Digital Electronics and Microprocessors

Class 22

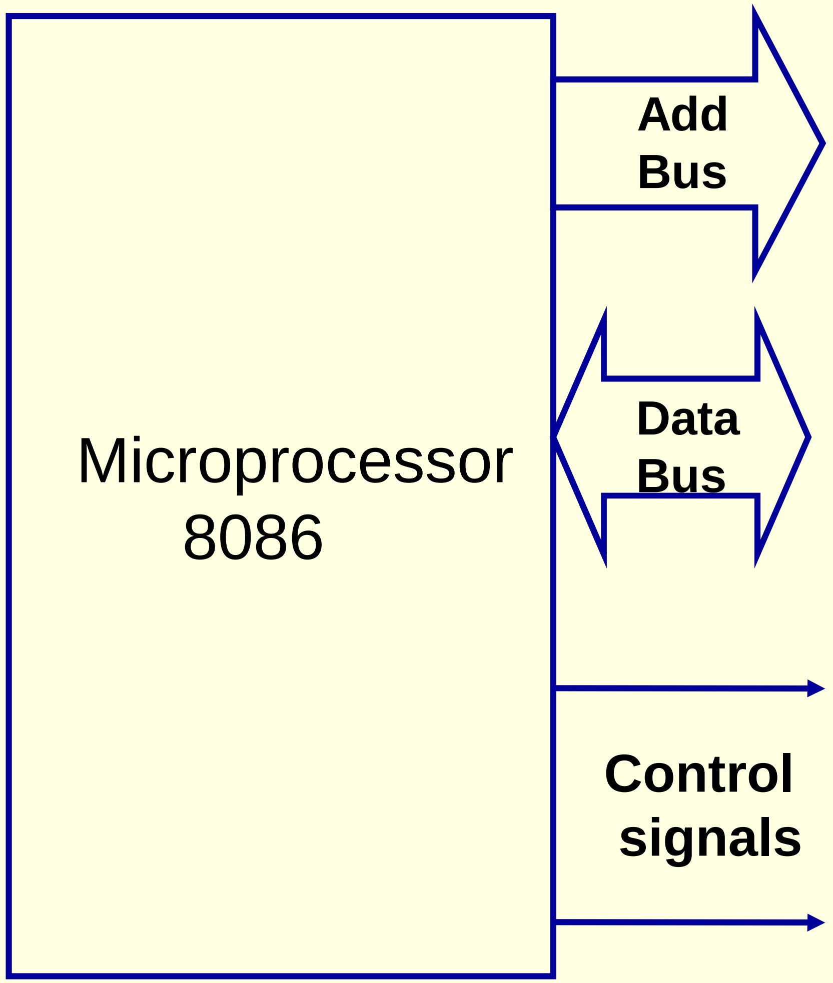
CHHAYADEVI BHAMARE

8086 microprocessor

