Prepared By: Mohamed Fathallah
Memory and Buses

First, we start by general Talk about the subject and Machine language, assembly language and high-level language.

Roller-coaster Tycoon

Buses in Memory and Microprocessors

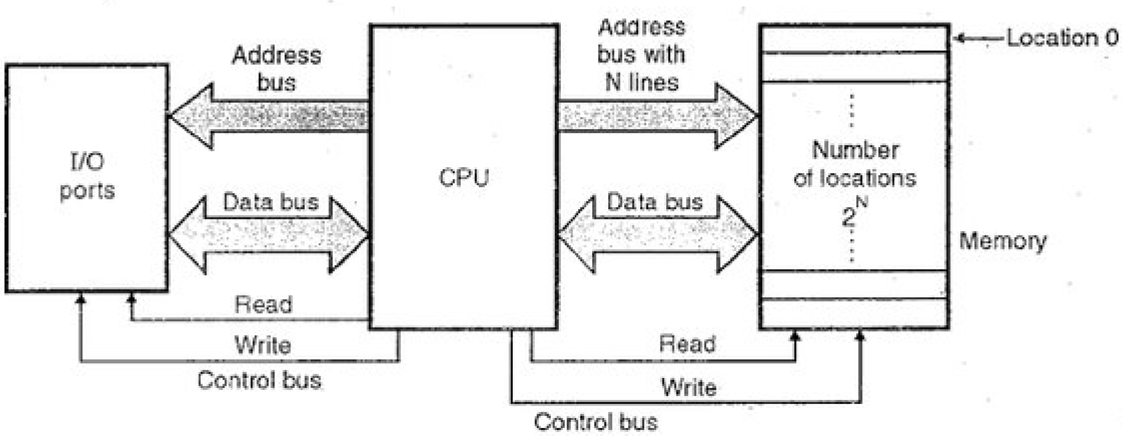


Fig. 1.2.3 : The three types of buses and their utility

Number of Memory Locations (Memory size) are determined by address buss and it is numbered in bytes for 8086.

If address bus is 20bit then you have 2^20 location (memory size)

So, the first memory address is 0, and the last memory address is 1,048,575 Bytes (since we start counting from 0).

The first memory address is 0x00000.

The last memory address is 0xFFFFF.

-Extra check one's complement and two's complements in c# and how computer deals with negative numbers.

(Negative numbers)

Question:

If You have memory of size 2mb, what is the address bus size?
(Remember everything should be in bytes)

Solution:

Memory Size= 2^number of bits in address bus

2x1024x1024=2097152 byte
using x^#of bytes =memory-size (locations)

Of bits=log (2097152)

________________Word and little endian____________________

1897H   this will be stored in memory starting from 7 for position 0H, 8 for position 1H and so on. (Different frameworks uses different endian).

_____________________Data Bus____________________

Transfer memory Data between Processor and I/O and memory.

16bit bus transfer 16bit of information at a time (clock cycle). Fetch. Or Store.

If data bus is 64bit for example it is better it will execute faster.

_____________________Control Signals____________________

The processors send address using address bus.

