COMPUTER ORGANIZATION AND ARCHITECTURE (COA)

EET 2211

4TH SEMESTER – CSE & CSIT

OVERVIEW OF 8086 MICROPROCESSOR

OVERVIEW OF 8086 MICROPROCESSOR

TOPICS TO BE COVERED

- Register Organization of 8086
- Architecture
- Addressing modes of 8086
- Instruction set of 8086

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Present an overview of the evolution of computer technology from early digital computers to the latest microprocessors.
- Present an overview of the evolution of the x86 architecture.

MICROPROCESSOR

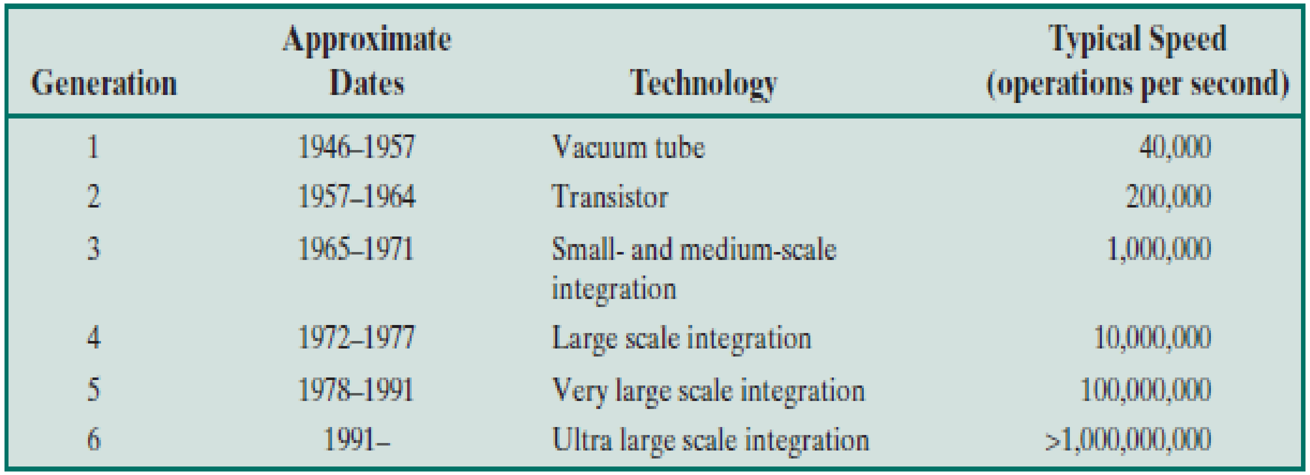
- **Microprocessor** is a miniature electronic device that contains the arithmetic, logic, and control circuitry necessary to perform the functions of a digital computer's central processing unit.
- It can interpret and execute program instructions as well as handle arithmetic operations.
- The first microprocessor was the Intel 4004, which was introduced in 1971.
- The production of inexpensive microprocessors enabled computer engineers to develop microcomputers.
- These computer systems are small but have enough computing power to perform many business, industrial, and scientific tasks.
- The microprocessor also permitted the development of so-called intelligent terminals, such as automatic teller machines and point-of-sale terminals employed in retail stores.
- The microprocessor also provides automatic control of industrial robots, surveying instruments, and various kinds of hospital equipment.
- It has brought about the computerization of a wide array of consumer products, including programmable microwave ovens, television sets, and electronic games. In addition, some automobiles feature microprocessor-controlled ignition and fuel systems designed to improve performance and fuel economy.

HISTORICAL BACKGROUND

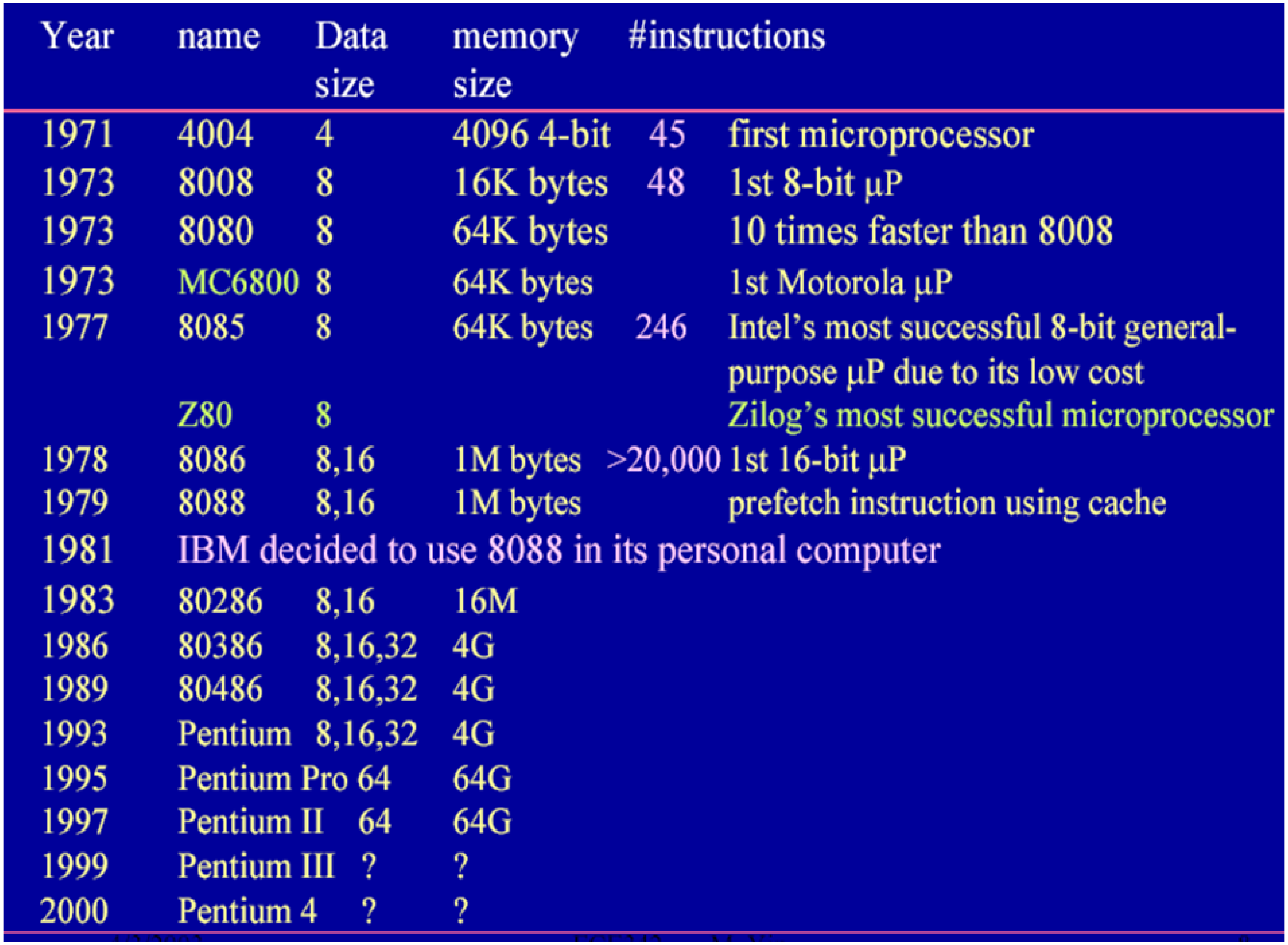
- The Mechanical Age
- The Electrical Age

1946	✓ The first general purpose programmable electronic computer system- ENIAC (Electronics Numerical Integrator and Calculator) was developed. ✓ 17,000 vacuum tubes ✓ 500 miles of wires ✓ Weighted over 30 tons ✓ Performed about 1,00,000 operations per second ✓ Programmed by rewiring its circuits
1948	Development of the transistor (Bell Labs)
1958	Invention of the integrated circuits (Texas)

COMPUTER GENERATIONS



Generation	Approximate Dates	Technology	Typical Speed (operations per second)
1	1946–1957	Vacuum tube	40,000
2	1957–1964	Transistor	200,000
3	1965–1971	Small- and medium-scale integration	1,000,000
4	1972–1977	Large scale integration	10,000,000
5	1978–1991	Very large scale integration	100,000,000
6	1991–	Ultra large scale integration	>1,000,000,000



Year	name	Data size	memory size	#instructions	
1971	4004	4	4096 4-bit	45	first microprocessor
1973	8008	8	16K bytes	48	1st 8-bit µP
1973	8080	8	64K bytes		10 times faster than 8008
1973	MC6800	8	64K bytes		1st Motorola µP
1977	8085	8	64K bytes	246	Intel's most successful 8-bit general-purpose µP due to its low cost
	Z80	8			Zilog's most successful microprocessor
1978	8086	8,16	1M bytes	>20,000	1st 16-bit µP
1979	8088	8,16	1M bytes		prefetch instruction using cache
1981	IBM decided to use 8088 in its personal computer				
1983	80286	8,16	16M		
1986	80386	8,16,32	4G		
1989	80486	8,16,32	4G		
1993	Pentium	8,16,32	4G		
1995	Pentium Pro	64	64G		
1997	Pentium II	64	64G		
1999	Pentium III	?	?		
2000	Pentium 4	?	?		