Address Bus – 20 lines – A_{19} – A_0

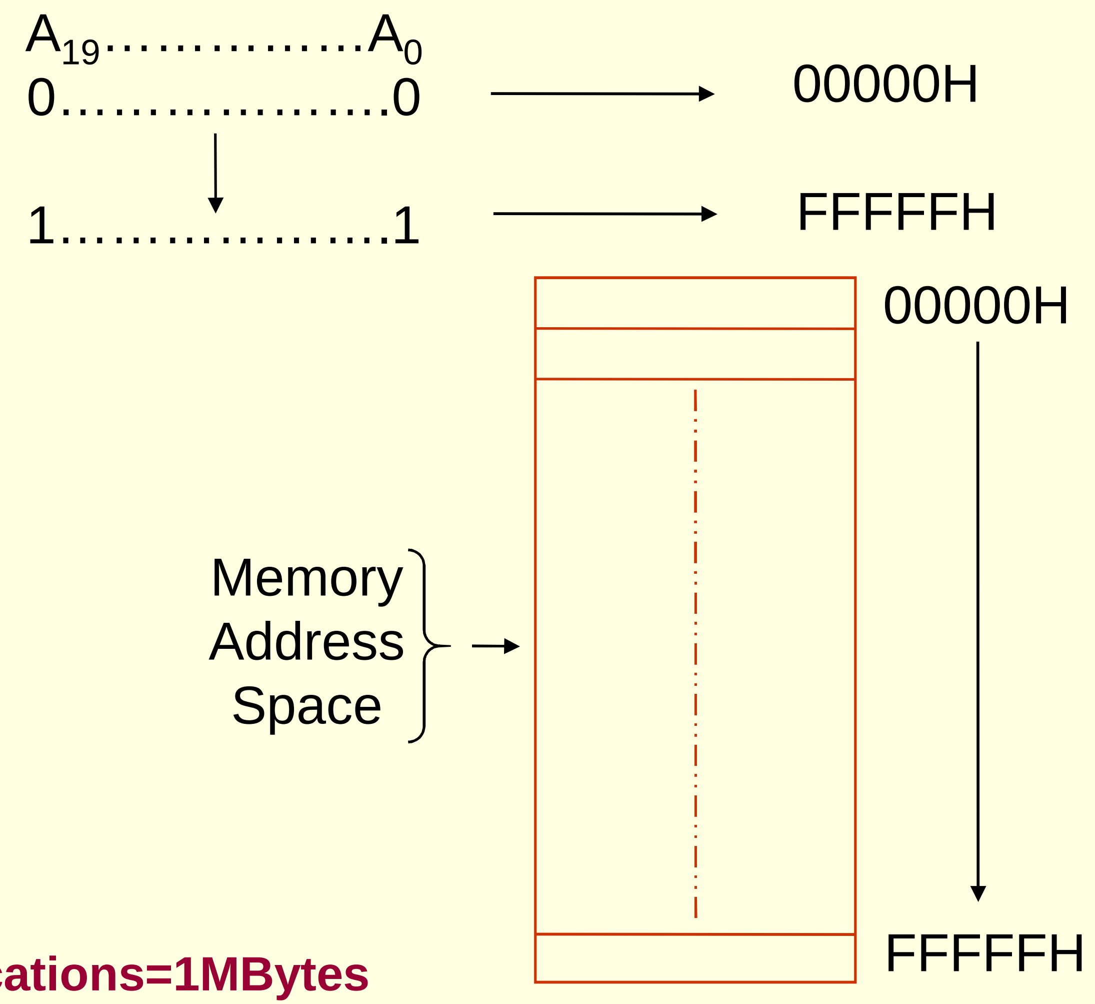
Data Bus – 16 lines – D_{15} – D_0

- 16 bit- microprocessor ?
- 16-bits data bus?