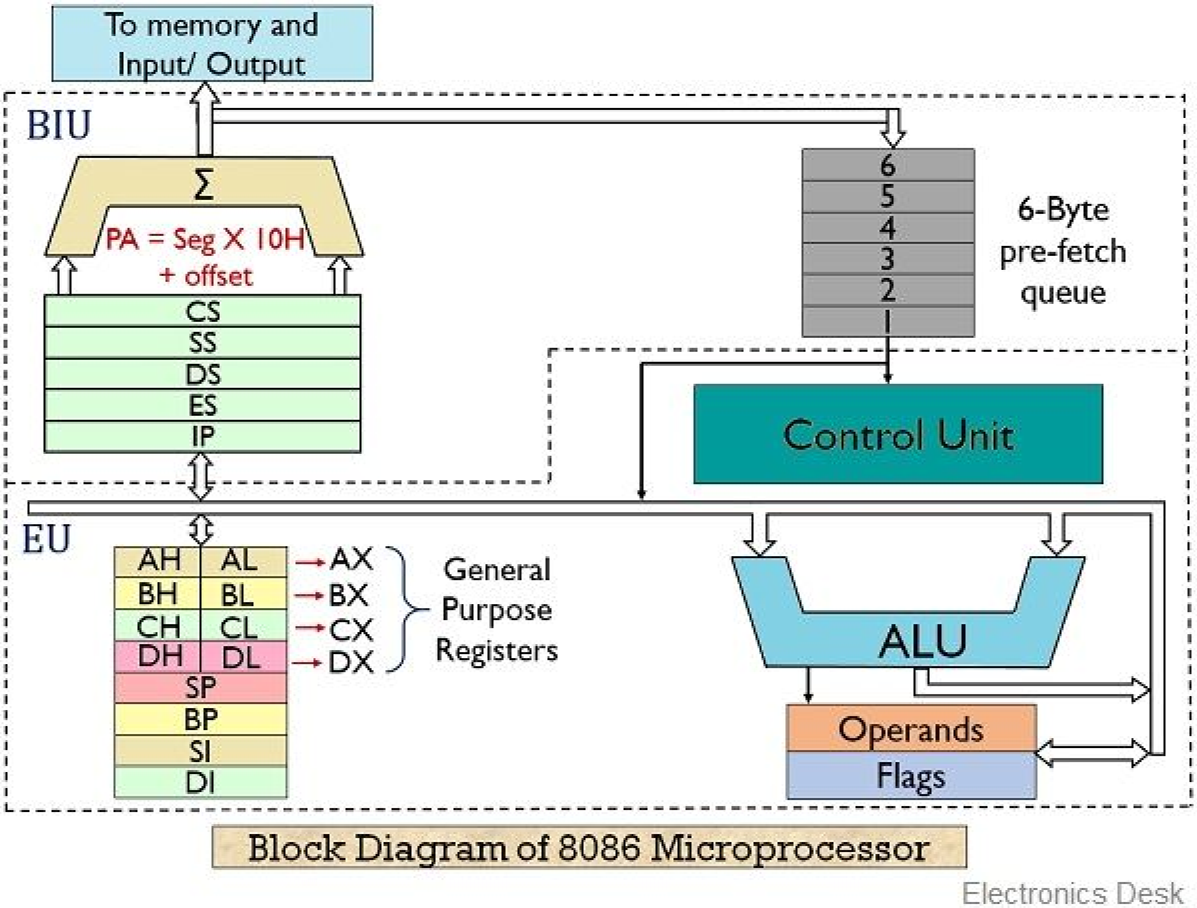
And in the same time, he sends MRDC signal to tell memory he wants to read from this address. (And so on with any operation)

----------------2024 Updates (Fetch-decode-execute cycle) -------------------

- **https://www.youtube.com/watch?v=Z5JC9Ve1sfI&t=441s&ab_channel=TomScott**
- **https://www.youtube.com/watch?v=jFDMZpkUWCw&ab_channel=ComputerScienceLessons**

_____________ Registers & virtual memory access _____________

(14) Registers in 8086 ,8088 ,80286 **(Let's Watch Video before this the usage of Registers)**



- **General Purpose Registers: (AX, BX, CX, DX) + SI, DI source and destination index. And BP (Base pointer)**

These registers are primarily used for ***arithmetic*** *and* ***data movement.***

AX: for arithmetic and I/O

BX: for data movement. (Data segment)

CX: used in loops as counter

DX: used with accumulator in arithmetic operation (multiplication, division) & I/O

SI: used for addressing the data memory (string instructions) – indirectly. -(Data segment)

DI: Same as SI -(Data segment)

BP: addressing but in stack memory only.

- **Specific purpose Registers: SP (stack Pointer), IP (Instruction pointer). Flag**

SP (Stack Pointer register): point to top of stack memory.

IP (Instruction pointer register): point to next instruction in the code segment.

Flags: Show the status of Microprocessor and do some controls.