EVOLUTION OF INTEL MICROPROCESSORS

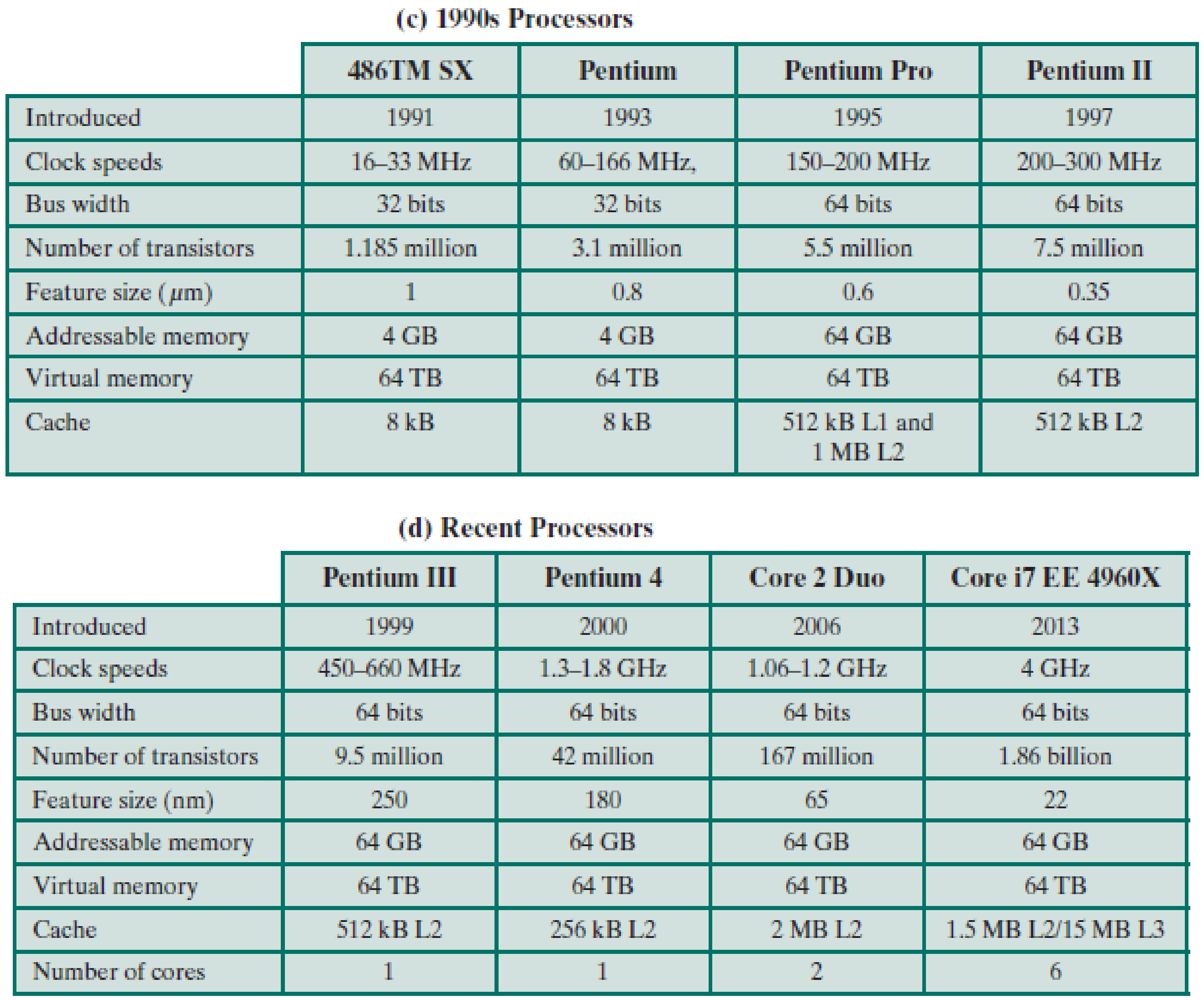
(a) 1970s Processors

	4004	**8008**	**8080**	**8086**	**8088**
Introduced	1971	1972	1974	1978	1979
Clock speeds	108 kHz	108 kHz	2 MHz	5 MHz, 8 MHz, 10 MHz	5 MHz, 8 MHz
Bus width	4 bits	8 bits	8 bits	16 bits	8 bits
Number of transistors	2,300	3,500	6,000	29,000	29,000
Feature size (μm)	10	8	6	3	6
Addressable memory	640 bytes	16 KB	64 KB	1 MB	1 MB

(b) 1980s Processors

	80286	**386TM DX**	**386TM SX**	**486TM DX CPU**
Introduced	1982	1985	1988	1989
Clock speeds	6–12.5 MHz	16–33 MHz	16–33 MHz	25–50 MHz
Bus width	16 bits	32 bits	16 bits	32 bits
Number of transistors	134,000	275,000	275,000	1.2 million
Feature size (μm)	1.5	1	1	0.8–1
Addressable memory	16 MB	4 GB	16 MB	4 GB
Virtual memory	1 GB	64 TB	64 TB	64 TB
Cache	—	—	—	8 kB

Contd.



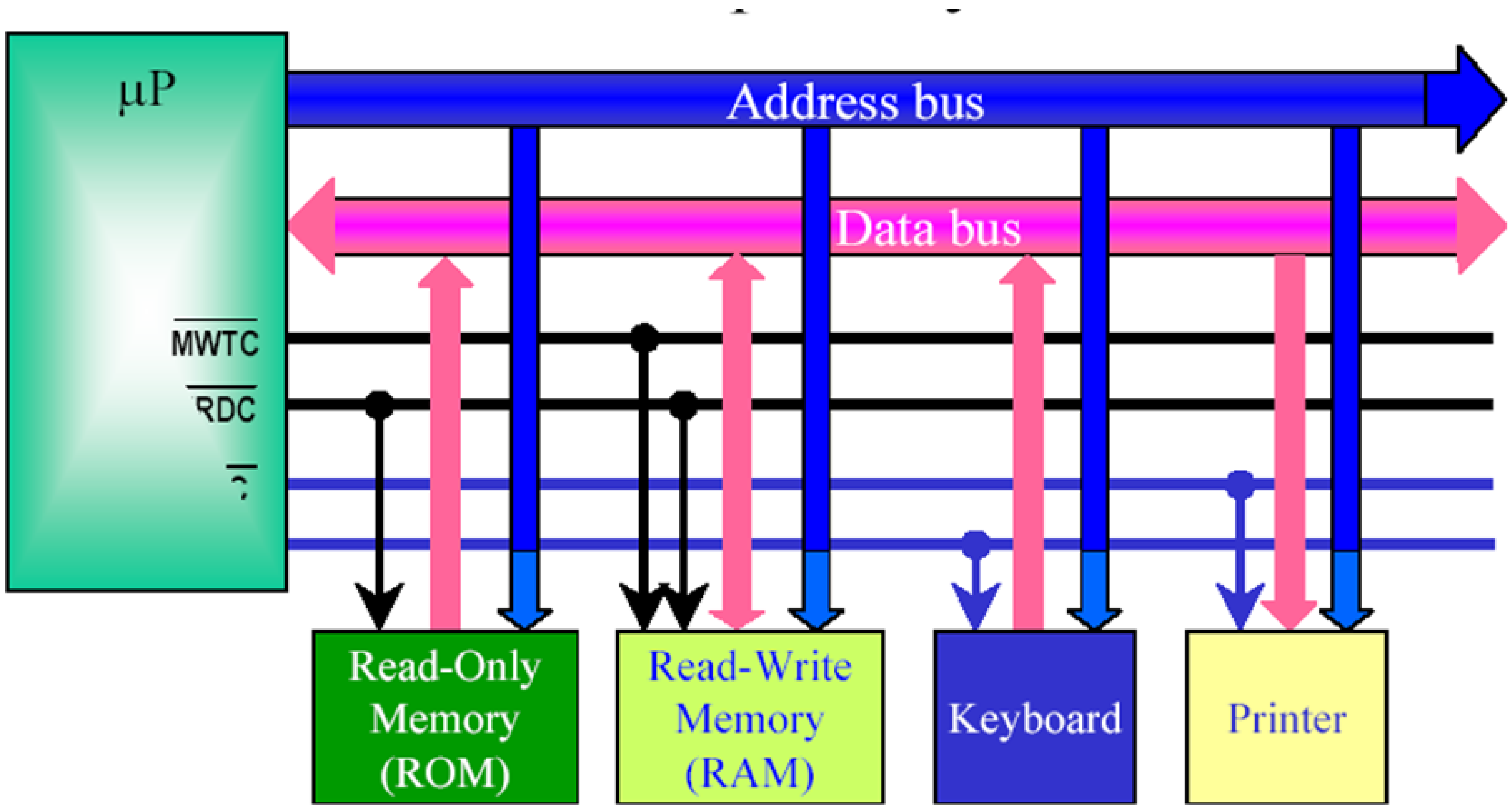
(c) 1990s Processors

	486TM SX	Pentium	Pentium Pro	Pentium II
Introduced	1991	1993	1995	1997
Clock speeds	16–33 MHz	60–166 MHz,	150–200 MHz	200–300 MHz
Bus width	32 bits	32 bits	64 bits	64 bits
Number of transistors	1.185 million	3.1 million	5.5 million	7.5 million
Feature size (μm)	1	0.8	0.6	0.35
Addressable memory	4 GB	4 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	8 kB	8 kB	512 kB L1 and 1 MB L2	512 kB L2

(d) Recent Processors

	Pentium III	Pentium 4	Core 2 Duo	Core i7 EE 4960X
Introduced	1999	2000	2006	2013
Clock speeds	450–660 MHz	1.3–1.8 GHz	1.06–1.2 GHz	4 GHz
Bus width	64 bits	64 bits	64 bits	64 bits
Number of transistors	9.5 million	42 million	167 million	1.86 billion
Feature size (nm)	250	180	65	22
Addressable memory	64 GB	64 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	512 kB L2	256 kB L2	2 MB L2	1.5 MB L2/15 MB L3
Number of cores	1	1	2	6

Block Diagram of a Microprocessor-based Computer system



OVERVIEW OF 8086 MICROPROCESSOR

- In April 1978, Intel introduced its first 16 bit microprocessor.
- Production started in May, eventually the 8086 was officially released on June 8.