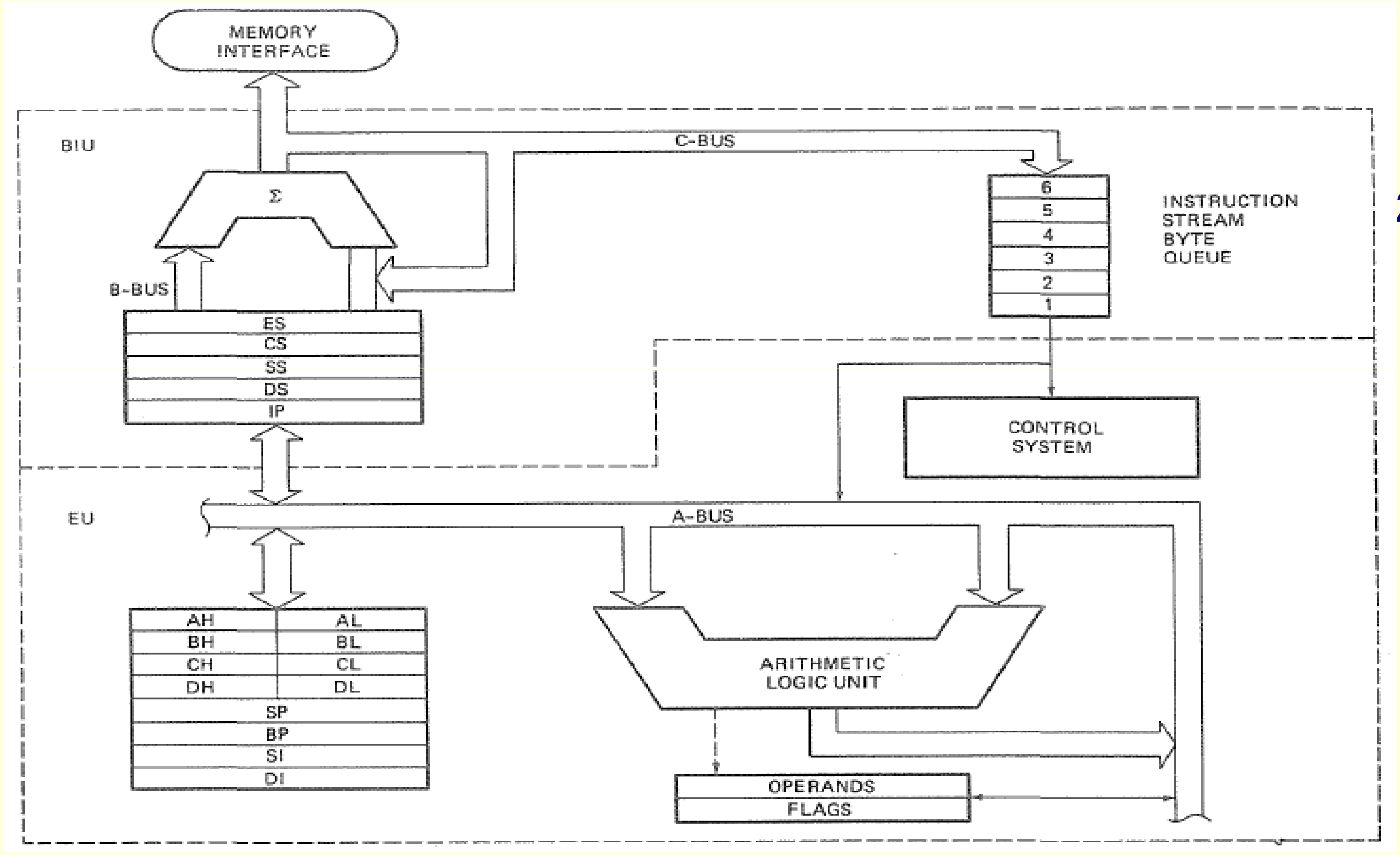
20 bits address bus?

- It can address any one of 1,048,576 (=2^{20}) memory locations/addresses.
- Each memory location is one byte wide.
- To store a word of 16 bit 2 memory locations are required.
- If the first byte of the word is at even address 8086 can read the entire word in one operation.
- If the first byte of the word is at an odd address, the 8086 will read the first byte with one bus operation and the second byte with another bus operation.



1,048,576 memory locations=1MBytes

8086 INTERNAL ARCHITECTURE



2 units are:

1. BIU

2. EU

Fig: 8086 Internal block diagram .

BIU and EU

- **BIU (bus interface unit)** sends out addresses, fetches instructions from memory, reads data from ports and memory, and writes data to ports and memory. In other words, the BIU handles all transfers of data and addresses on the buses for the execution unit.
- **EU (execution unit)** of the 8086 tells the BIU where to fetch instructions or data from, decodes instructions, and executes instructions.