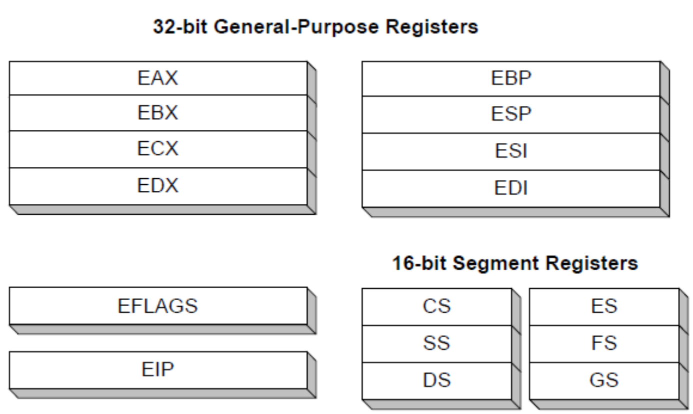
- **Segment Registers: CS, SS, DS, ES**

CS Register – code segment – determine the start of code segment in memory.

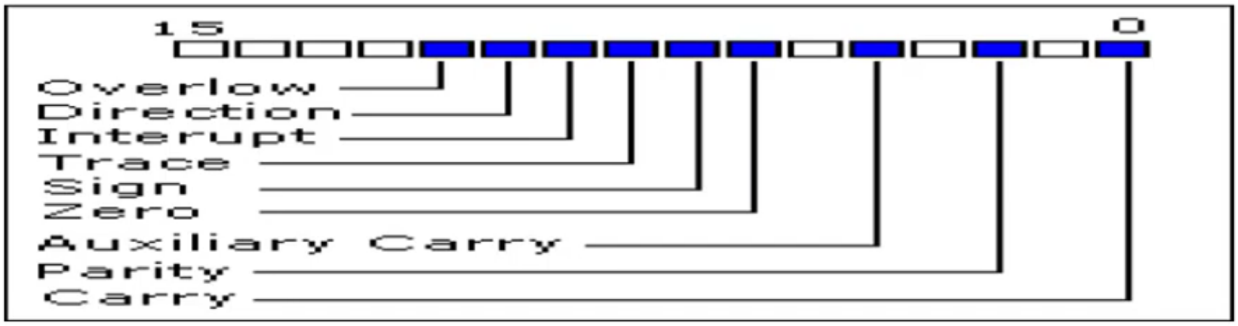
DS Register – data segment - …

SS Register – stack segment - …

ES Register - Extra segment - …



- **E for 32bit R for 64 bits**



Flag Register- 9bits each bit in this register determines a flag in the microprocessor, some bit even control the Microprocessor.

Status Flags: reflects the result of arithmetic or logical operations. 0 ,2 ,4,6,7,11

Control Flags: 8 ,9, 10, Trace, interrupt, direction.

---

1. **CF (Carry Flag)-unsigned overflow: 1 indicate overflow from adding or subtracting.**
2. **PF(Parity Flag): 1 indicate that the result number is even , 0 is odd.**
3. **AF(Auxiliary flag): used to indicate unsigned-overflow results from addition or subtraction operation between bit position 3 and 4 of the result.**
4. **ZF(Zero Flag): 1 indicate the result of arithmetic operation is 0**
5. **SF(Sign Flag): 1 indicate the result of the number is –**
6. **OF(Overflow Flag): 1 indicate of overflow and 0 for no overflow.**
7. **DF(Direction Flag): control the direction of the loop (decrement or increment) of string instructions (STD (set direction) or CLD (clear direction)).**
8. **IF(Interrupt flag): enable interrupt (STI (set interrupt), CLI (clear interrupt))**
9. **TF(Trap Flag): used to start debugging.**

Status Flags

- *Carry: unsigned arithmetic out of range*
- *Overflow: signed arithmetic out of range*
- *Sign: result is negative*
- *Zero: result is zero*