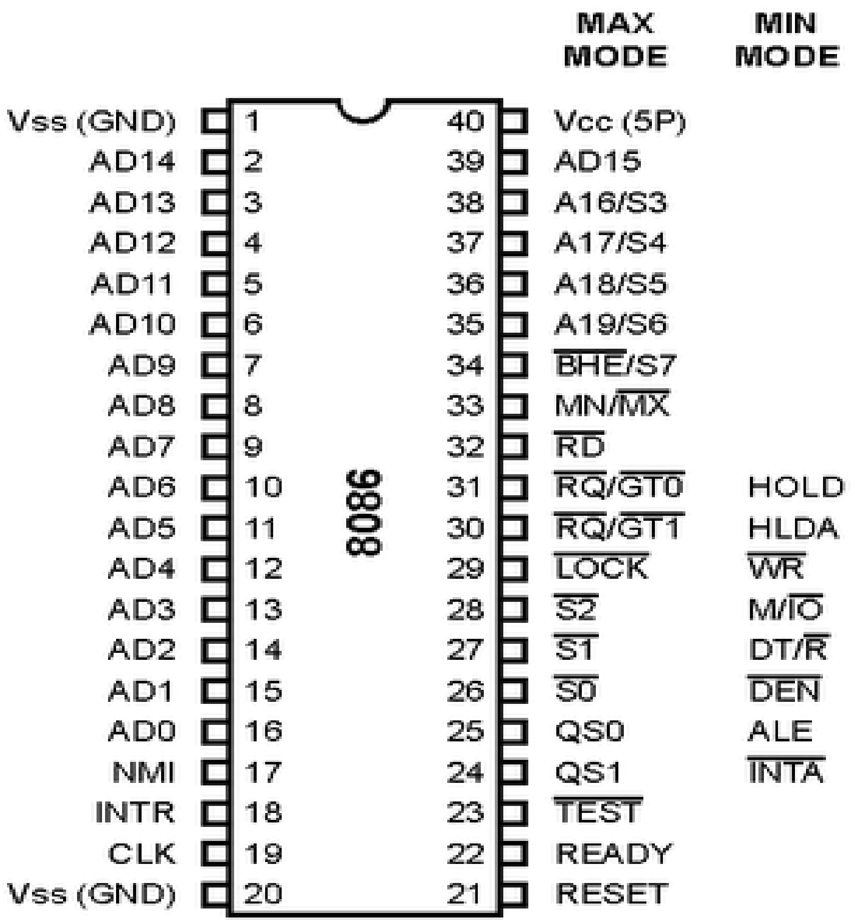


Fig. : Architecture diagram of 8086 Microprocessor IC.

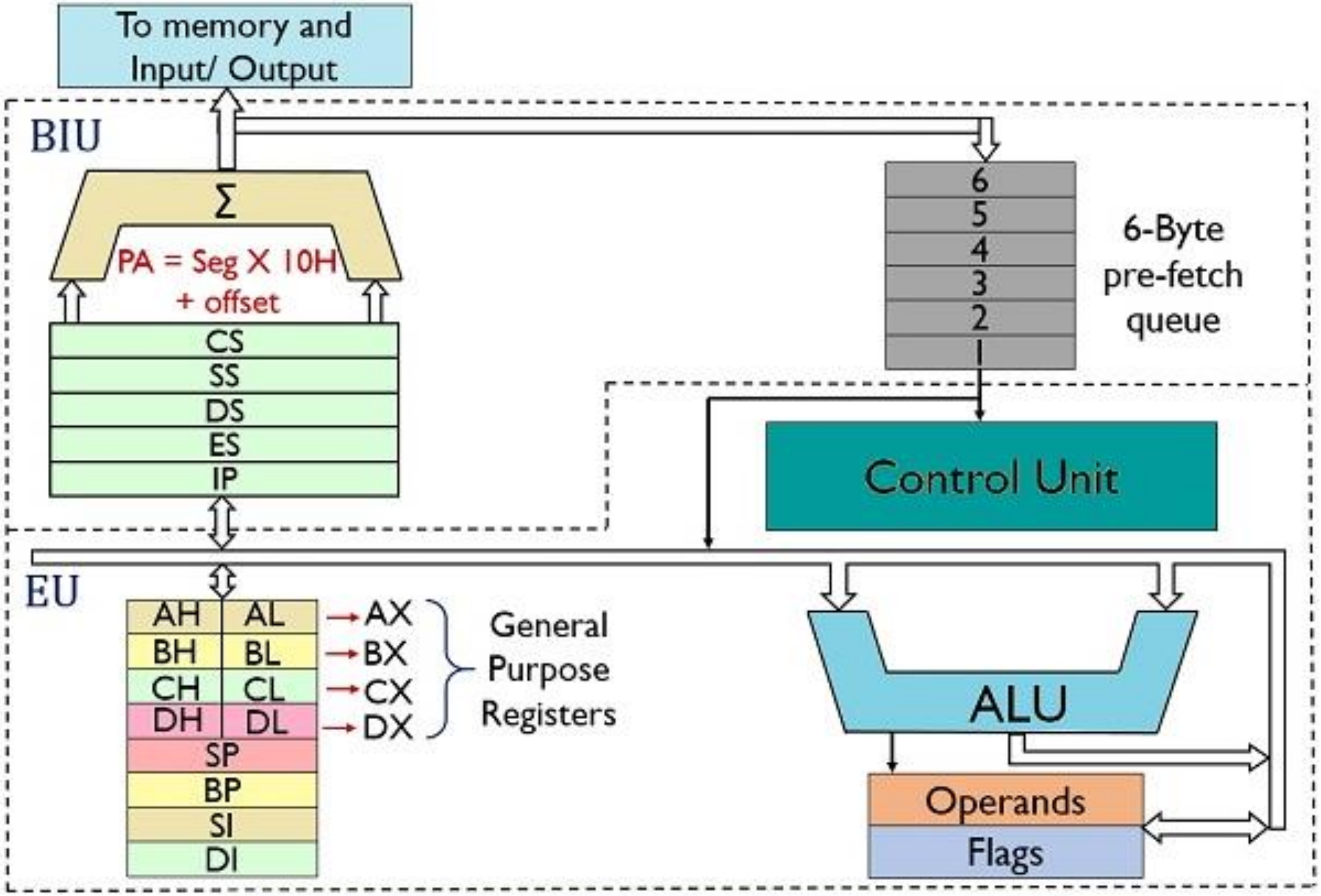
FEATURES OF 8086

The most prominent features of a 8086 microprocessor are as follows:

- ✓ It is a 40 pin dual in line package IC.
- ✓ It is a 16-bit microprocessor.
- ✓ 8086 has a 20-bit address bus and can access up to 2^20 (1 MB) memory locations.
- ✓ It can support up to 64K I/O ports.
- ✓ It provides 14, 16-bit registers.
- ✓ Word size is 16 bits and double word size is 4 bytes.
- ✓ It has multiplexed address and data bus AD0-AD15 and A16-A19.

Contd.

- ✓ It requires +5V power supply.
- ✓ It can pre-fetch up to 6 instruction bytes from memory and queues them in order to speed up instruction execution.
- ✓ It has multiplexed address and data bus AD0-AD15 and A16-A19.
- ✓ It requires single phase clock with 33% duty cycle to provide internal timing.
- ✓ Address ranges from 00000H to FFFFFH.
- ✓ Memory is byte addressable – every byte has a separate address.
- ✓ 8086 is designed to operate in two modes: Minimum and Maximum.



