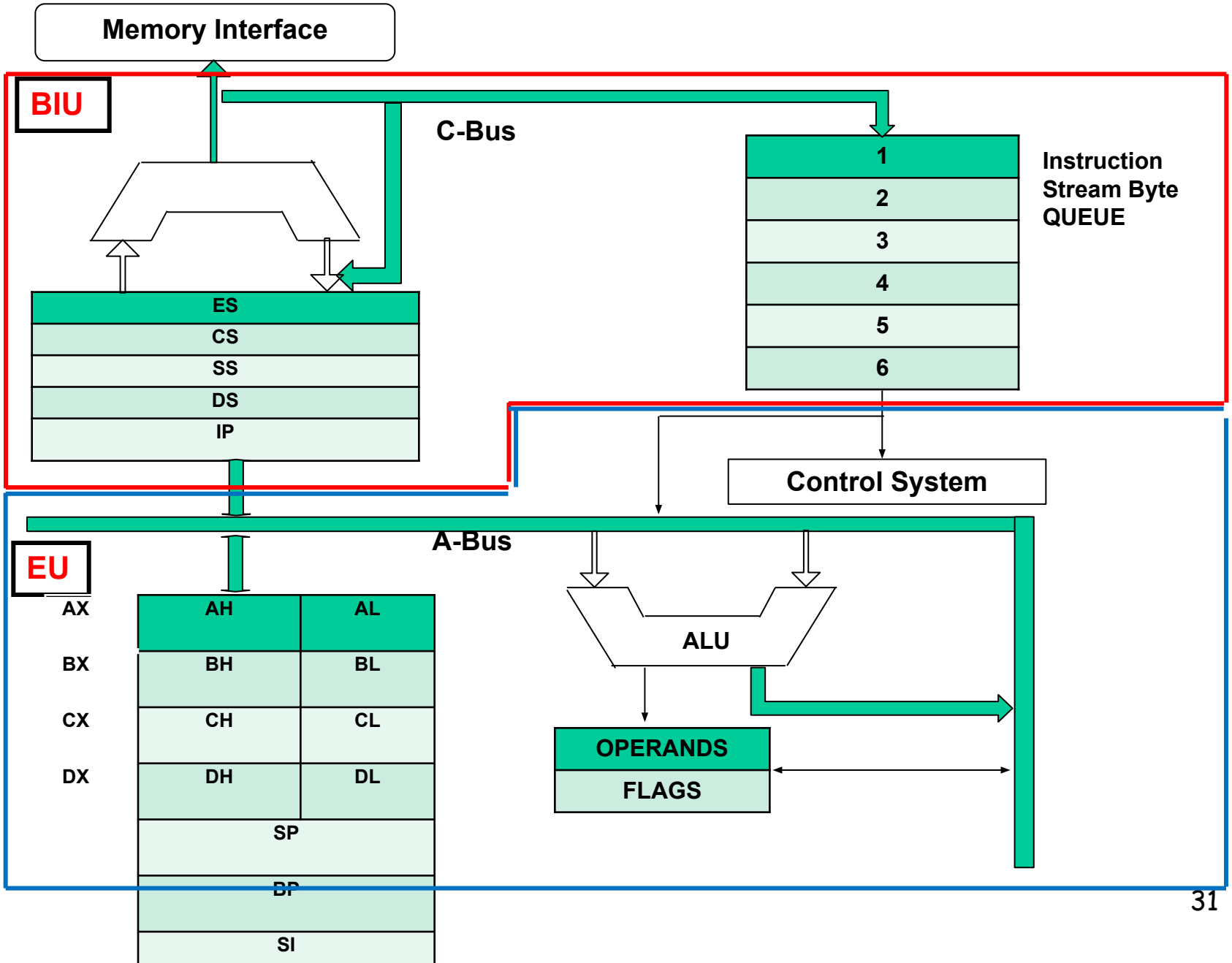
Miscellaneous Features:

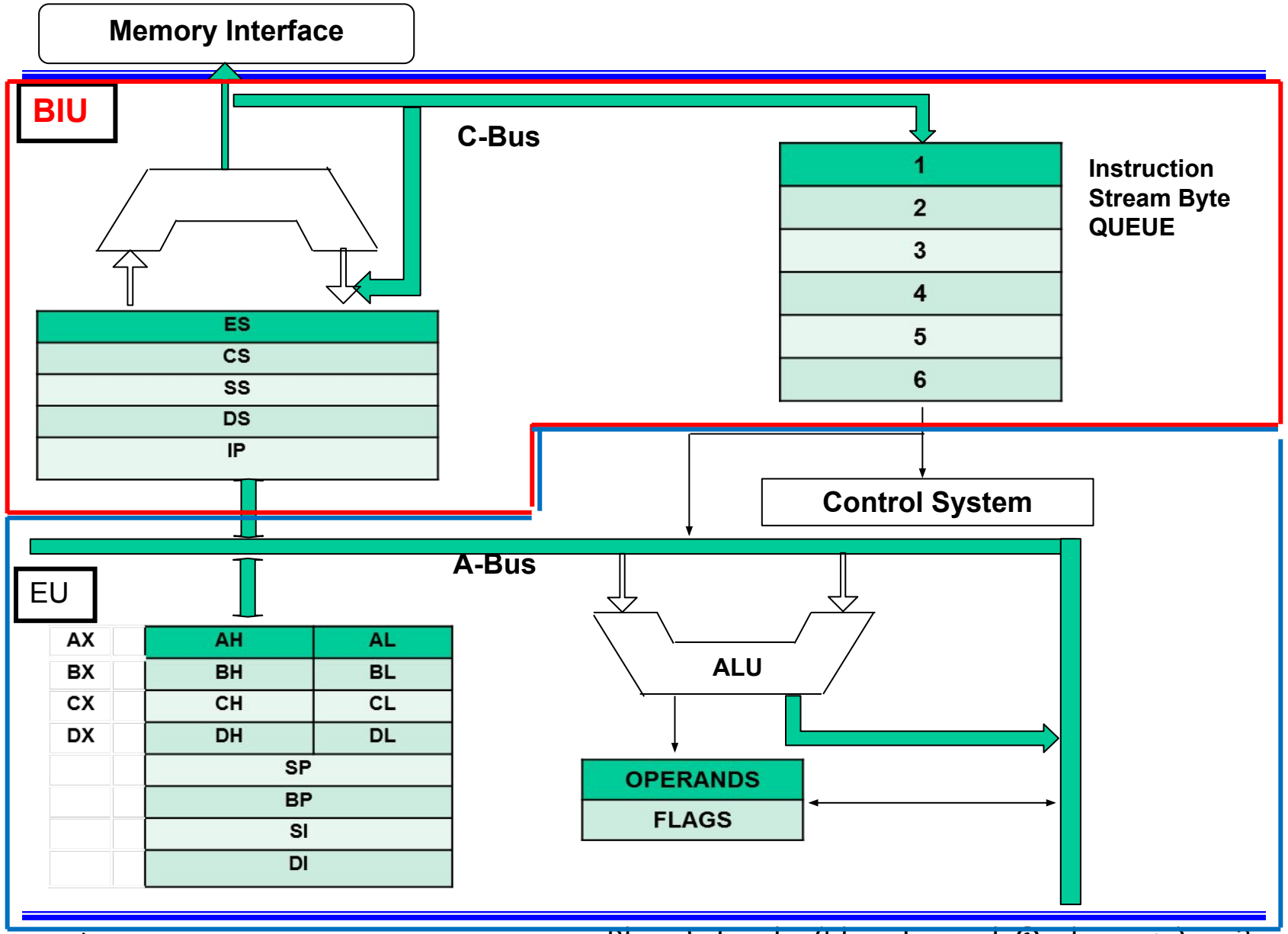
1) It has 256 types of vectored interrupts.
2) It has fourteen 16-bit registers.
3) Instruction set supports Multiply and Divide operations
4) Can perform operations on bit, byte, word or a string