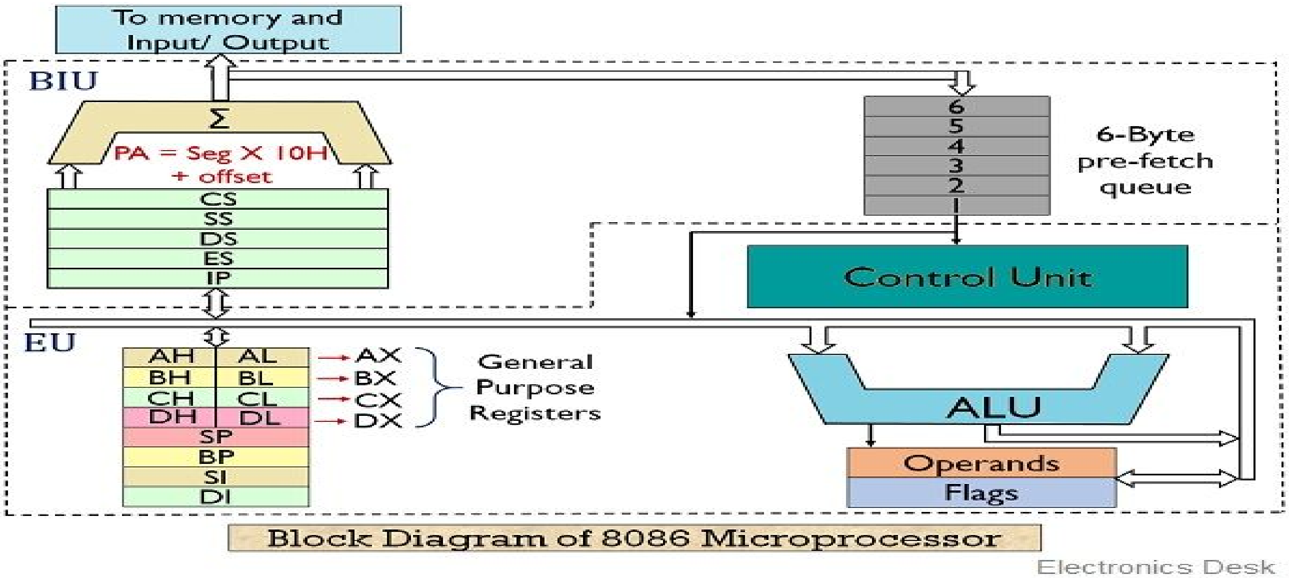
Example: Zero Flag

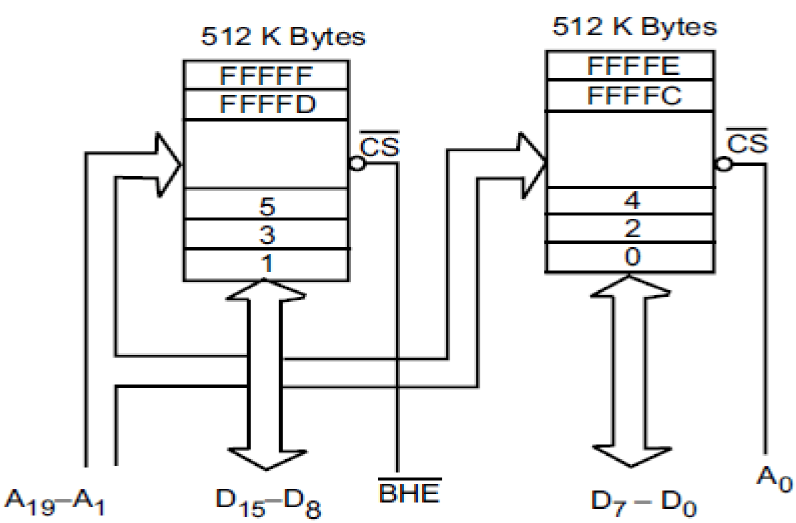
```
                    ; initially, assume ZF = 0
mov   AL,55H        ; ZF is still zero
sub   AL,55H        ; result is 0
                    ; ZF is set (ZF = 1)
mov   CX,0          ; ZF remains 1
inc   CX            ; result is 1
                    ; ZF is cleared (ZF = 0)
```

Summary of registers: (we focus only on 16bit registers not the 32bit)

32-Bit registers	Name	16- and 8-bit sub-registers	Brief description and/or primary use
eax	Accumulator	ax, ah, al	Arithmetic and logic
ebx	Base	bx, bh, bl	Arrays
ecx	Counter	cx, ch, cl	Loops
edx	Data	dx, dh, dl	Arithmetic
esi	Source index	si	Strings and arrays
edi	Destination index	di	Strings and arrays
esp	Stack pointer	sp	Top of stack
ebp	Base pointer	bp	Stack base
eip	Instruction pointer	ip	Points to next instruction
eflags	Flag	flags	Status and control flags

______ Virtual memory access (Starting in Lab 2) _______________





- Its 20-bit address bus can address 1MB of memory, **it segments it into 16 64kB segments.**
- 8086 works only with **four 64KB segments** within the whole 1MB memory.