EU (execution unit)

Major components are

- Control circuitry
- Instruction decoder
- ALU
- Flag register
- General purpose registers

Control circuitry

- Control circuitry directs internal operation

Decoder

- Decoder in EU translates instructions fetched from memory into series of actions which the EU carries out

ALU

- The EU has a 16-bit ALU which can add, subtract. AND, OR, XOR, Increment, Decrement, complement, shift binary numbers.

Flag Register

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

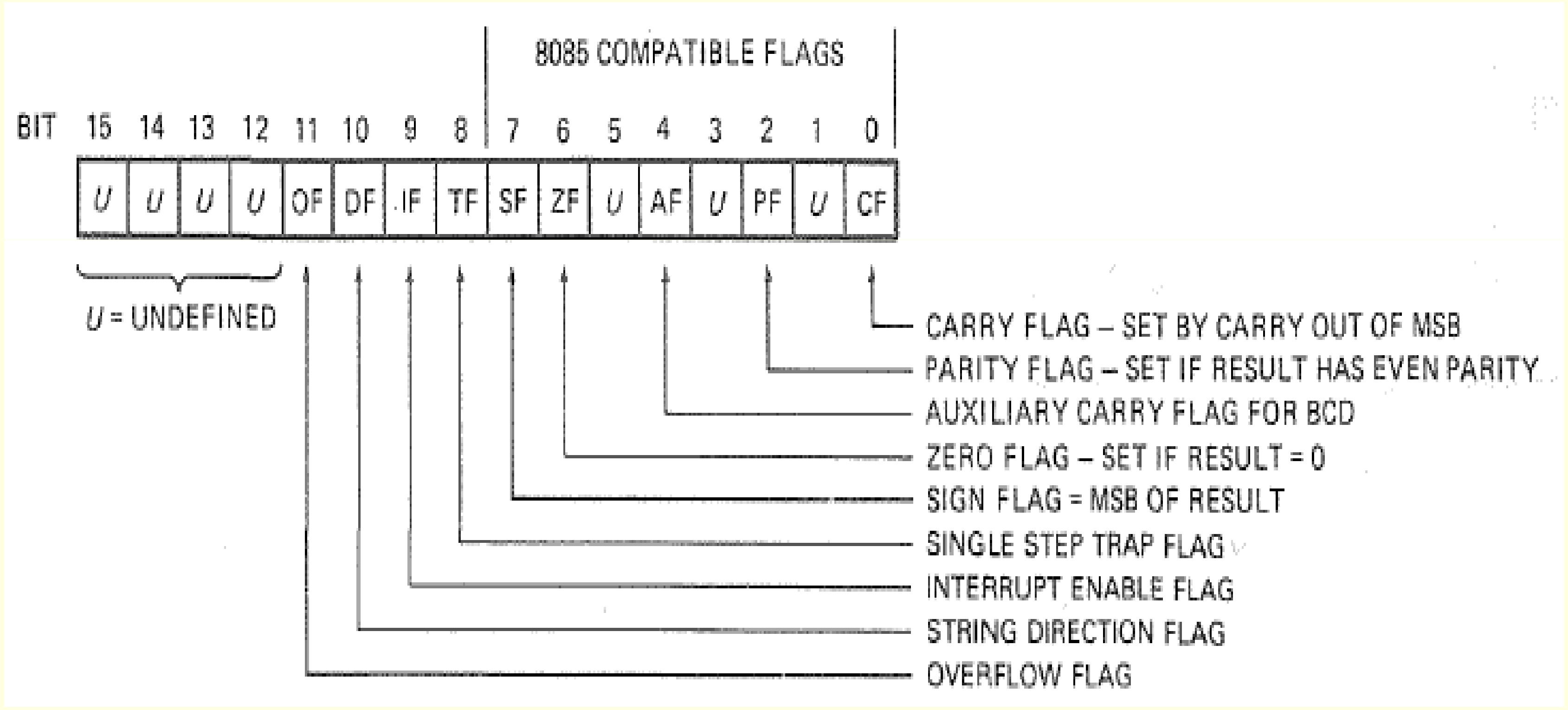

8085 COMPATIBLE FLAGS
BIT 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
U U U U OF DF IF TF SF ZF U AF U PF U CF
U = UNDEFINED
CARRY FLAG – SET BY CARRY OUT OF MSB
PARITY FLAG – SET IF RESULT HAS EVEN PARITY
AUXILIARY CARRY FLAG FOR BCD
ZERO FLAG – SET IF RESULT = 0
SIGN FLAG = MSB OF RESULT
SINGLE STEP TRAP FLAG
INTERRUPT ENABLE FLAG
STRING DIRECTION FLAG
OVERFLOW FLAG

Fig: Flag register

Contd..

Flags:

- -Conditional Flags
- -Control Flags

Conditional Flags:

- C (Carry)
- P (Parity)
- A (Auxiliary Carry)
- Z (Zero)
- S (Sign)
- O (Overflow)

Control Flags:

- T (Trap)
- I (Interrupt)
- D (Direction)

General purpose registers

- 8 GP registers are there
- Each GP register is of 8 bits.
- To store 16 bit data pairs of registers can be used as shown here.
- GP registers are used for temporary data storage.

AX	→	AH (8 bit)	AL (8 bit)
(Accumulator)			
BX	→	BH	BL
(Base Register)			
CX	→	CH	CL
(Used as a counter)			
DX	→	DH	DL
(Used to point to data in I/O operations)			