8086 – Pin Diagram

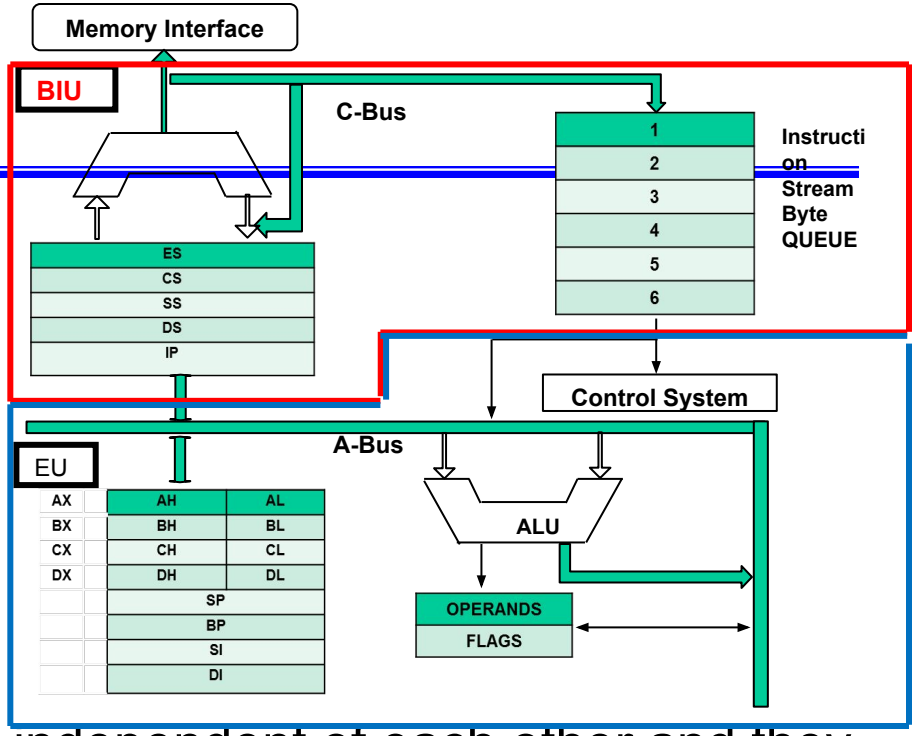


8086 - Architecture




Memory Interface
BIU
C-Bus
1
2
3
4
5
6
Instruction Stream Byte QUEUE
ES
CS
SS
DS
IP
Control System
A-Bus
EU
AX
BX
CX
DX
AH
BH
CH
DH
AL
BL
CL
DL
SP
BP
SI
DI
ALU
OPERANDS
FLAGS

8086 - Architecture



- 8086 has 2 stage μP.
- 2 parts:

i. Bus Interface Unit (BIU)

ii. Execution Unit (EU)

- These 2 units are completely independent of each other and they share the work of CPU
- This type of work division speeds up the processing and reduces the processing time
- This is called as pipeling

Bus Interface Unit(BIU)

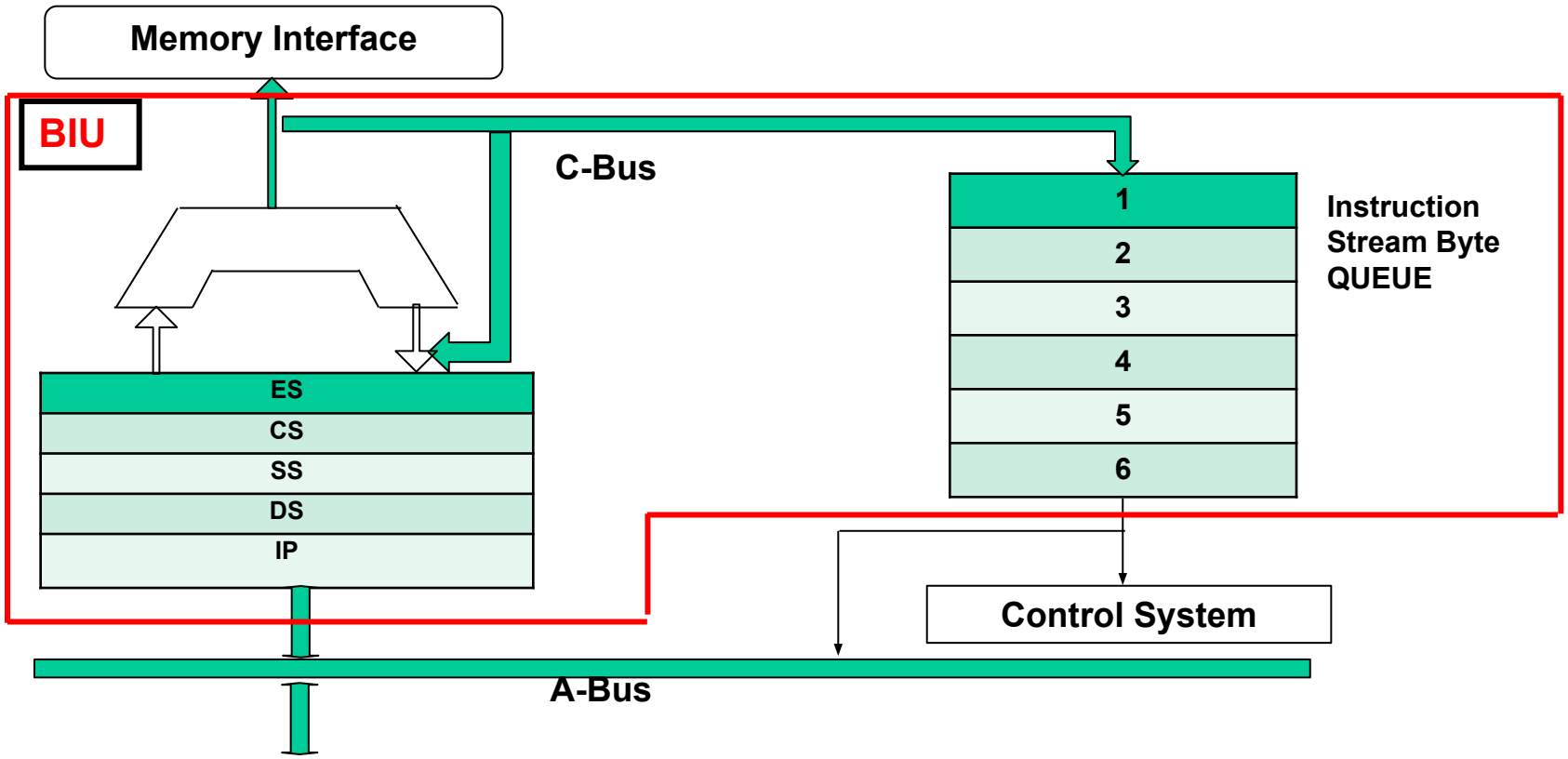
8086- Architecture

Bus Interface Unit(BIU):

- Interface to the outside world.
- Operates w.r.t. to machine cycles.
- Responsible for all ext. operations

8086- Architecture

Bus Interface Unit(BIU):



8086- Architecture

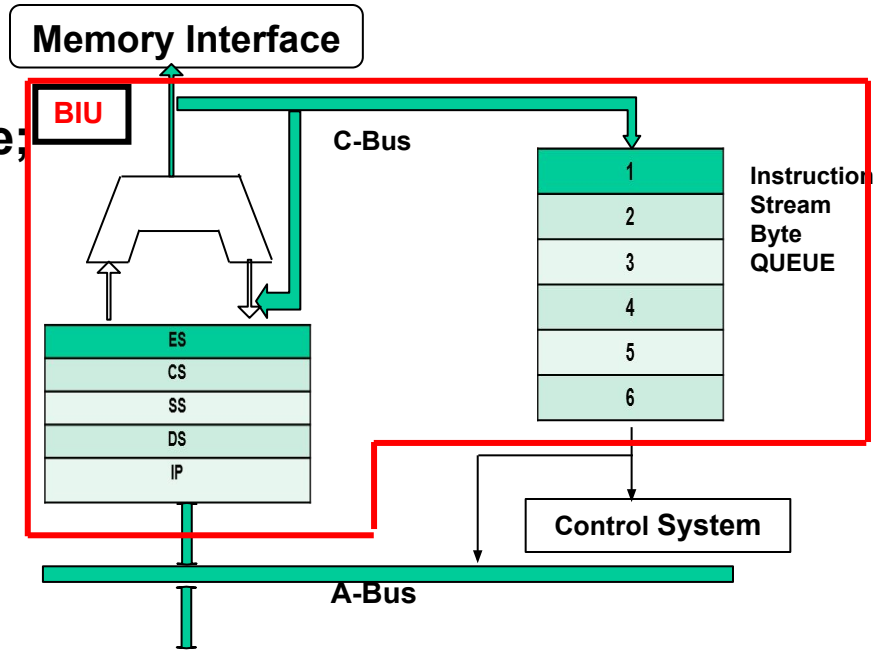
Bus Interface Unit(BIU):