8086 microprocessor works with physical memory directly, this not good idea to do but this how it all started.

The problem here is memory is 20bit and Memory segment Registers are 16bit this mean they can used only to access 2^16 not 2^20 , thus why we multiple 2^16 by 2^4 or 10H.

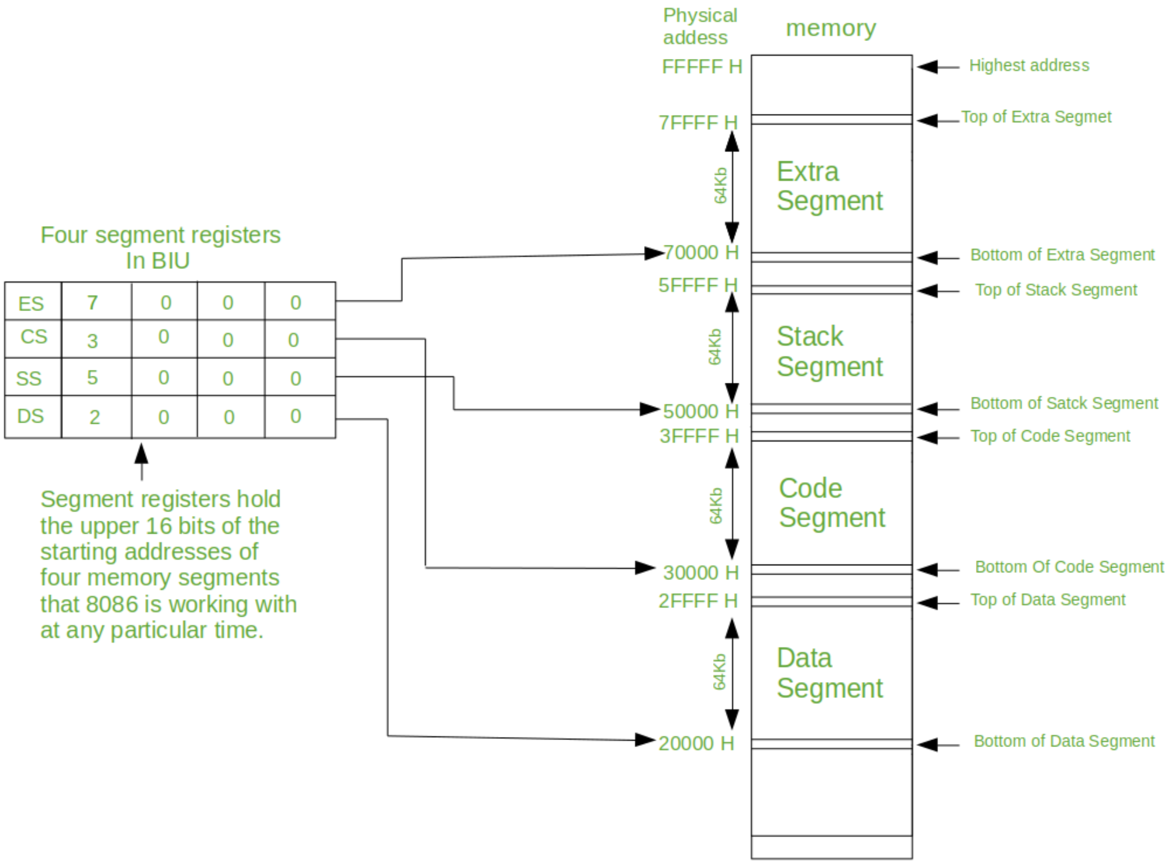
```
CS = 4321H IP = 1000H
then CS x 10H = 43210H + offset =  44210H
Here Offset = Instruction Pointer(IP)
```

CS point to the start of Code segment and offset to the position of the instruction.

Segment	*Offset*	*Special Purpose*
CS	IP	Instruction address
SS	SP or BP	Stack address
DS	BX, DI, SI, an 8-bit number or a 16-bit number	Data address
ES	DI for string instructions	String destination address

This Section is 2-hour content without any practice from student.