BIU (bus interface unit)

Major Components are

- Queue
- Segment registers and IP register

The Queue

- While the EU is decoding an instruction or executing an instruction which does not require use of the buses, the BIU fetches up to six instruction bytes for the following instructions. The BIU stores these prefetched bytes in a first-in—first-out register set called a queue.
- This prefetch-and-queue scheme greatly speeds up processing.
- Except in the cases of JMP and CALL instructions, where the queue must be dumped and then reloaded starting from a new address.
- Fetching the next instruction while the current instruction executes is called pipelining.

Segment registers and Instruction Pointer (IP)

- **1. Code Segment (CS) register, 2. Data Segment (DS) register, 3. Stack Segment (SS) register and 4. Extra Segment (ES) register.**
- IP register holds the 16-bit address, or offset, of the next code byte within code segment.
- Contents of IP decide which instruction will be executed next.

There are two types of addressing schemes:
1. An Absolute Address, such as 04A26H, is a 20 bit value that directly references a specific location.
2. A Segment Offset Address, combines the starting address of a segment with an offset value.

Memory organization

- At any given time the 8086 works with only four 65,536-byte (64-Kbyte) segments within the 1,048,576- byte (1-Mbyte) range.
- Four segment registers in the BIU are used to hold the upper 16 bits of the starting addresses of four memory segments that the 8086 is working with at a particular time.