Block Diagram of 8086 Microprocessor

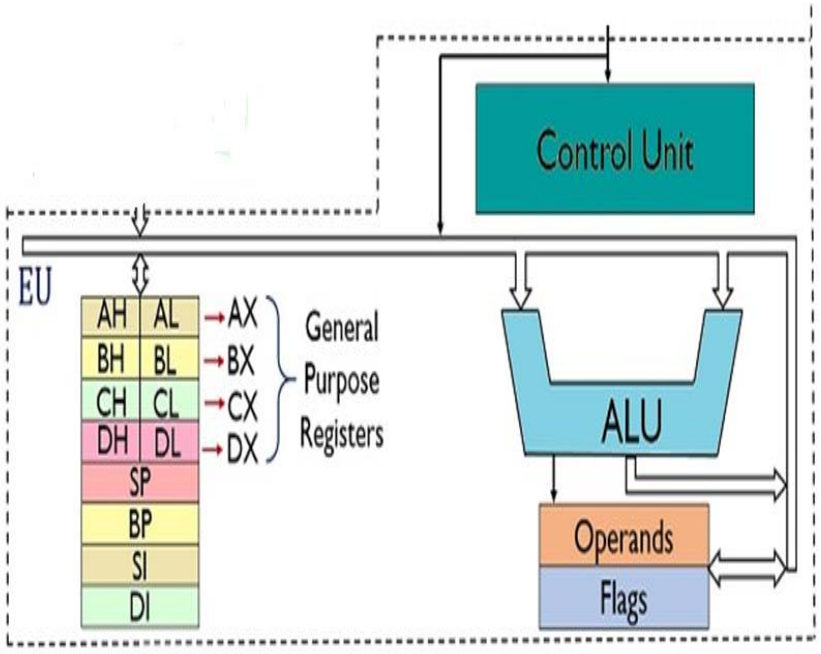
- 8086 has two blocks: BIU and EU.
- The BIU handles all transactions of data and address on the buses for EU.
- The BIU performs all bus operations such as instruction fetching, reading and writing operands for memory and calculating the addresses of the memory operands. The instruction bytes are transferred to the instruction queue.
- EU executes instructions from the instruction system byte queue.
- BIU contains Instruction queue, Segment registers, Instruction pointer and Address adder.
- EU contains Control circuitry, Instruction decoder, ALU, Pointer and index register and Flag register.

EU (Execution Unit)

- ✓ Decodes instructions fetched by the BIU.
- ✓ Generates control signals.
- ✓ Executes instructions.

Main components are:

- ✓ Instruction Decoder
- ✓ Control System
- ✓ Arithmetic Logic unit
- ✓ General Purpose Registers
- ✓ Flag Register
- ✓ Pointer and Index Registers



INSTRUCTION DECODER

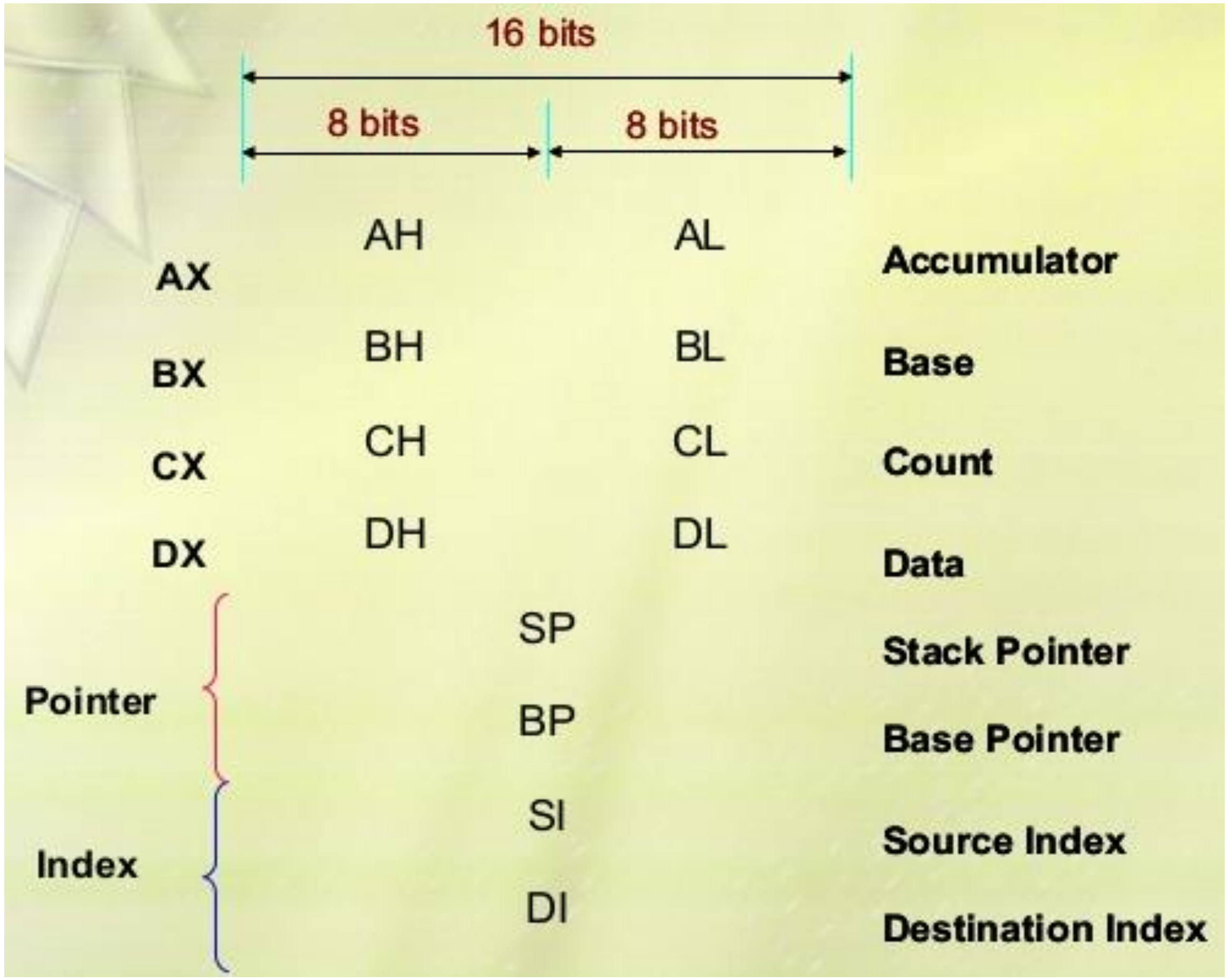
Translates instructions fetched from memory into a series of actions which EU carries out.

CONTROL SYSTEM

Generates timing and control signals to perform the internal operations of the microprocessor.

ARITHMETIC LOGIC UNIT

EU has a 16-biti ALU which can ADD, SUB, AND, OR, increment, decrement, complement or shift binary numbers.


16 bits
8 bits
8 bits
AX
AH
AL
Accumulator
BX
BH
BL
Base
CX
CH
CL
Count
DX
DH
DL
Data
Pointer
SP
Stack Pointer
BP
Base Pointer
Index
SI
Source Index
DI
Destination Index

General Purpose Registers

- EU has 8 general purpose registers.
- Can be individually used for storing 8-bit data.
- AL register is also called Accumulator.
- Two registers can also be combined to form 16-bit registers.
- The valid register pairs are – AX, BX, CX, DX.

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

GPR (contd.)

REGISTER	PURPOSE
AX	Word multiply, word divide, word I/O
AL	Byte multiply, byte divide, byte I/O, decimal arithmetic
AH	Byte multiply, byte divide
BX	Store address information
CX	String operation, loops
CL	Variable shift and rotate
DX	Word multiply, word divide, indirect I/O (used to hold I/O address during I/O instructions. If the result is more than 16-bits, the lower order 16-bits are stored in accumulator and higher order 6-bits are stored in DX register)

Flag Register

- 8086 has a 16-bit flag register.

- A flag is a flip-flop which indicates some conditions produced by the execution of an instruction or controls certain operations of the EU.

- Contains 9 active flags (out of 16 flags) and the remaining 7 are undefined.

- There are two types of flags in 8086:

i. Conditional flags – six flags, set or reset by EU on the basis of results of some arithmetic operations or else also known as status flags as they indicates some conditions.

ii. Control flags – three flags, used to control certain operations of the processor.

Flag Register (contd.)

U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF

U= unused

1.	CF	CARRY FLAG	CONDITIONAL FLAGS
2.	PF	PARITY FLAG	
3.	AF	AUXILIARY FLAG	
4.	ZF	ZERO FLAG	
5.	SF	SIGN FLAG	
6.	OF	OVERFLOW FLAG	
7.	TF	TRAP FLAG	CONTROL FLAGS
8.	IF	INTERRUPT FLAG	
9.	DF	DIRECTION FLAG	

FLAG	PURPOSE
CF	Holds the carry after addition or the borrow after subtraction. Also indicates some error conditions as dictated by some programs and procedures.
PF	PF=0= odd parity ; PF=1=even parity
AF	Holds the carry (half carry) after addition or borrow after subtraction between bit positions 3 and 4 of the result (e.g. in BCD addition or subtraction)
ZF	Shows the result of the arithmetic or logic operation.
SF	Holds the sign of the result after an arithmetic/logic instruction execution.
TF	A control flag – it enables the trapping through an on-chip debugging feature.
IF	A control flag – controls the operation of the INTR (interrupt request). I=0=INTR pin disabled; I=1= INTR pin enabled.
DF	A control flag – it selects either the increment or decrement mode for DI and-or SI registers during the string instructions.
OF	Overflow occurs when signed numbers are added or subtracted. An overflow indicates the result has exceeded the capacity of the machine.