- Performs the following functions

i. Generates 20 bit physical address for mem access
ii. Fetches Instruction from memory and I/O
iii. Transfers data to and from memory and I/O
iv. Supports pipelining using 6 byte instruction Queue

8086- Architecture

Main components of BIU: 3 components



- Segment Registers: **4 special purpose; 16 bit**
- Instruction Pointer (IP reg). **16 bit**
- Instruction Queue: **6 – Byte Pre-Fetch Queue.**
- Address generation and bus control:

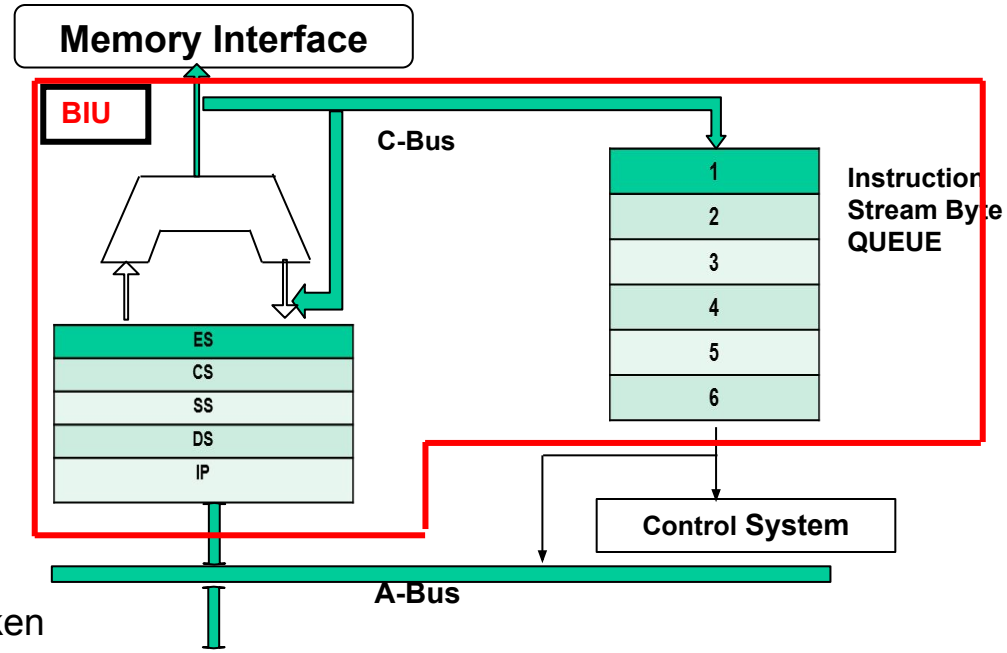
i. The BIU contains a dedicated adder - Used to produce the 20-bit address.

ii. The Bus Control Logic - Generates all the bus control signals such as read and write signals for memory and I/O.

8086- Architecture

Main components of BIU:

- 6 Byte Pre-Fetch Queue
- 6 Byte FIFO RAM used to implement pipelining



Total time taken

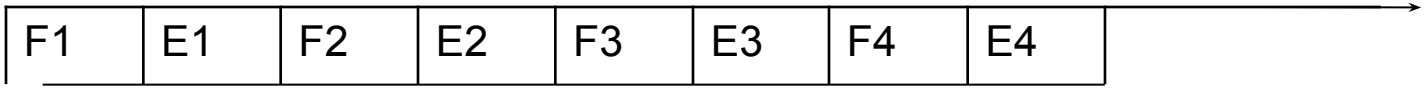
8086- Architecture

Instruction Queue

- EU decodes & executes an instruction
- When EU is decoding & executing an instruction, BIU fetches upto 6 instruction bytes for the next instructions.
- These bytes are called as prefetched bytes and are stored in FIFO reg set called queue

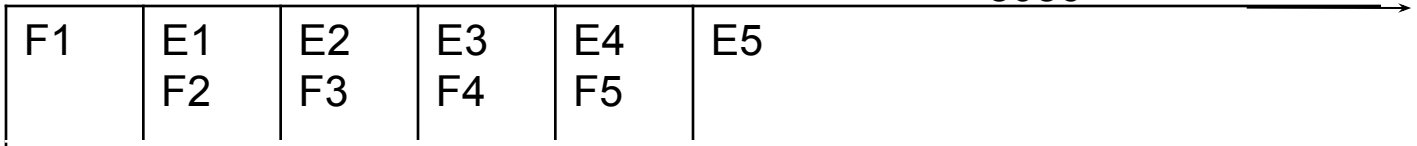
8086- Architecture

Non- Pipelined Processor
eg 8085



Total time taken

Pipelined Processor eg
8086

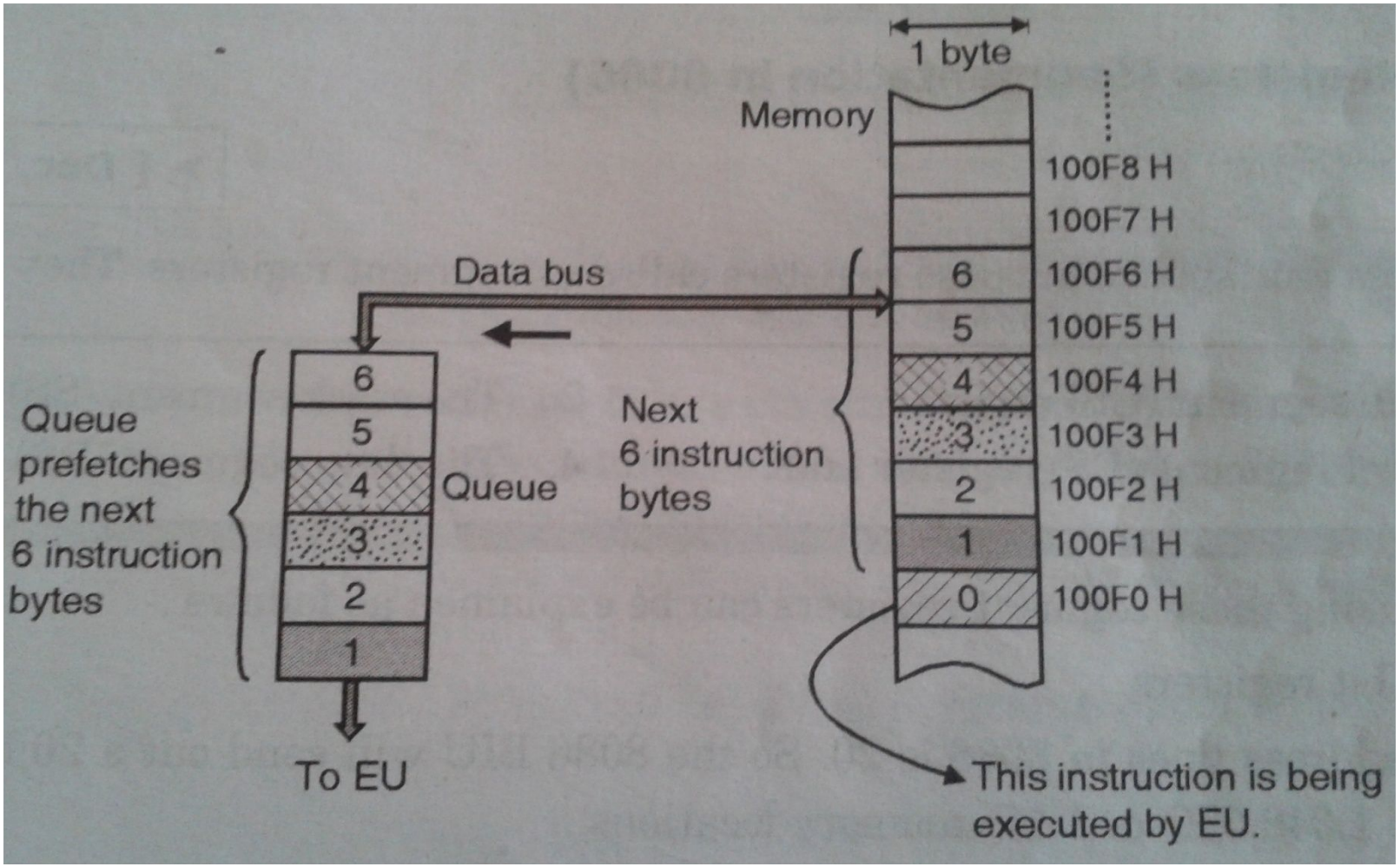


Overlapping Fetching
& Execution

Total time taken

F- FETCH IN BIU
E – EXECUTE IN EU

8086 - PIPELINING



8086- Architecture

Pipelining:

- The process of fetching the next instruction, when the present instruction is being executed is called pipelining
- BIU fills in the queue, untill the entire queue is full
- BIU restarts filing in the queue when atleast 2 locations of queue are vacant.