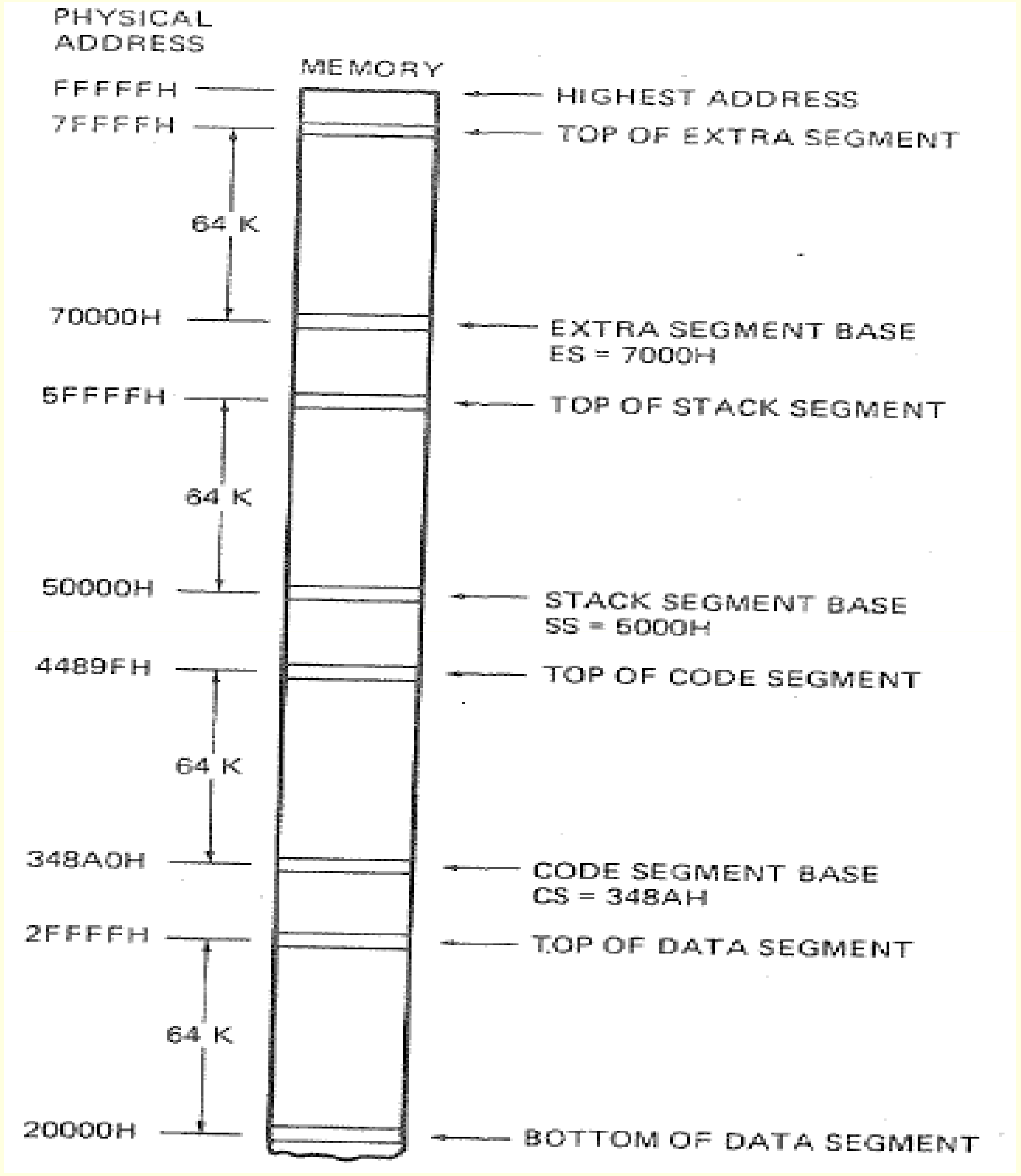
8086 introduced memory segmentation to overcome the 16-bit addressing barrier of earlier chips.

Segments and Addressing
Segments are special area defined in a program for containing the code, the data, and the stack. **Segment Offset within a program, all** memory locations within a segment are relative to the segment starting address. The distance in bytes from the segment address to another location within the segment is expressed as an **offset (or** displacement).
To reference any memory location in a segment, the processor combine the segment address in a segment register with the offset value of that location, that is, its distance in byte from the start of the segment.
Specifying addresses
To represent a segment address and its relative offset we use the notation:

Memory Segmentation

Fig:

One way four 64-Kbytes segments might be positioned within 1MByte address space of 8086.



This constraint was put on the location of segments so that it is only necessary to store and manipulate 16-bit numbers when working with the starting address of a segment.

