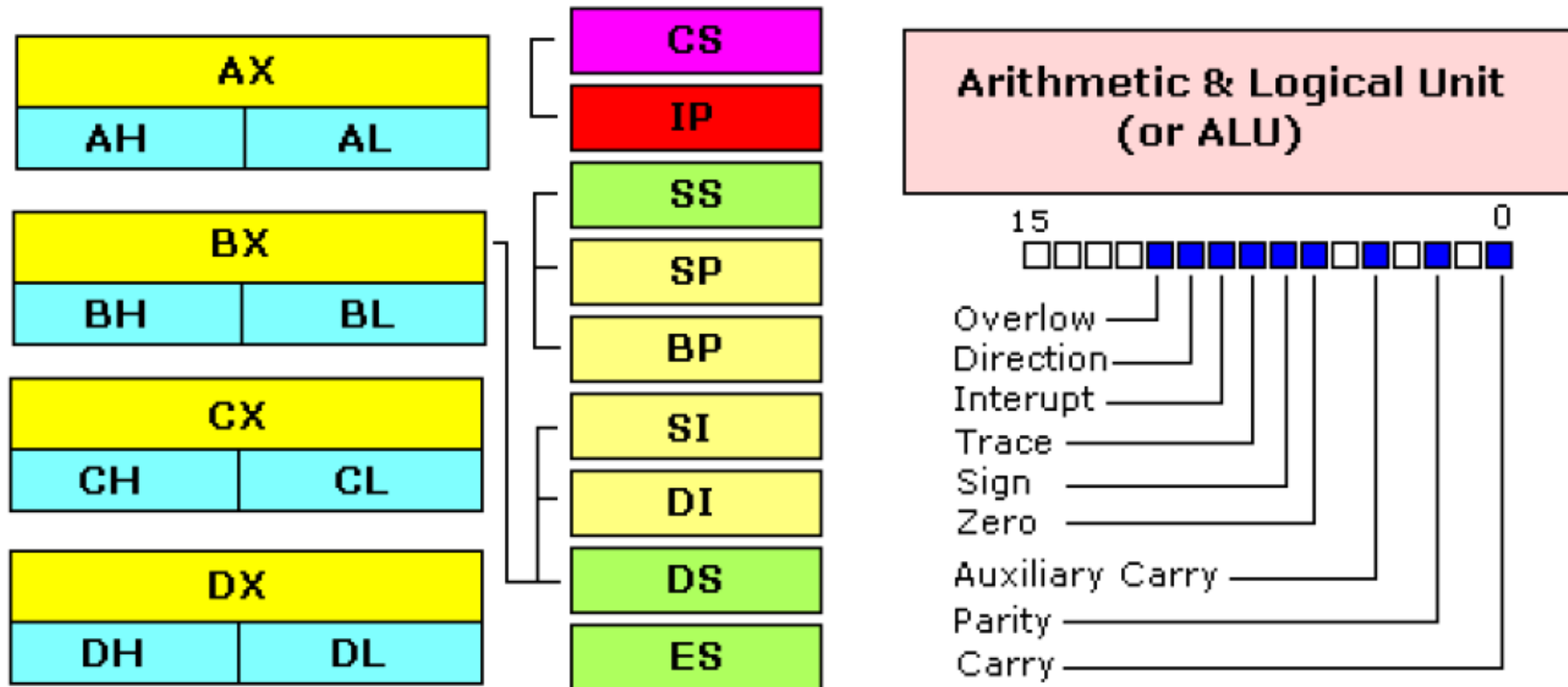


Lab 2

Main Memory
Data Transfer Instructions (Mov Inst.)
Logical & Physical Address

Inside the CPU

Central Processing Unit (or CPU)



Inside the CPU

- 1. GENERAL PURPOSE REGISTERS**
- 2. SEGMENT REGISTERS**
- 3. SPECIAL PURPOSE REGISTERS**

GENERAL PURPOSE REGISTERS

8086 CPU has 8 general purpose registers, each register has its own name:

- **AX** - the **Accumulator** register (divided into **AH / AL**).
- **BX** - the **Base Address** register (divided into **BH / BL**).
- **CX** - the **Count** register (divided into **CH / CL**).
- **DX** - the **Data** register (divided into **DH / DL**).
- **SI** - **Source Index** register.
- **DI** - **Destination Index** register.
- **BP** - **Base Pointer**.
- **SP** - **Stack Pointer**.