8086- Architecture

Advantages of pipelining:

- EU reads the next instruction from queue in BIU thus is faster
- EU can execute instructions almost continually instead of having to wait for the BIU to fetch a new instruction.

8086 - PIPELINING

- There are conditions that will cause the EU to enter a "wait" mode.
- JMP or CALL instruction appears in the main program:
  i. Queue is flushed out and queue is made empty
  ii. It is then reloaded with the new instructions from the JMP or CALL location

8086- Architecture

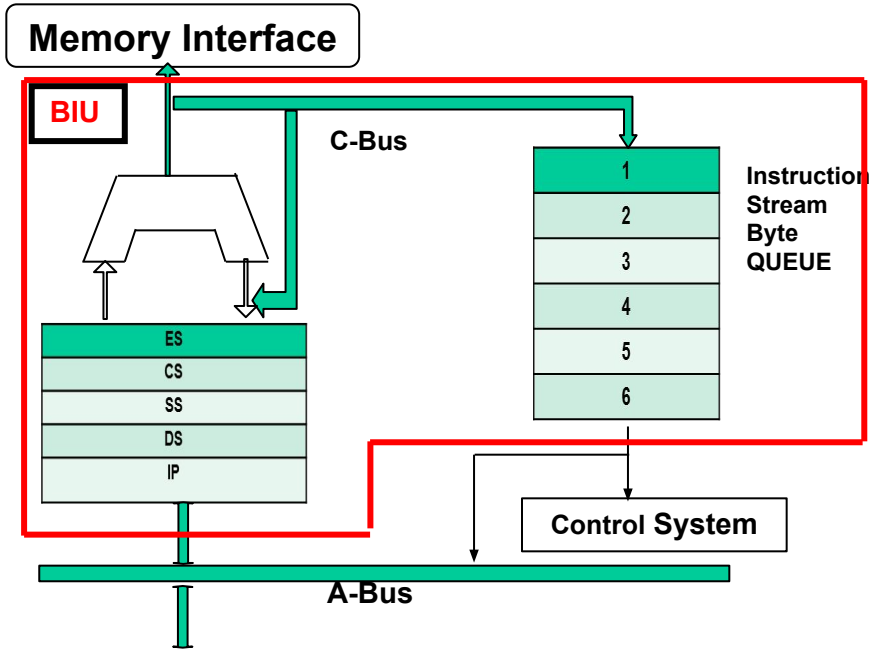
- BIU contains **4 special purpose registers**
- **Segment Registers** hold the upper 16 bits of the base/ starting addr of the 4 memory segments

i. Code Segement (CS) – Program memory

ii. Data Segement (DS) – Data memory

iii. Stack Segement (SS) – Stack memory

iv. Extra Segement (ES) – Extra Segment



8086 - Internal Registers of 8086 (cont..)

Code segment (CS) - 16-bit register

- It contains address of 64 KB segment with processor instructions.
- The processor uses **CS segment** for accessing all instructions referenced by **instruction pointer** (IP) reg.
- CS register cannot be changed directly.
- The CS register is automatically updated during far jump, far call and far return instructions.

8086 - Internal Registers of 8086 (cont..)

Stack segment (SS) - 16-bit register

- It contains address of 64KB segment with program stack.
- By default, all data referenced by **the stack pointer (SP) and base pointer (BP)** registers is located in the stack segment.
- SS register can be changed directly using POP instruction.

8086 - Internal Registers of 8086 (cont..)

Data segment (DS) - 16-bit register

- It contains address of 64KB segment with program data.
- By default, all data referenced **by general purpose registers (AX, BX, CX, DX) and index register (SI, DI)** is located in the data segment.
- DS register can be changed directly using POP and LDS instructions.

8086 - Internal Registers of 8086 (cont..)

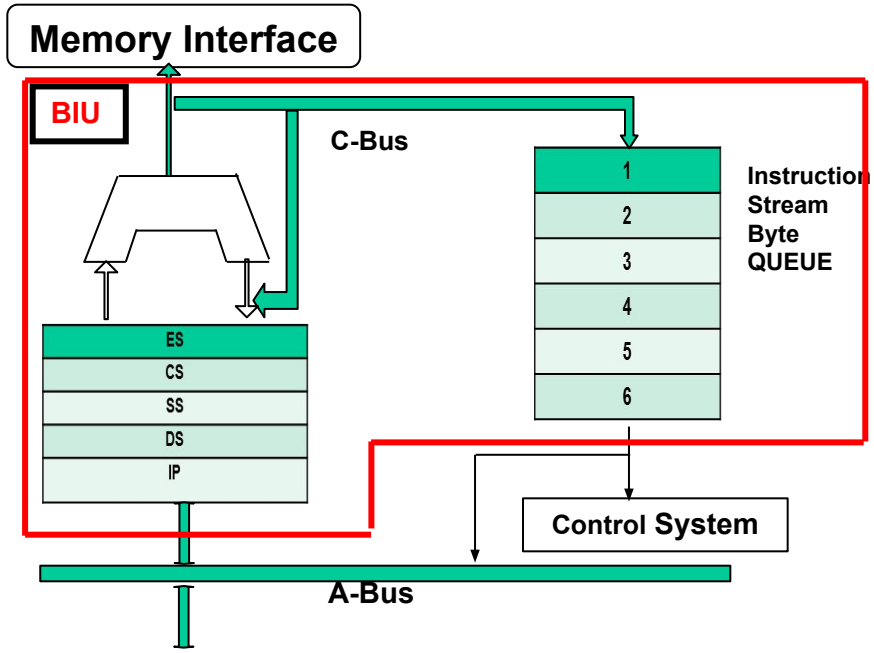
Extra segment (ES) - 16-bit register

- It contains address of 64KB segment, usually with program data.
- By default, the processor assumes that the DI register references the ES segment in string manipulation instructions.
- ES register can be changed directly using POP and LES instructions.

8086- Architecture

- **Instruction Pointer (IP reg). 16 bit**

- Hold the 16 bit addr of the next code byte within the code seg.
- The contents of IP is referred as offset.
- Calculates the addr of the next instruction.
- It is incremented after every instruction.