Physical address calculation

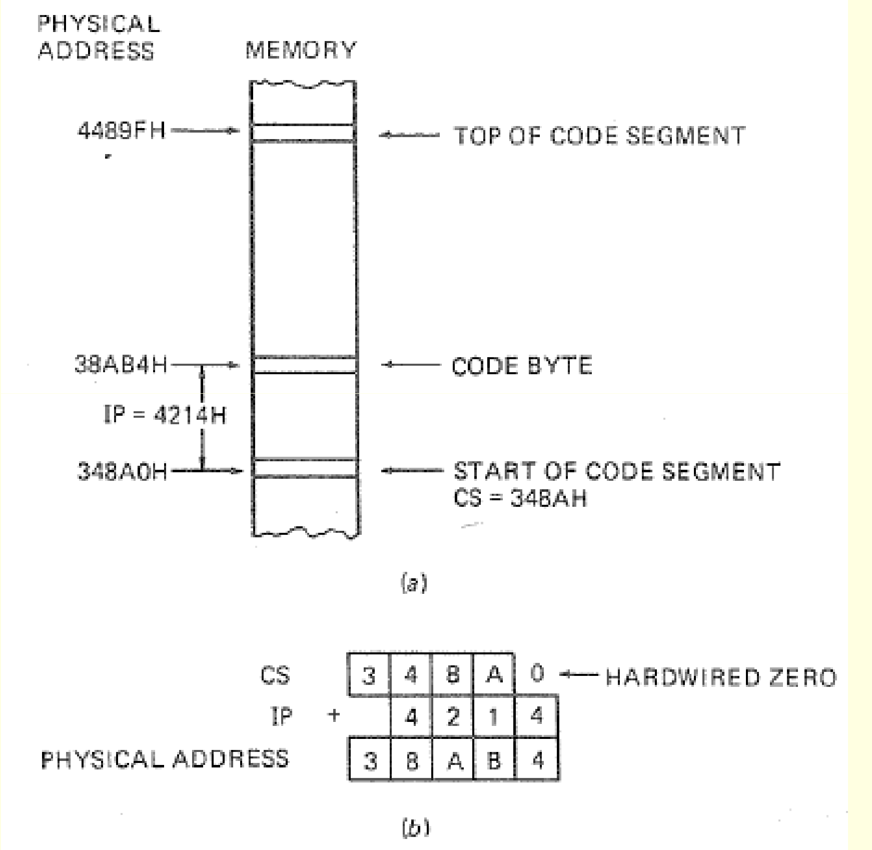


FIGURE 2-10 Addition of IP to CS to produce the physical address of the code byte. (*a*) Diagram. (*b*) Computation.

Specifying addresses

To represent a segment address and its relative offset we use the notation:

Segment: offset

Thus **34BA:4214 denotes offset 4214H from segment 34BAH.**

The actual address it refers to is obtained in the following way:

1- Add zero to the right hand side of the segment address.

2- Add to this the offset.

Hence the actual address referred to by **34BA:4214** is 38AB4 H

Contd..

CS = 2000H $\leftarrow$ Base address

IP = 3000H $\leftarrow$ Offset address

2000H : 3000H

Physical address

$$
\begin{array}{r}
= \quad 20000H \;+ \\
3000H \\
\hline
23000H \\
\hline
\end{array}
$$

8086 – Default 16 bit segment and offset address combinations

Segment	offset	special purpose
CS	IP	Instruction Address
SS	SP	Stack address
DS	BX,DI,SI an 8-bit number 16 – bit number	Data address
ES	DI for string Instructions	String destination address

SS and SP Reg.

A stack is a section of memory set aside to store addresses and data while a subprogram executes.
The stack segment register is used to hold the upper 16 bits of the starting address for the program stack.