Pointer and Index Registers

- ✓ Used to keep offset addresses.
- ✓ Used in various forms of memory addressing.
- ✓ In the case od SP and BP the default reference to form a physical address is the Stack segment (SS).
- ✓ The index registers (SI and DI) and the BX generally default to the Data segment register (DS).
- ✓ SP – stack pointer – used with SS to access the stack segment.
- ✓ BP – base pointer – primarily used to access data on the stack and can be used to access data in other segments.

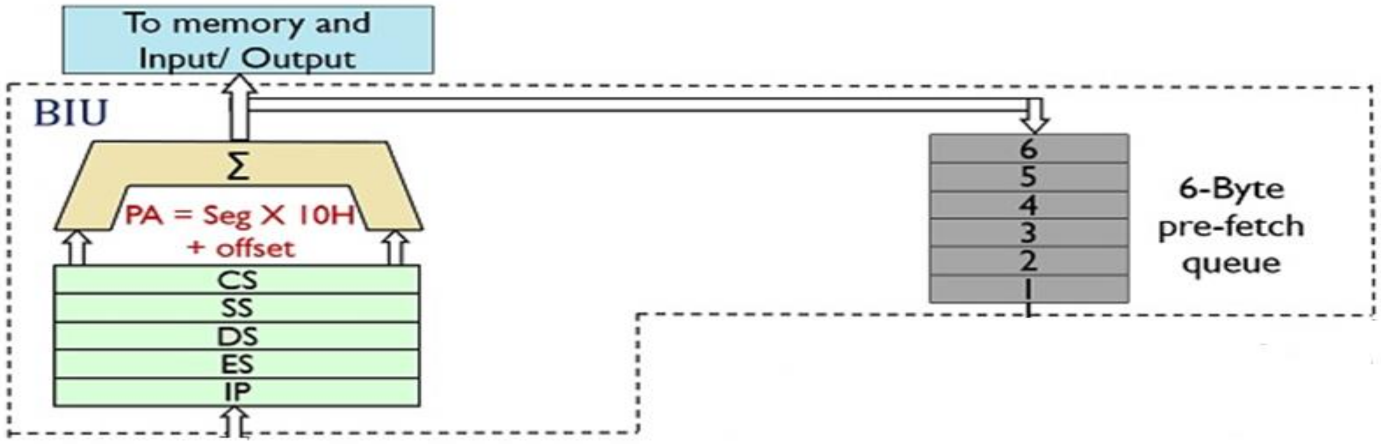
Pointer and Index Registers (contd.)

- ✓ SI – Source index register – it is required for some string operations. When the string operations are performed, the SI register points to memory locations in the data segment which is addressed by the DS register. Thus SI is associated with the DS in string operations.
- ✓ DI – destination index register – it is also required for some string operations. When string operations are performed, the DI register points to memory locations in the data segment which is addressed by the ES register. Thus, DI is associated with the ES in string operations.
- ✓ The SI and DI registers may also be used to access data stored in arrays.

BIU (Bus Interface Unit)

Main components are :

- 6 bytes Instruction Queue (Q)
- Segment Registers (CS, DS, ES, SS)
- Instruction Pointer (IP)
- The address summing block



Instruction Queue

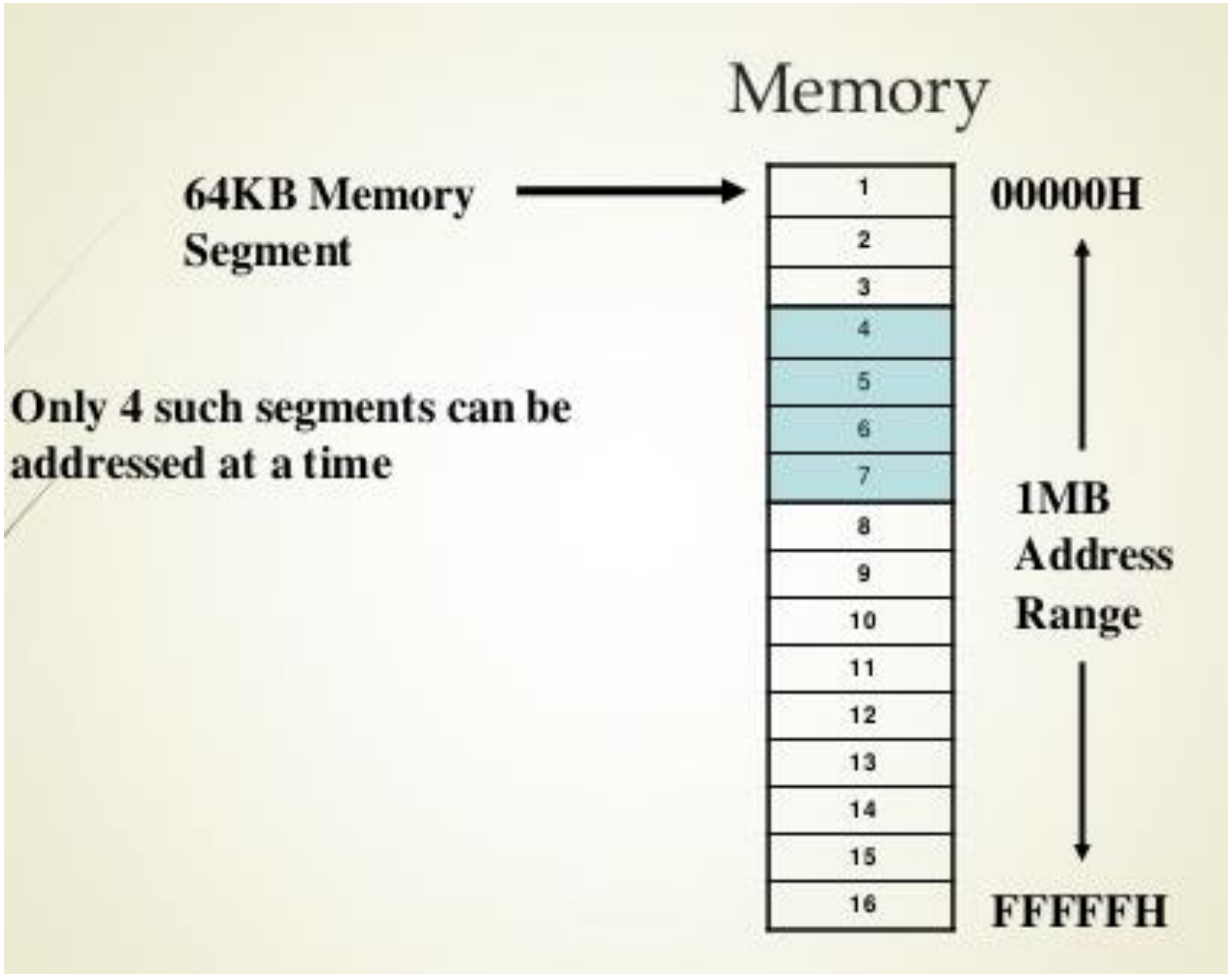
- 8086 employs parallel processing.
- The BIU uses a mechanism known as an instruction stream queue to implement a pipeline architecture.
- When EU is busy decoding or executing current instruction, the buses of 8086 may not be in use.
- At that time, BIU can use buses to fetch up to six instruction bytes for the following instructions.
- BIU stores these pre-fetched bytes in a FIFO register called Instruction Queue.
- When EU is ready for its next instruction, it simply reads the instruction from the queue in BIU.

Pipelining

- EU of 8086 does not have to wait in between for BIU to fetch next instruction byte from memory.
- So the presence of a queue in 8086 speeds up the processing.
- Fetching the next instruction while the current instruction executes is called pipelining.

Memory Segmentation

- 8086 has a 20-bit address bus.
- So it can address a maximum of 1MB of memory.
- 8086 can work with only four 64KB segments at a time within this 1MB range.
- These four memory segments are called:
  (i) **CODE** segment
  (ii) **STACK** segment
  (iii) **DATA** segment
  (iv) **EXTRA** segment


Memory
64KB Memory Segment
Only 4 such segments can be addressed at a time
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
00000H
1MB Address Range
FFFFFH