SEGMENT REGISTERS

- **CS - Code Segment** points at the segment containing the current program.
- **DS - Data Segment** generally points at segment where variables and data are defined.
- **ES - Extra segment** register, it's up to a coder to define its usage.
- **SS - Stack Segment** points at the segment containing the stack.

SPECIAL PURPOSE REGISTERS

- **IP** - the instruction pointer. it points to currently executing
- **Flags Register** - determines the current state of the processor.

Main Memory and Memory Access

To access memory we can use these four registers: **BX, SI, DI, BP**.

Combining these registers inside [] symbols, we can get different memory locations. These combinations are supported (addressing modes):

[BX + SI] [BX + DI] [BP + SI] [BP + DI]	[SI] [DI] d16 (variable offset only) [BX]	[BX + SI] + d8 [BX + DI] + d8 [BP + SI] + d8 [BP + DI] + d8
[SI] + d8 [DI] + d8 [BP] + d8 [BX] + d8	[BX + SI] + d16 [BX + DI] + d16 [BP + SI] + d16 [BP + DI] + d16	[SI] + d16 [DI] + d16 [BP] + d16 [BX] + d16

d8 - stays for 8 bit displacement.

d16 - stays for 16 bit displacement.

For Easy remember

There is an easy way to remember all those possible combinations using this chart:

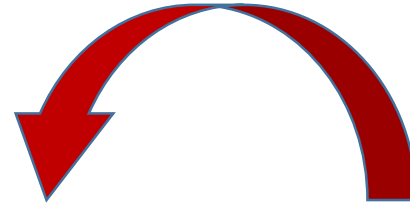
BX	SI	+ disp
BP	DI	

Data Transfer Instructions

MOV instruction

MOV instruction

The MOV instruction: **copies** a word or byte of data from a specified source to a specified destination.



MOV Destination, Source

MOV instruction

These types of operands are supported:

MOV REG, memory

MOV memory, REG

MOV REG, REG

MOV memory, immediate

MOV REG, immediate

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

memory: [BX]

immediate: 5, -24, 3Fh, 10001101b, etc...

MOV instruction

For segment registers only these types of **MOV** are supported:

MOV SREG, memory

MOV memory, SREG

MOV REG, SREG

MOV SREG, REG

SREG: DS, ES, SS, and only as second operand: CS.

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

memory: [BX]

MOV instruction

Notes:

- The source and destination **cannot** both be memory locations.
- The source and destination must both be of the same type (bytes or words).
- MOV instruction does not affect any flag.

MOV instruction

MOV CX, 037AH ; Put immediate number 037AH to CX

MOV AL, CL

ret

MOV instruction

MOV [437AH],55h

MOV BL,[437AH]

Ret

MOV instruction

```
MOV [BX],1234H ; explain how the value store in main memory  
MOV AL,[BX]  
MOV DL,[BX+1]  
RET
```

MOV instruction

```
MOV [BP+SI+88H],6654H
```

```
MOV BL,[BP+SI+88H]
```

```
RET
```

MOV instruction

MOV [1234h], [2234h] ; Try this, what the result?

ret

MOV instruction

MOV AL, 1234h ;Try this, what the result

MOV AH, 34h

Ret

MOV instruction

MOV CH, 01011111b ; set CH to binary value.

Ret ; remove b

; try only 0101 MOV CH, 0101

MOV instruction

MOV BX,1234H ; how to put value in segment register?