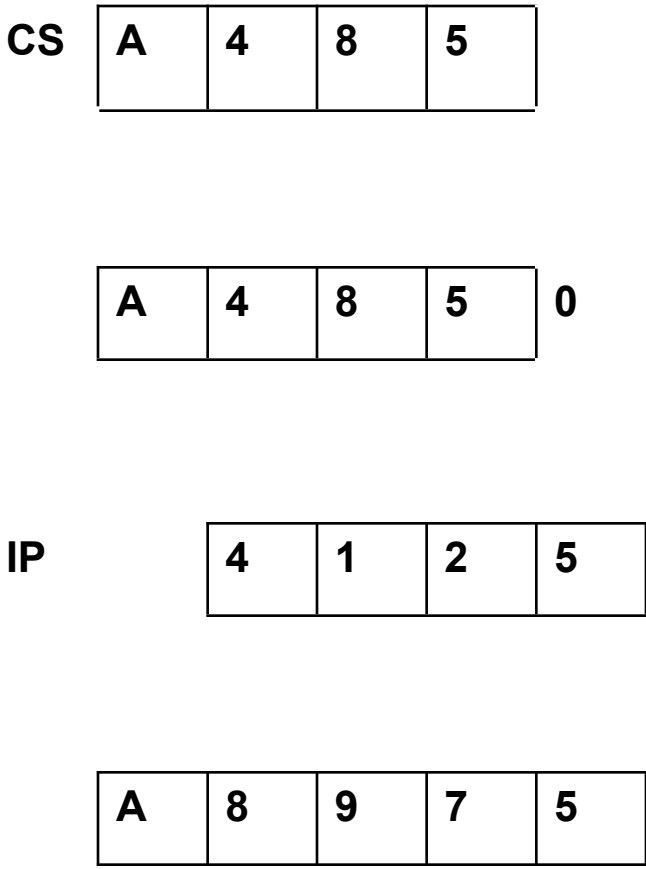
Physical Address Generation

The CS Reg. contains the upper 16 bits of the starting address of the Code segment

BIU will automatically insert "0" for the lowest 4 bits of the segment base address to get 20 bit physical address

The IP reg contains Offset

Add the starting address of code segment (20 bit) to the Offset to get the physical addressof the location containing the next code byte to be fetched



8086- Architecture

Generation of Physical Address

CS	**A**	**4**	**8**	**5**	**0**
IP +		4	1	2	5
Physical Address	A	8	9	7	5

Hardwired Zero

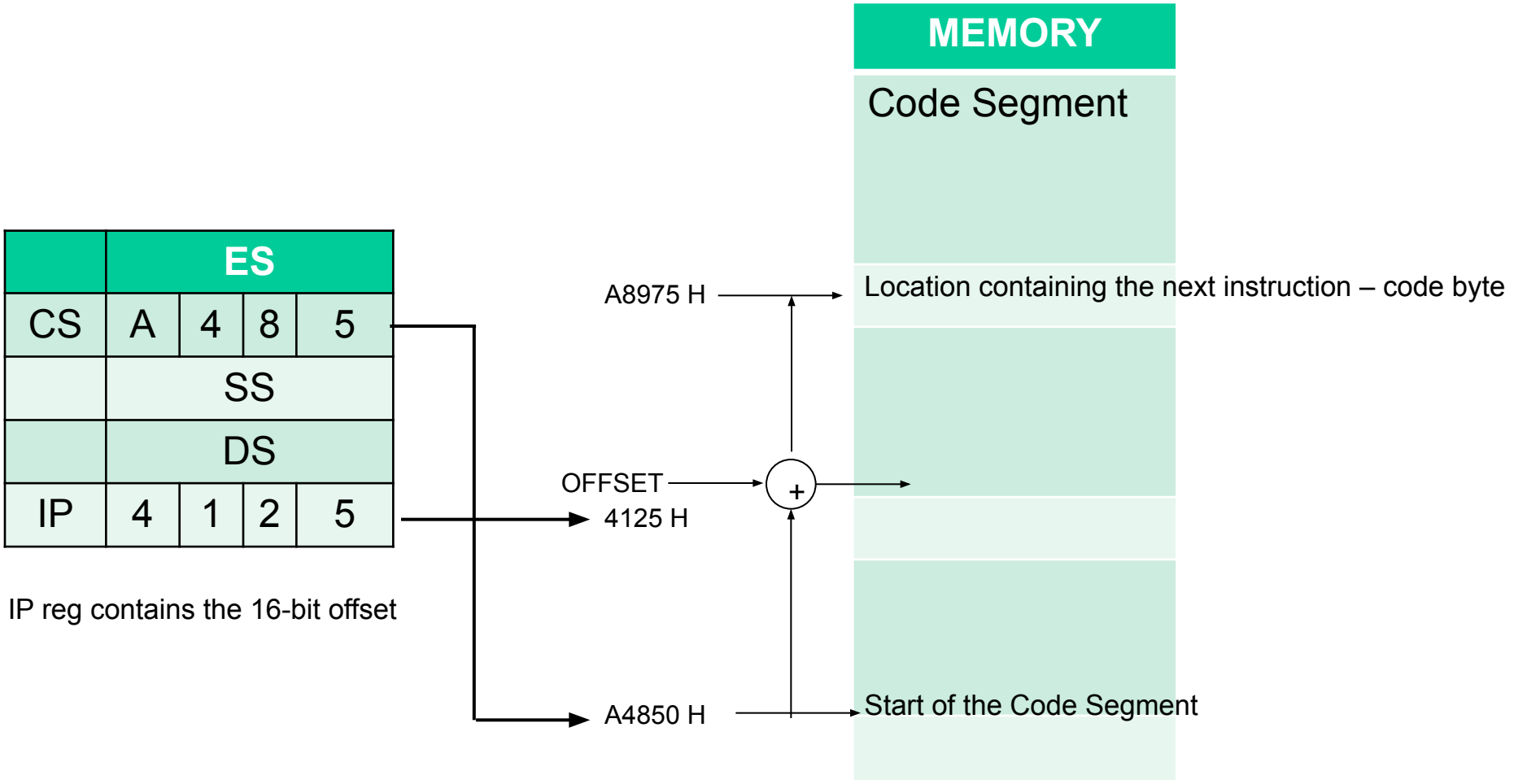
20 bit physical address of the location containing the next code byte to be fetched

Representation of Physical Address

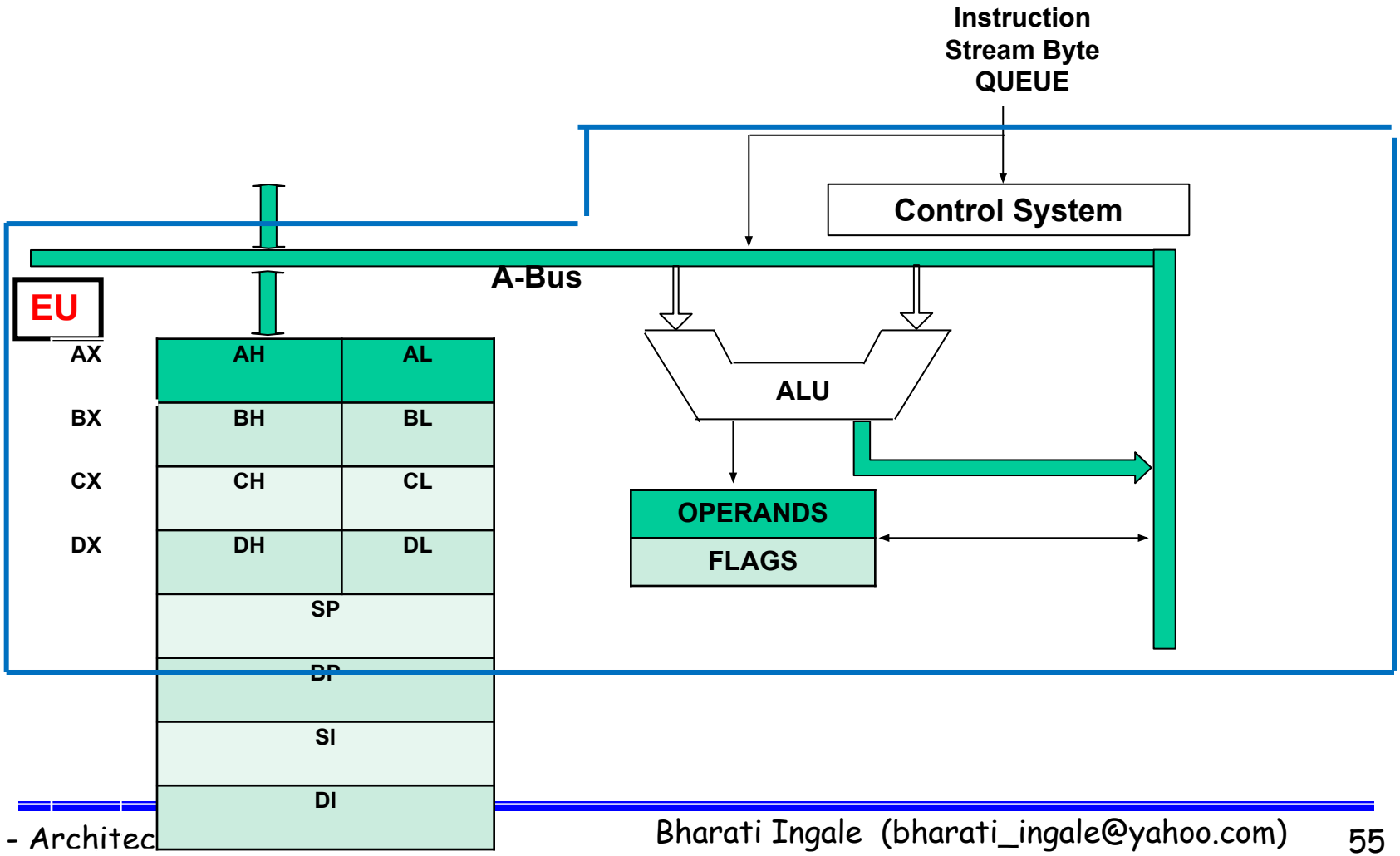
Segment Base	:	**Offset**

Bharati Ingale (bharati_ingale@yahoo.com)