CODE SEGMENT

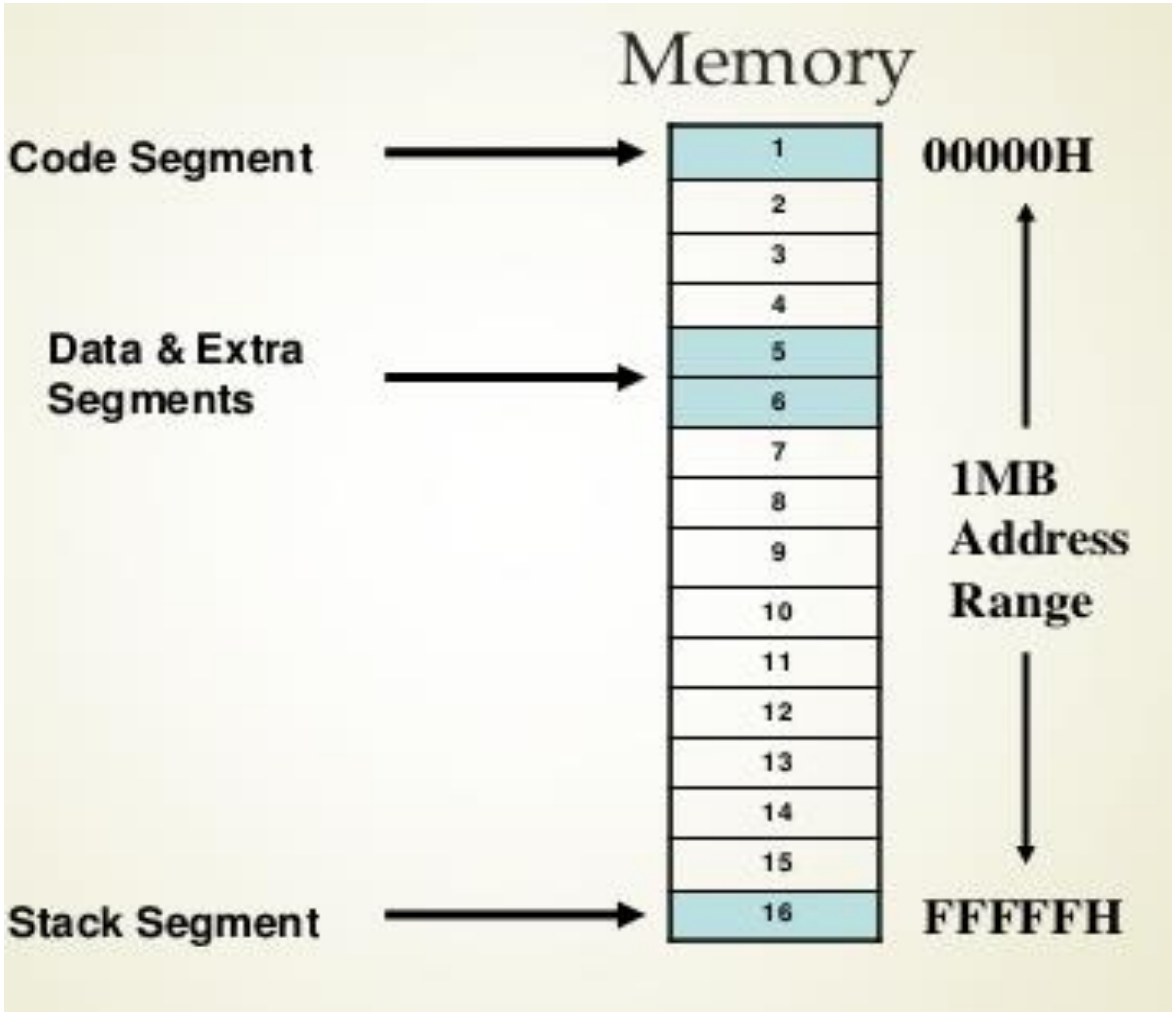
The part of memory from where BIU is currently fetching instruction code bytes. It is used for storing the instructions.

STACK SEGMENT

A section of memory set aside to store address and data while a subprogram executes. It is sued as a stack and is used to store the return address.

DATA AND EXTRA SEGMENTS

Used for storing data values or data bytes to be used in the program.

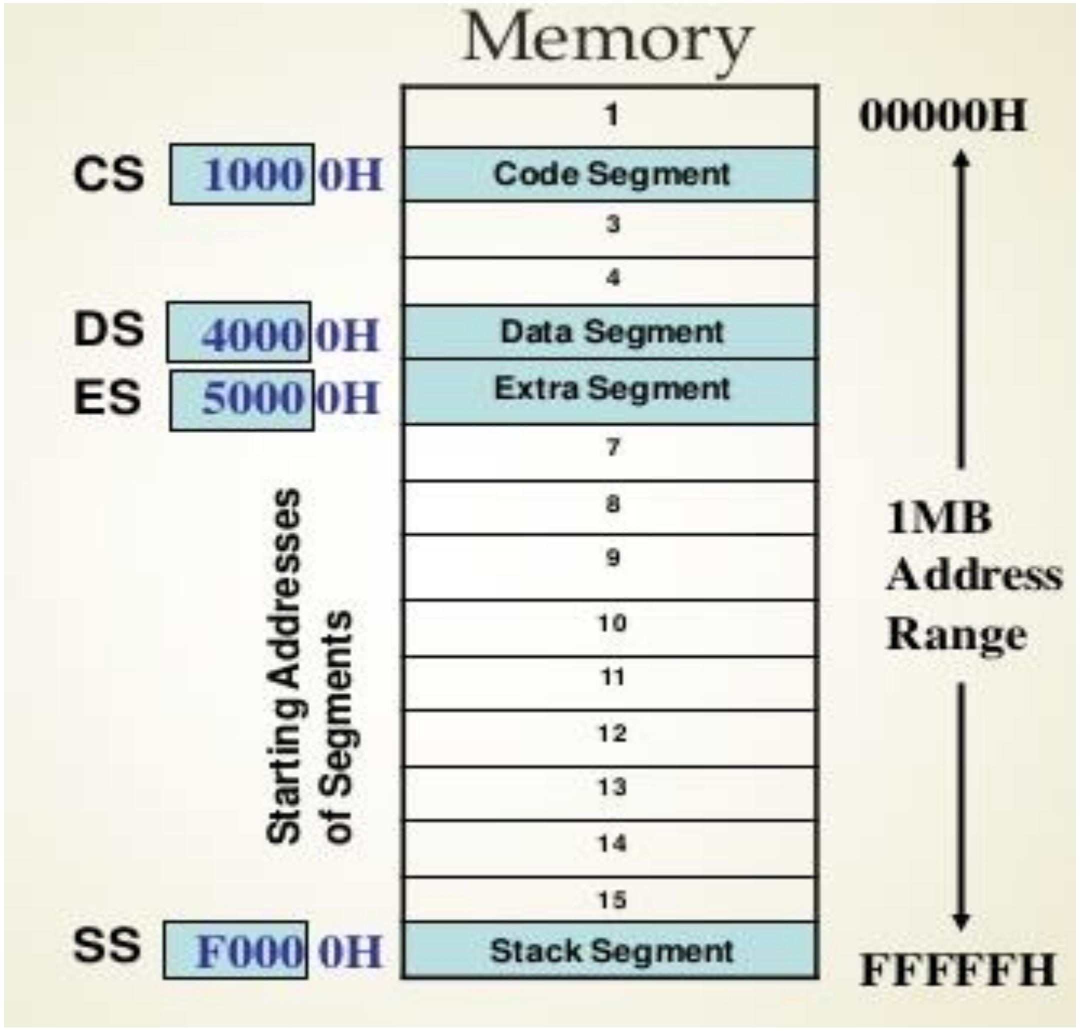

Memory
Code Segment
Data & Extra Segments
Stack Segment
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
00000H
1MB Address Range
FFFFFH

Segment Registers

- It holds the upper 16-bits of the starting address for each of the segments.

- The four segment registers are:
  (i) **CS - CODE** segment register
  (ii) **SS - STACK** segment register
  (iii) **DS - DATA** segment register
  (iv) **ES - EXTRA** segment register

- The size of each segment is 64 KB.
- A segment may be located any-where in the memory.
- Each of these segments can be used for a specific function.