MOV DS,BX

RET

MOV instruction

MOV [BX],1234H ; explain how the value store in main memory

MOV [BX+2],1215H

MOV AL,[BX]

MOV DL,[BX+1]

MOV AH,[BX+2]

MOV DH,[BX+3] ; try +5

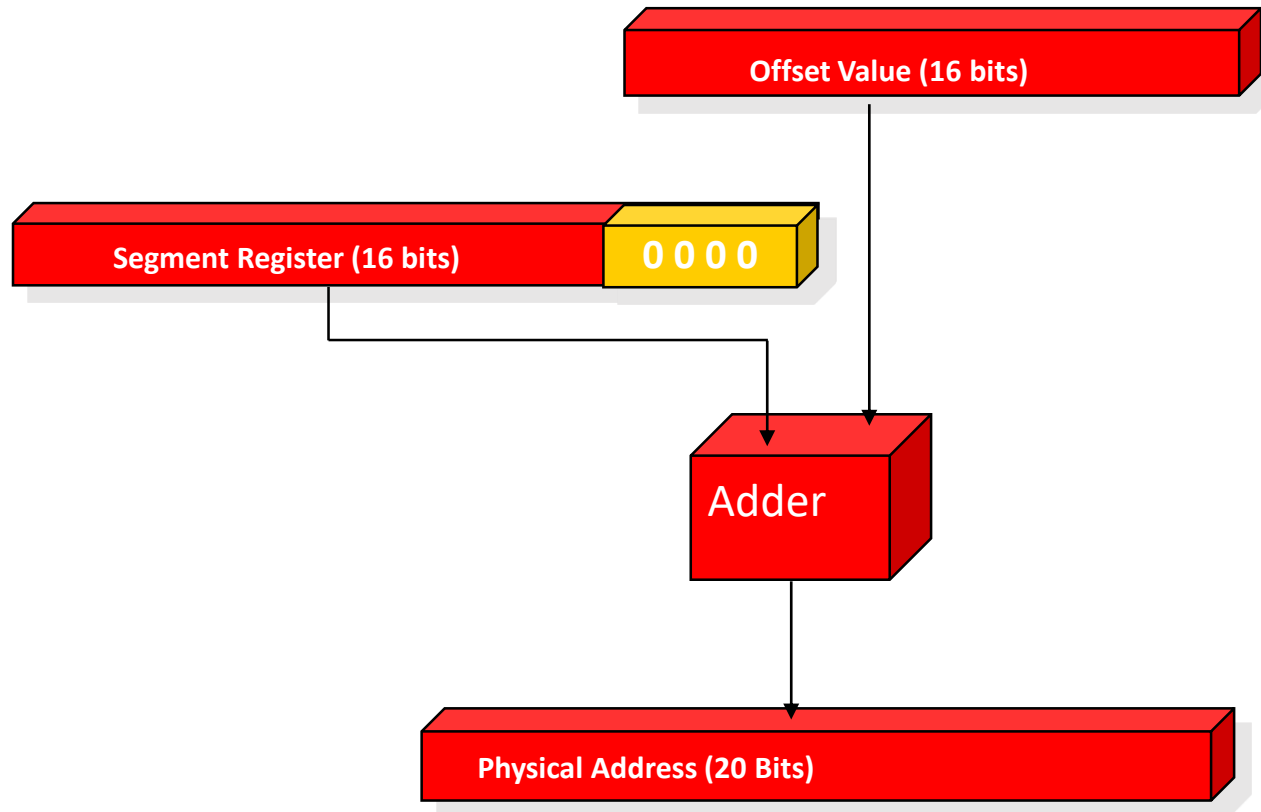
RET

MOV instruction

Q1: Load 34CDh in to SS register?

Q2: Copy the lower 8bit from BX to high bits in AX?

Physical Address and logical address



Physical Address and logical address

The value in segment register (**CS, DS, SS, ES**) is called a "**segment**" and the value in purpose register (**BX, SI, DI, BP**) is called an "**offset**".

Logical address : Segment : offset

Ex: DS contains value **1234h**, SI contains the value **7890h**

Logical address: 1234:7890

Physical address = (Segment base*10H) + Offset Value.

$$= 1234h * 10h + 7890h = 19BD0h$$

Physical Address and logical address

Ex: find the physical address of a given

Logical address: A4FBH:4872H ?