8086- Architecture



Execution Unit



8086- Architecture

Execution Unit: the main function is decoding and execution of the instructions.

i. **Fetches** the instruction from the Instruction Queue, decodes & executes it.

ii. **Performs** arithmetic, logic, decision making & data transfer operations

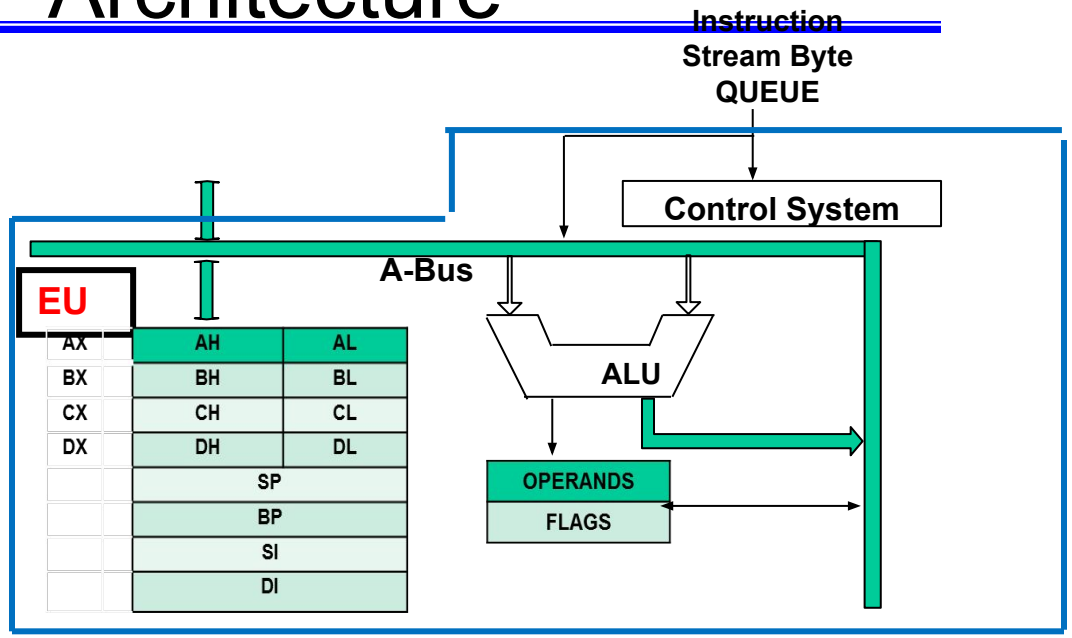
iii. **Sends request signals** to BIU to access the external module.

Does not operate w.r.t Bus cycles but operates w.r.t T States.

8086- Architecture

The Execution Unit:

- It has the following units:
  - i. Arithmetic Logic unit (ALU)
  - ii. Flag Register.
  - iii. General Purpose Registers
  - iv. Control Unit
  - v. Decoder
  - vi. Pointer and Index Register



8086- Architecture

- ALU 16 bit

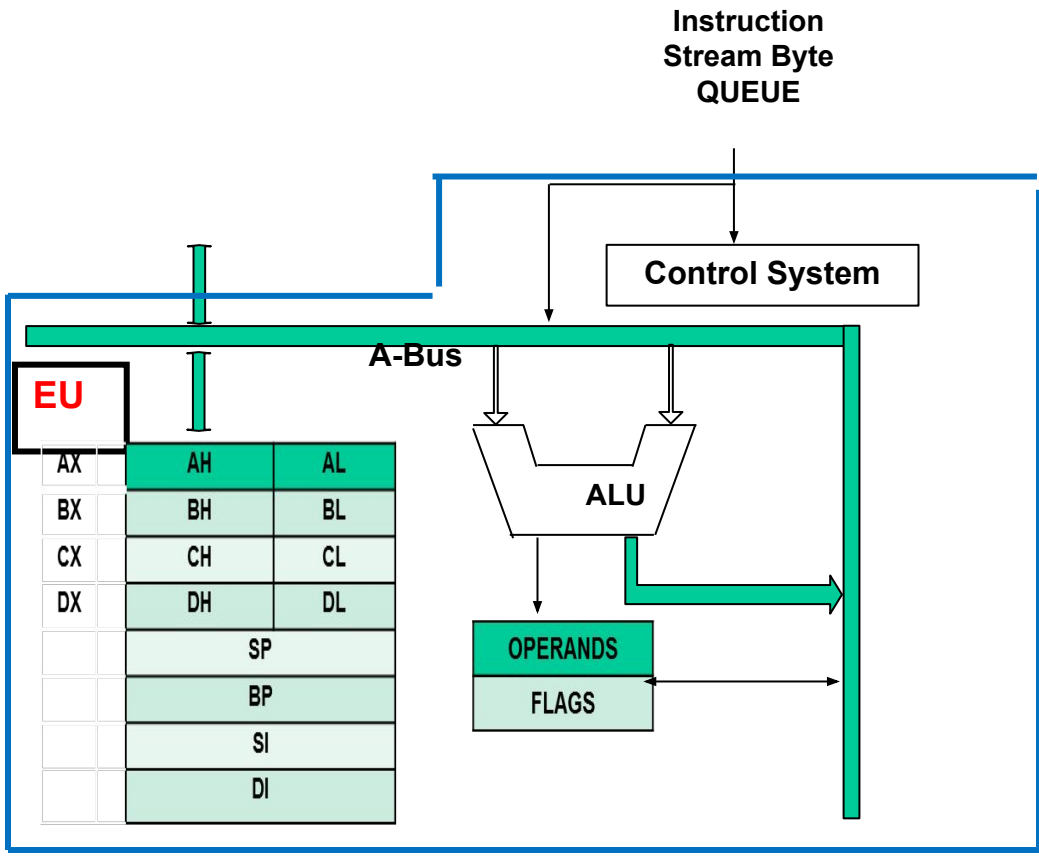
- Performs 8,16 bit arithmetic & logical operations