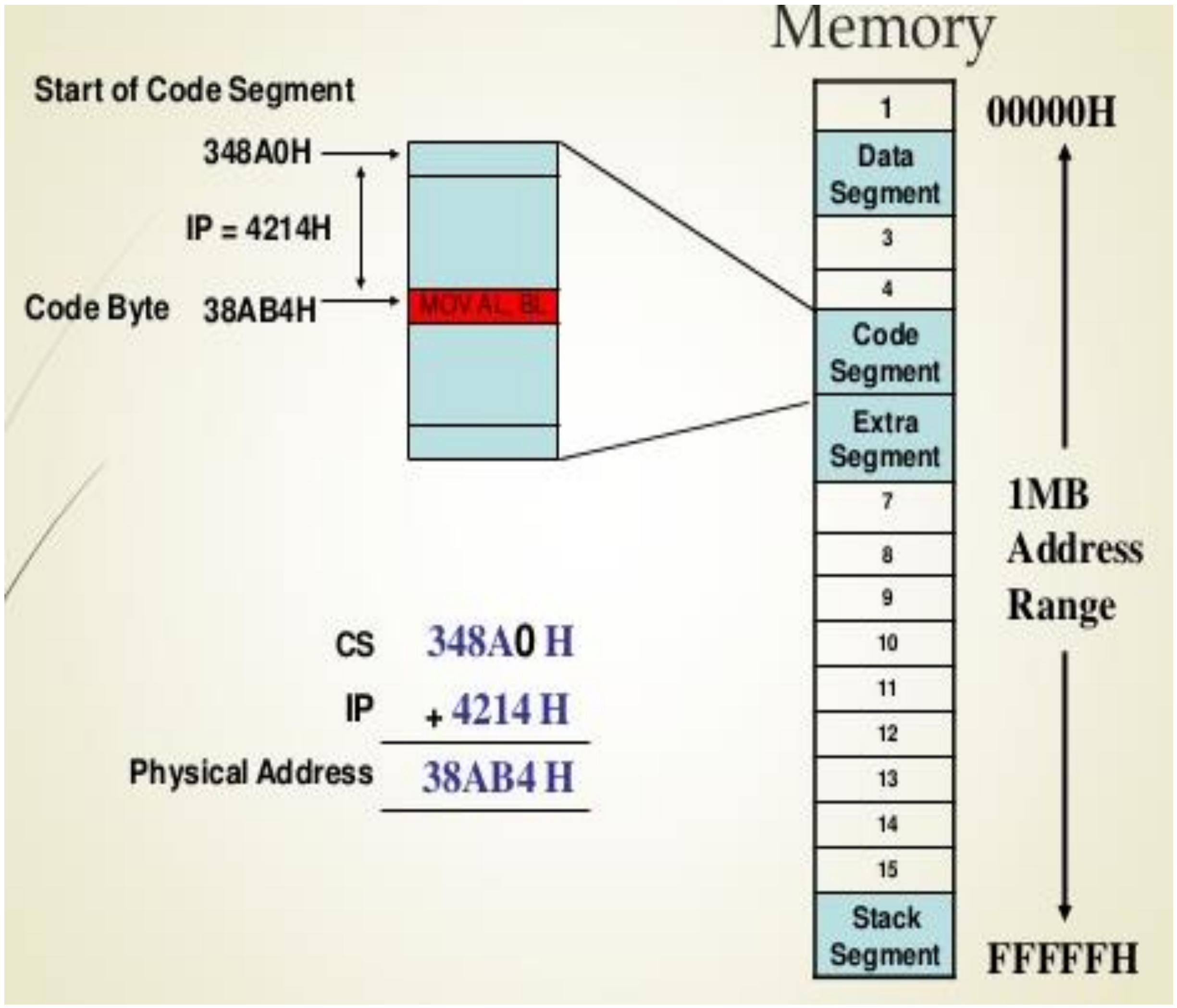

Memory
CS 1000 0H
DS 4000 0H
ES 5000 0H
SS F000 0H
Starting Addresses of Segments
1
Code Segment
3
4
Data Segment
Extra Segment
7
8
9
10
11
12
13
14
15
Stack Segment
00000H
1MB Address Range
FFFFFH

Segment Registers (contd.)

- ✓ Address of a segment is of 20-bits.
- ✓ A segment register stores only upper 16 bits of the starting address of the corresponding segments.
- ✓ The 1-bit contents of the segment registers in the BIU actually points to the starting location of a particular segment.
- ✓ BIU always inserts zeros for the lowest 4-bits of the 20-bit starting address.
- ✓ E.g. if CS = 348AH, then the code segment will start at 348A0H.
- ✓ A 64-KB segment can be located anywhere in the memory, bus will start at an address with zero in the lowest 4-bits.
- ✓ Segments may be overlapped or non-overlapped.

IP (Instruction Pointer) Register

- It is a 16-bit register.
- Holds 16-bit offset, of the next instruction byte in the code segment.
- BIU uses IP and CS registers to generate the 20-bit address of the instruction to be fetched from memory.


Memory
Start of Code Segment
348A0H
IP = 4214H
Code Byte 38AB4H
MOV AL, BL
1
Data Segment
3
4
Code Segment
Extra Segment
7
8
9
10
11
12
13
14
15
Stack Segment
00000H
1MB Address Range
FFFFFH
CS 348A0 H
IP + 4214 H
Physical Address 38AB4 H

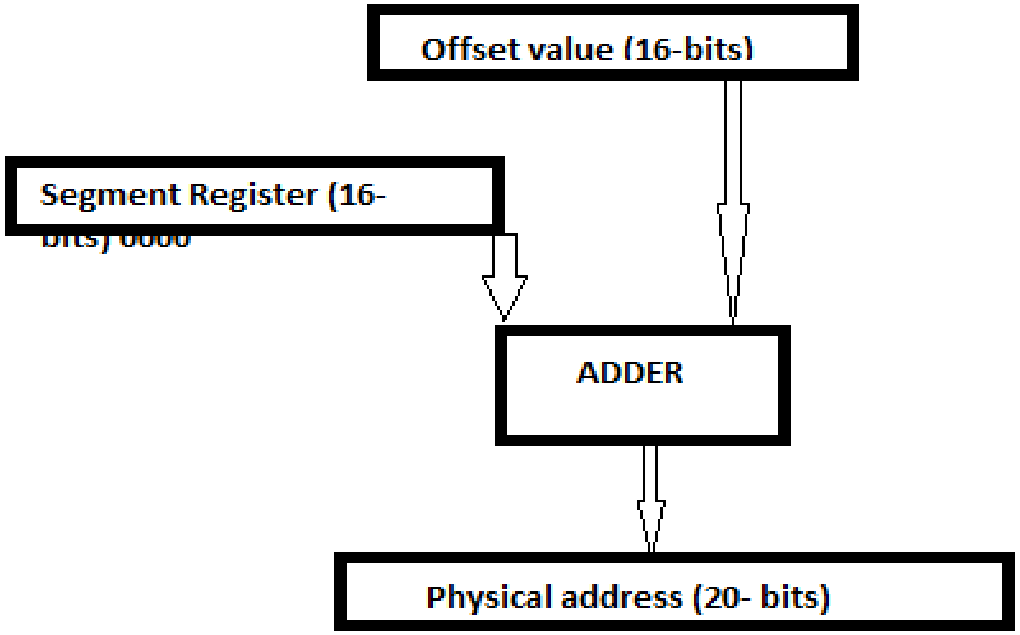
SS (Stack Segment) Register and SP (Stack Pointer) Register

- Upper 16-bits of the starting address of stack segment is stored in SS register.
- It is located in BIU.
- SP register holds a 16-bit offset from the start of stack segment to the top of the stack.
- It is located in EU.

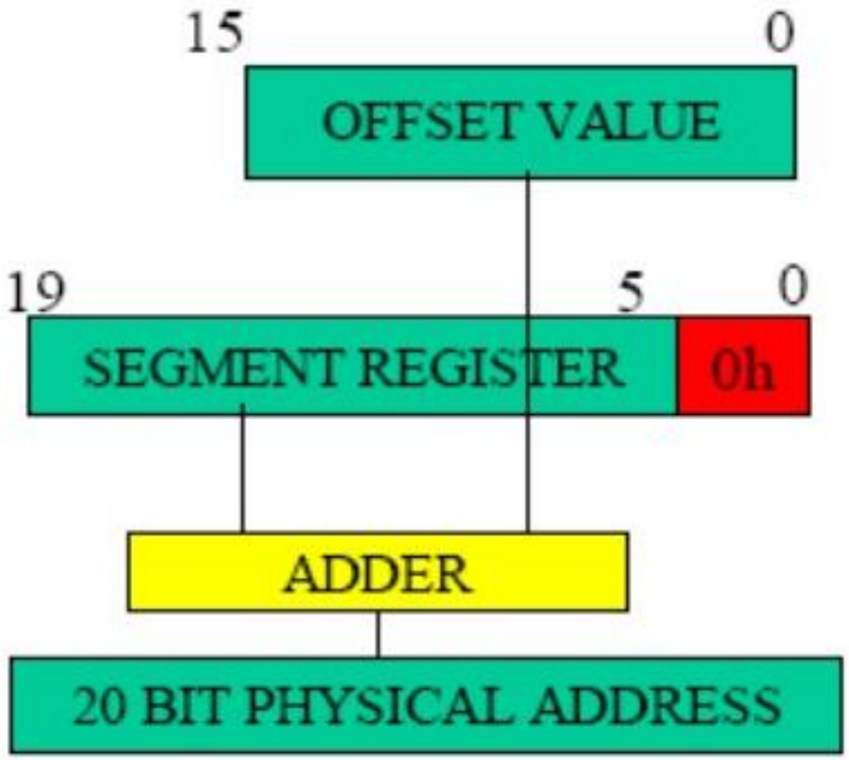
Other Pointer and Index Registers

- Base Pointer (BP) register
- Source index (SI) register
- Destination Index (DI) register
- Can be used for temporary storage of data.
- Main use is to hold a 16-bit offset of a data word in one of the segments.