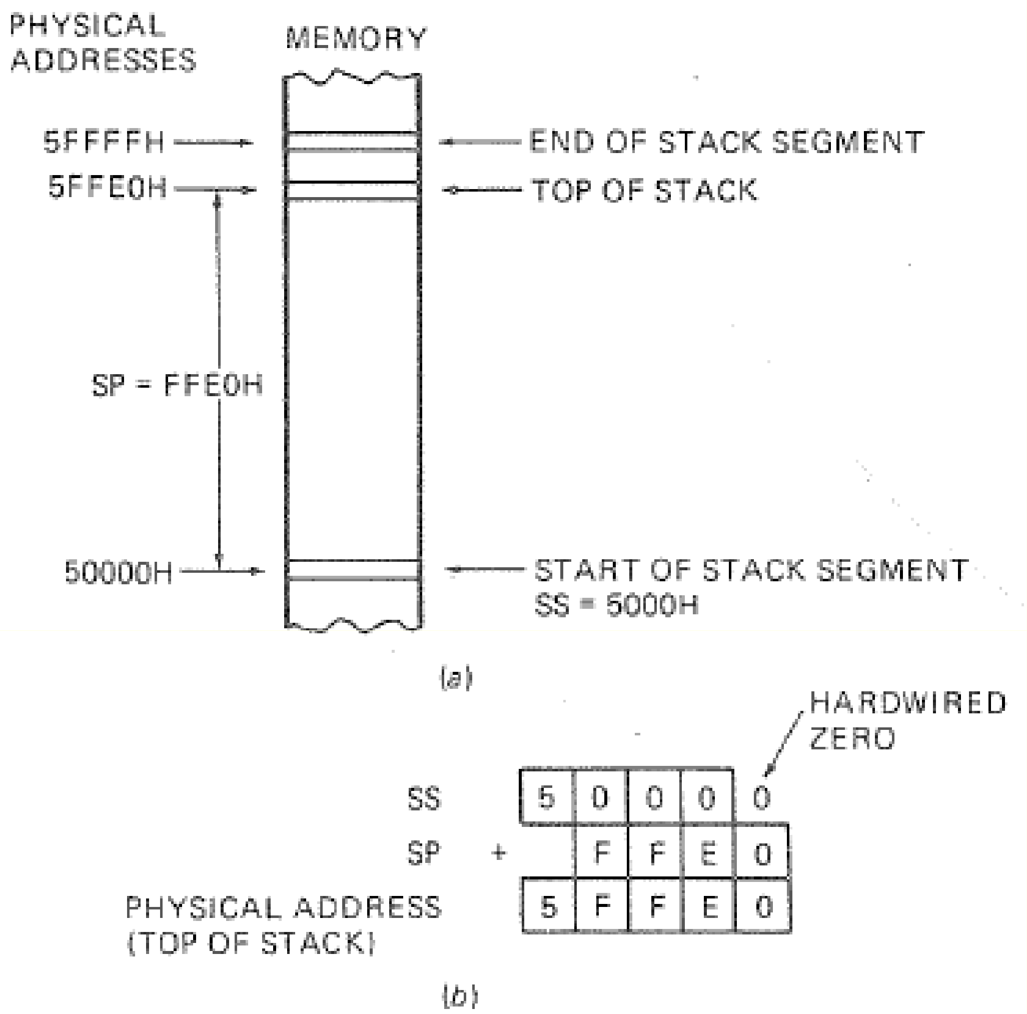


FIGURE 2-11 Addition of SS and SP to produce the physical address of the top of the stack. (*a*) Diagram. (*b*) Computation.

Why Intel went for segment:offset approach?

- 8086 has to manipulate and store only 16-bit quantities instead of 20-bit quantities. This makes for an easier interface with 8- and 16-bit-wide memory boards and with the 16-bit registers in the 8086.

Pointer and Index Registers

- EU contains a 16-bit base pointer (BP) register. It also contains a 16-bit source index (SI) register and a 16-bit destination index (DI) register.
- BP, SI and DI: These three registers can be used for temporary storage of data just as the general-purpose registers described before. However, their main use is to hold the 16-bit offset of a data word in one of the segments.
- SI, for example. can be used to hold the offset of a data word in the data segment.