- Operand Register 16 bit

- Holds operands
- Not available to programmer

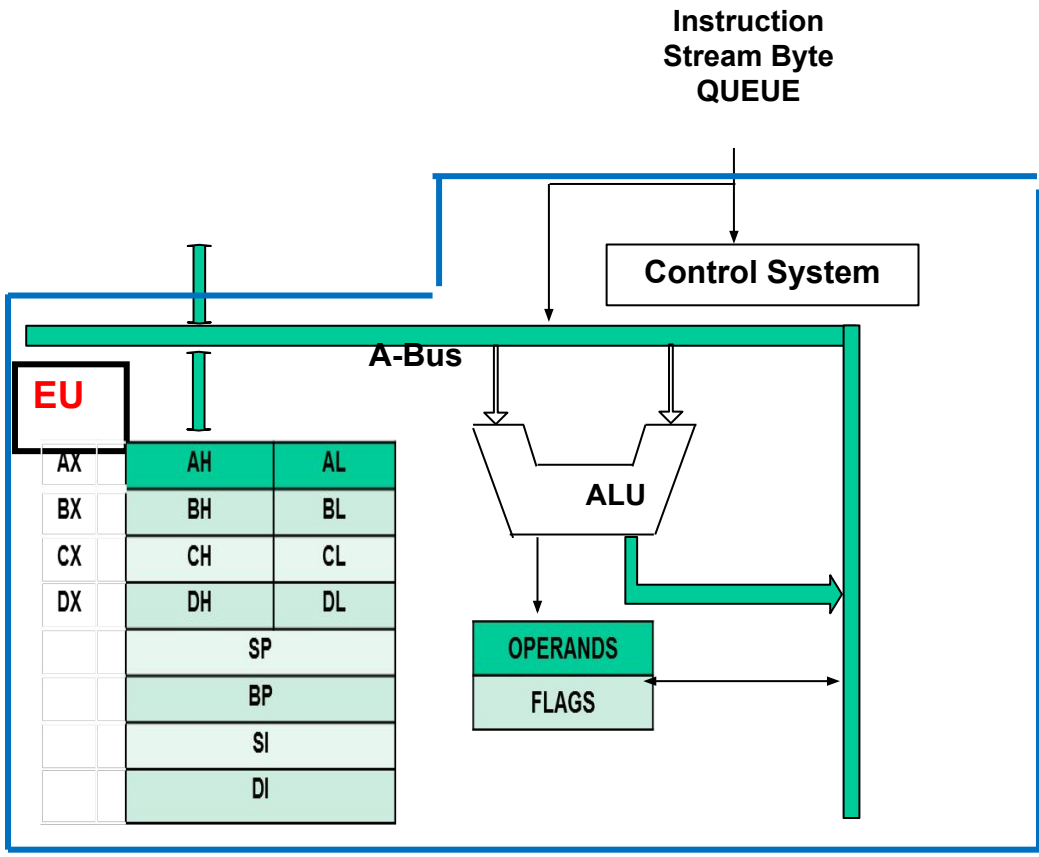
- Control Circuitry

- Used for directing the internal operation



8086- Architecture

- Instruction register & Instruction Decoder



- EU fetches opcode from the queue into the instruction register.
- The instruction decoder decodes it and sends the info to the control circuit for execution

8086- Architecture

Flag Register

- Flag is a flipflop
- 9 flags
- 2 types: 6-Status & 3-Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	OF	DF	IF	TF	SF	ZF	X	AF	X	PF	X	CF

Control Flag		Status Flag	
1	DF – Direction	1	SF - Sign
2	IF – interrupt Enable	2	ZF – Zero
3	TF – Trap	3	AF – Auxillary Carry
		4	PF – Parity
		5	CF – Carry
		6	OF - Overflow

- Stat
- Con

8086- Architecture

General Purpose Registers

- 8 bit register- AH,AL,BH,BL,CH,CL,DH,DL
- 16 bit registers – AX,BX,DX
- 32 bit registers – DX:AX

AX	**AH**	**AH**	Accumulator
BX	**BH**	**BH**	Base Register
CX	**CH**	**CH**	Count Register
DX	**DH**	**DH**	Data Register

8086- Architecture

General Purpose Registers:

Special functions of general purpose register

1. AX reg: – 16 bit accumulator

- Holds operands and results
- Holds data during I/O operation
- Holds data during string operations

8086- Architecture

General Purpose Registers:

Special functions of general purpose register

2. BX reg:– 16 bit base register

- The only GPR which may be used for indirect addressing: Holds memory addr in Base Relative addressing mode
- MOV [BX],AX

8086- Architecture

General Purpose Registers

3. CX reg:– 16 bit count register

- It is the default counter register for 3 instructions: Loop, Rotate & with prefixes Rep
- Holds count for instructions like Loop, Rotate etc.
- Loop instructions: indicate iterations
- Shift & Rotate: indicate no.of shifts or rotations (only CL used 8 bit)
- String instruction with prefixes Rep: Indicates the size of string block

8086- Architecture

General Purpose Registers

4. DX reg:– 16 bit data register

- Used with AX to hold result greater than 16bit
- Used to hold port no for IN and OUT instructions
- Used as a only pointer for 16 bit address accesses of I/O port in Indirect addressing mode

8086- Architecture

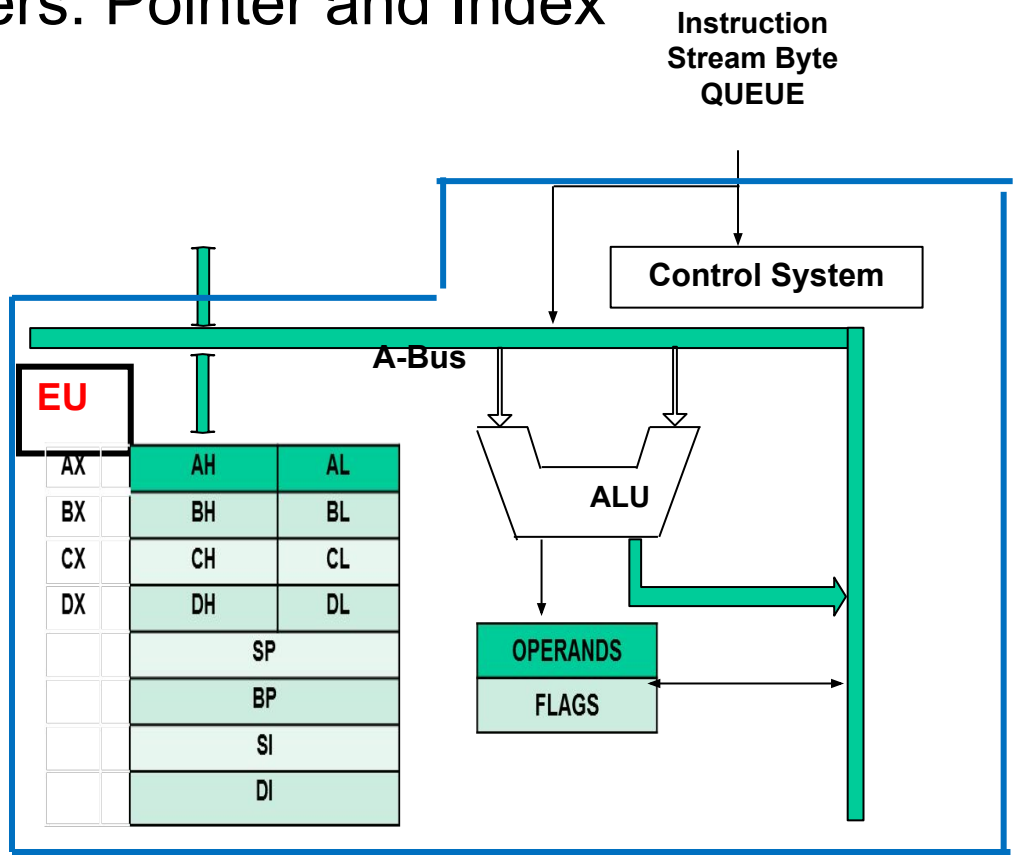
- Special Purpose Registers: Pointer and Index

i. Stack Pointer (SP 16 bit)

ii. Base Pointer (BP 16 bit)

iii. Source Index(SI 16 bit)

iv. Destination Index(DI 16 bit)

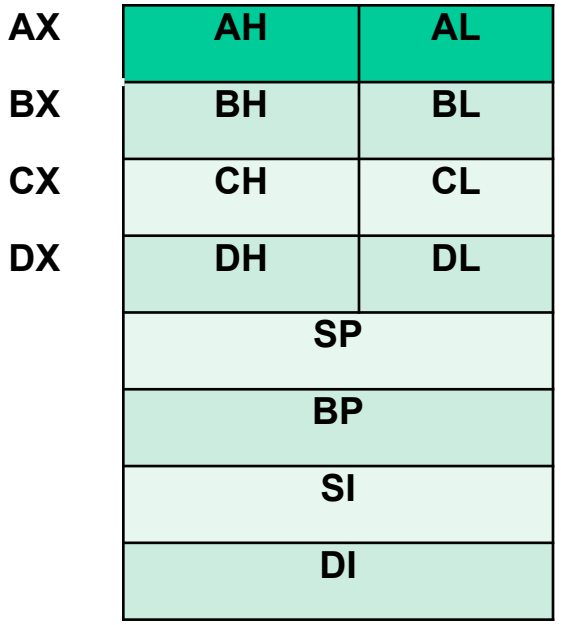


8086- Architecture

Special Purpose Registers: Pointer and Index

- Used to hold 16 bit offset of data word in one of the segments

1. **Base Pointer (BP 16 bit)**
   - Holds the 16 bit offset relative to the stack segment (SS) register.
   - Specifically used when we pass parameter via stack