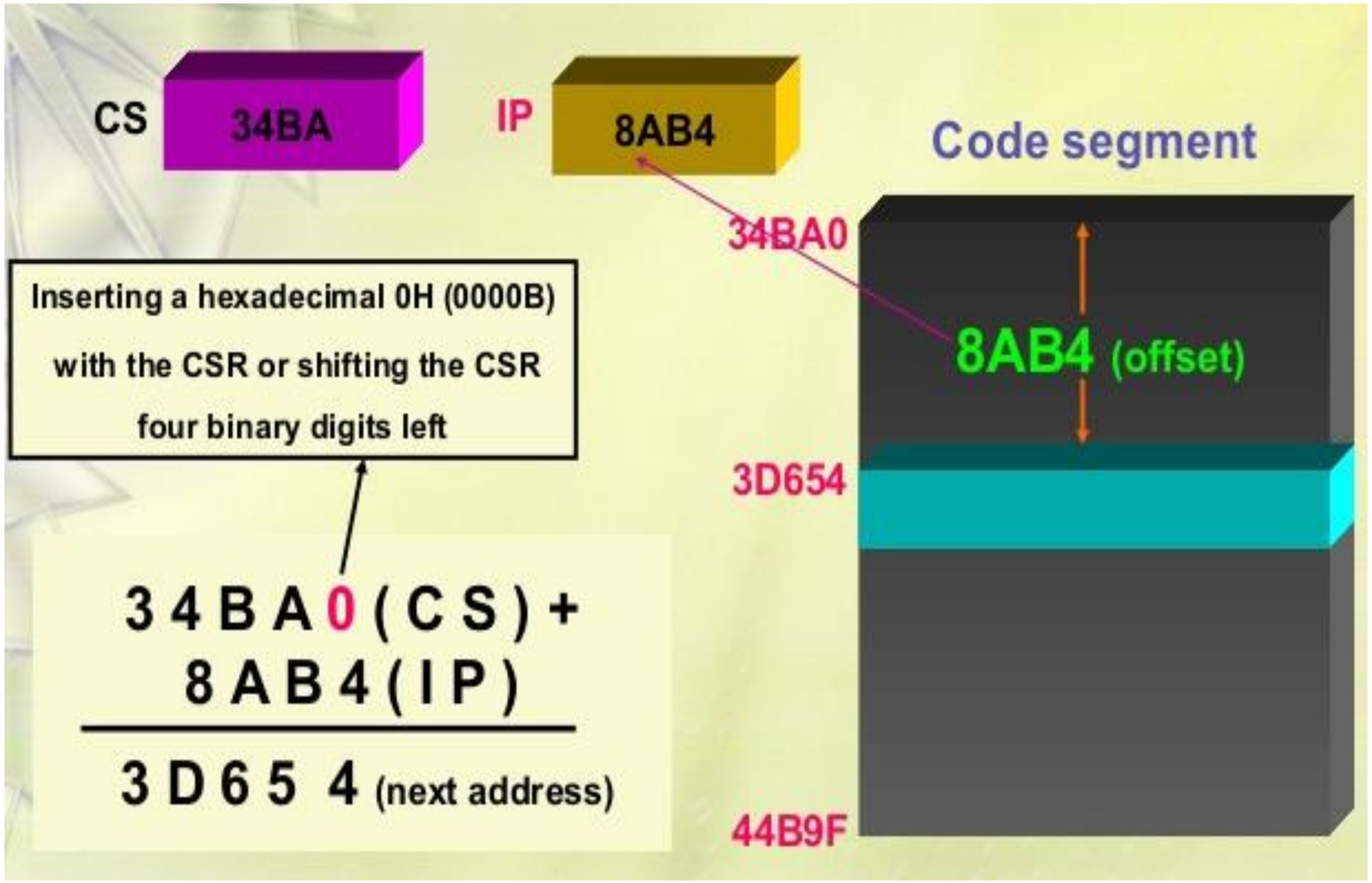
Memory Address Generation



Example


15
0
OFFSET VALUE
19
5
0
SEGMENT REGISTER
0h
ADDER
20 BIT PHYSICAL ADDRESS

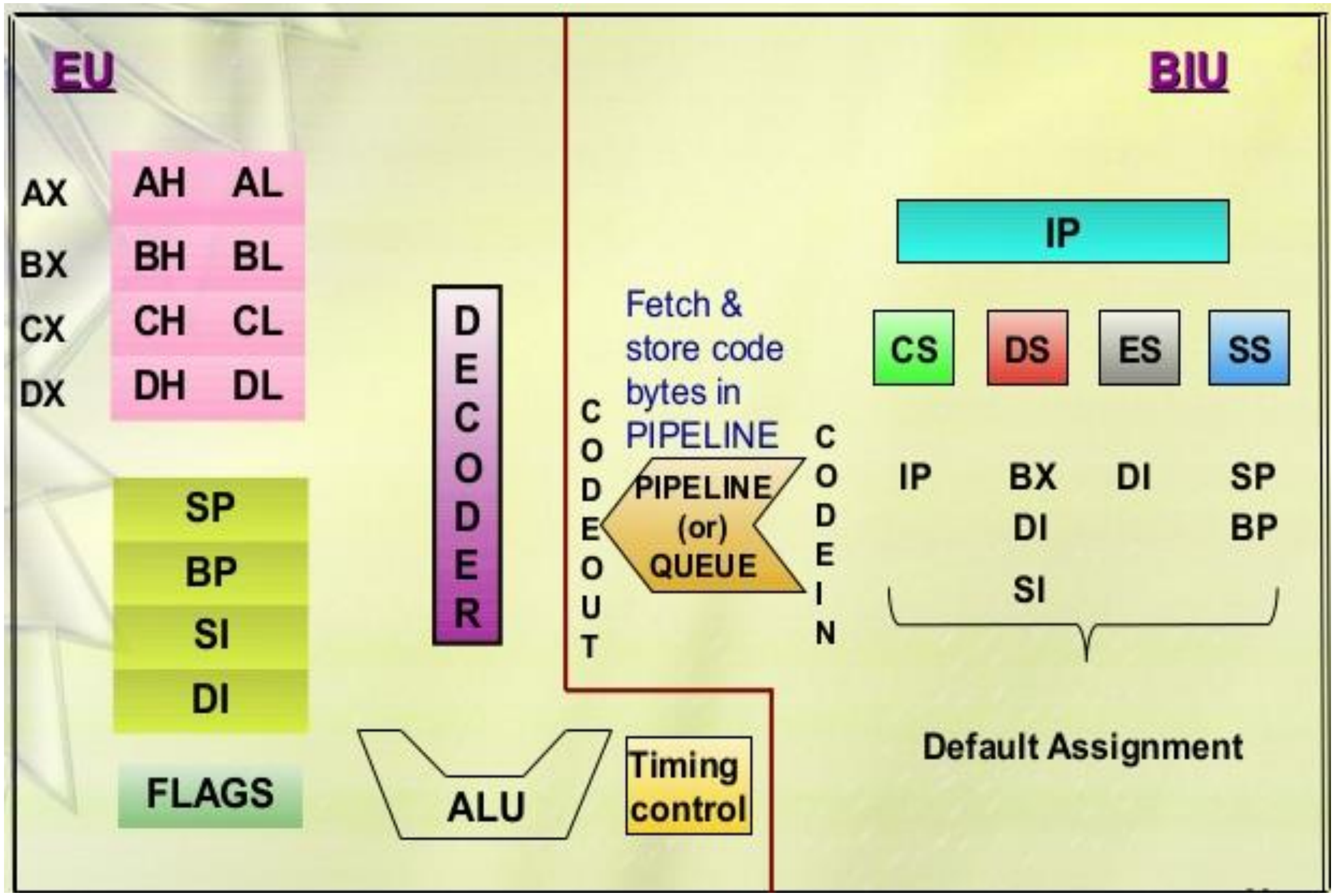
Example showing the CS:IP scheme of address formation



Segment and Address Register Combination

- CS:IP
- SS:SP – SS:BP
- DS:BX – DS:SI
- DS:DI (for other than string operations)
- ES:DI (for string operations)

Summary of Registers and Pipeline of 8086 Microprocessor



Instruction Set

8086 supports 6 types of instructions-

1. Data Transfer Instructions

   Mnemonics: MOV, XCHG, PUSH, POP, IN, OUT

2. Arithmetic Instructions

   Mnemonics: ADD, ADC, SUB, SBB, INC, DEC,MUL,DIV,CMP

3. Logical Instructions

   Mnemonics: AND, OR, XOR, TEST, SHR, SHL, RCR, RCL

Instruction Set

4. String Manipulation Instructions

   Mnemonics: REP, MOVS, CMPS,SCAS,LODS, STOS

5. Processor Control Instructions

   Mnemonics: STC, CMC, STD, CLD, STI,CLI, NOP, HLT, ESC, LOCK

6. Control Transfer Instructions

   Mnemonics: CALL, RET, JMP

ADDRESSING MODES

- The different ways in which a source operand is denoted in an instruction is known as addressing modes.

- There are 8 different addressing modes in 8086 programming.

1. Immediate addressing mode
2. Register addressing mode
3. Direct addressing mode
4. Register indirect addressing mode
5. Based addressing mode
6. Indexed addressing mode
7. Base-index addressing mode
8. Base-indexed with displacement mode

Addressing Modes (Contd.)

1. Immediate addressing mode: The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode.

2. Register addressing mode : It means that the register is the source of an operand for an instruction.

3. Direct addressing mode : the addressing mode in which the effective address of the memory location is written directly in the instruction..

4. Register indirect addressing mode : this addressing mode allows data to be addressed at any memory location through an offset address held in any of following registers : BP, BX, DI and SI.

Addressing Modes (Contd.)

5. Base addressing mode : In this addressing mode, the offset address of the operand is given by the sum of the contents of the BX/BP registers and 8-bit/16-bit displacement.

6. Indexed addressing mode : In this addressing mode, the operands offset address is found by adding the contents of SI or DI register and 8-bit/16-bit displacements.

7. Base-index addressing mode : In this addressing mode, the offset address of the operand is computed by summing the base register to the contents of an Index register.

8. Base indexed with displacement addressing mode : In this addressing mode, the operands offset is computed by adding the base register contents. An index register contains an 8-bit or 16-bit displacement.

APPLICATIONS OF 8086

- Gaming devices.
- Mobile phones, laptops and electronic gadgets.
- Traffic light controllers
- Home appliances like washing machines and microwave ovens.
- Frequency counters and synthesizers.
- Digital clocks.

THANK YOU