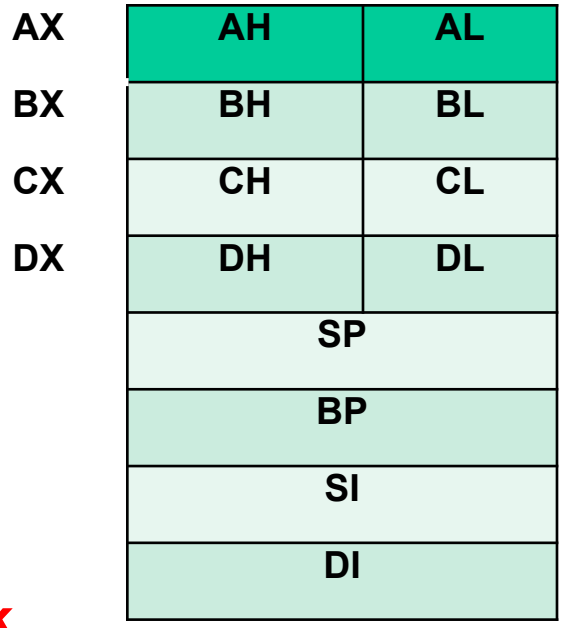


Also used as an offset register in base addressing mode.

8086- Architecture

2. Stack Pointer (SP 16 bit)

- Holds the 16 bit offset address relative to stack segment (SS) register
- SP is used in sequential access of stack segment
- **Always points to the top of the stack**



3. Source Index (SI 16 bit)

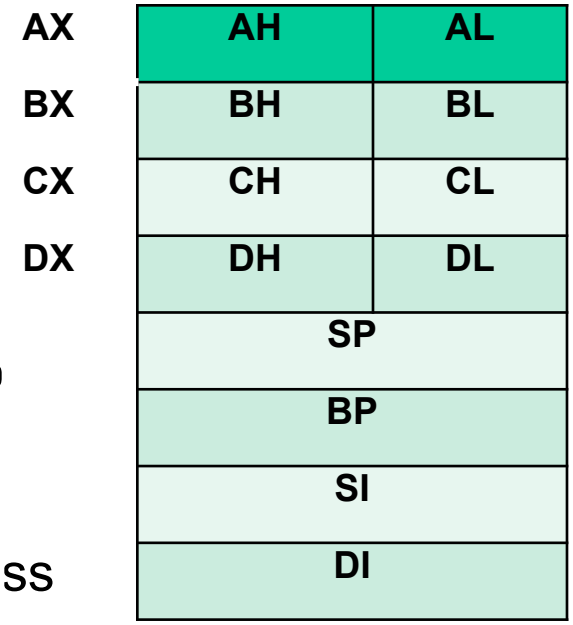
- Holds the 16 bit offset of a data word in the data segment (DS)
- The physical address is generated by adding a hard wired 0 to the segment base(DS) and then adding the offset in the index reg to it.

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL
	SP	
	BP	
	SI	
	DI	

8086- Architecture

4. Destination Index (DI 16 bit)

- Holds the 16 bit offset of a data word in the extra segment (ES)
- SI and DI used for string related instructions
- Eg: To move block of data from memory to memory
  - SI – to point the source memory address
  - DI – to point the destination memory address

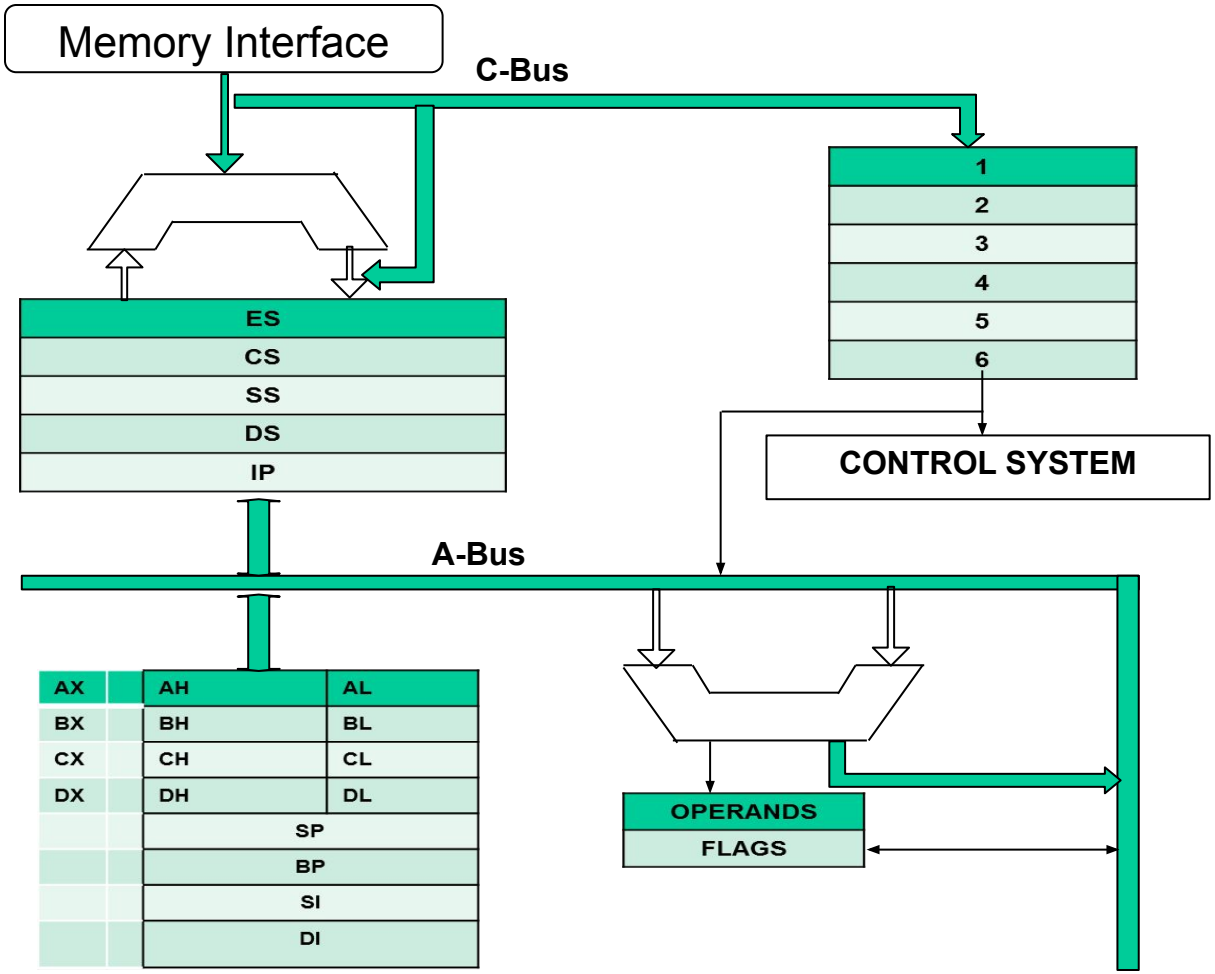


Default Segment Register	Pointer Register
CS	IP
DS	BX
DS	SI
DS(ES in case of STRING Operation)	DI
SS	SP
SS	BP

8086- Architecture

Bus Interfacing Unit (BIU)	Execution Unit (EU)
It sends out addresses	It tells the BIU, from where to fetch the instruction or data
It fetches instruction from memory	It decodes the fetched instructions
It reads the data from memory and ports.	It executes the decoded instructions
So BIU takes care of all the address and data transfer on the buses	EU takes care of performing operations on the data
Hence it is called the external world interface of the processor	EU is called as the execution heart of the processor.
It works in synchronous with machine cycles	It works in synchronous with t- states

Thank You



Bharati Ingale (bharati_ingale@yahoo.com)

8086 - Architecture