This basic knowledge student must have to be able to solve the assembly language problem and understand the code.



Extra For reading and improve your knowledge.

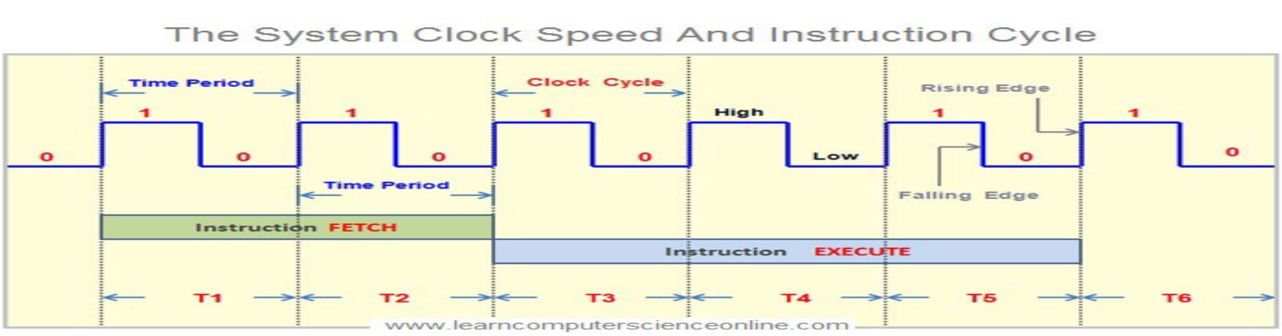
2-What is CPU Clock Cycle (CPU Speed, Frequency):

CPU Speed is measured by how many Clocks Cycle (ticks) CPU can perform in 1 second (Hz)

1 Hz it is 1 cycle per second – 1Ghz is 1 billion cycle per second -1 MHz is 1 million cycle per second

Clock Cycle: is single electrical plus in a CPU during this electrical plus CPU can perform a specific instruction (fetch, decode, execute, store) not all instructions take same number clock cycles. (Fetch can take more cycles or ticks or plus to finish and fetch is part of machine cycle.)

Some instructions require more cycles from CPU to be completed.



Prepared By: Mohamed Fathallah
Memory and Buses
Extra Knowledge on Clock cycle: CPU speed doesn't only depend on clock cycles per seconds it also depends on the architecture of your PC (cache, multiprocessing, Power, Core count, Memory paired).

Overclocking: is to raise the cup clock speed higher than the manufacturer has set it, this can cause problem since most of PC devices are synch to the CPU base clock cycle and pc is stable in this case as you increase clock speed pc might crash –this doesn't happen unless power is low.

Hint: overclocking will increase the heat of your laptop increase it A LOT and consume a lot of power.

Overheating can shorten the life of components such as processors and hard drive and it will make the computer to make it self slow to reduce heat

--2024 Updates

Clock Cycle

- **Definition**: A clock cycle is the basic unit of time in a microprocessor, determined by the clock frequency. It represents a single oscillation of the clock signal.
- **Function**: The clock signal synchronizes all operations within the microprocessor. Each cycle can trigger various operations, such as data transfer, arithmetic operations, or control signal generation.
- **Example**: If a processor has a clock speed of 2 GHz, it means the clock cycle duration is 0.5 nanoseconds (1 second / 2 billion cycles).

Machine Cycle

- **Definition**: A machine cycle is a sequence of operations that the processor performs to execute a single machine-level instruction. It typically consists of several clock cycles.
- **Components**: A machine cycle may include fetching the instruction from memory, decoding it, executing it, and storing the result. The number of clock cycles in a machine cycle can vary based on the complexity of the instruction being executed.
- **Example**: A simple instruction might take 3 clock cycles to complete, while more complex instructions could take many more.

Instruction Cycle

- **Definition**: The instruction cycle refers to the complete process of fetching, decoding, and executing an instruction. It includes all the necessary machine cycles to perform these tasks.
- **Phases**:
- **Fetch**: Retrieve the instruction from memory.
- **Decode**: Interpret the instruction to understand what actions are required.

- **Execute**: Carry out the instruction, which may involve multiple machine cycles depending on the instruction's complexity.
- **Example**: An instruction cycle may take several machine cycles, which in turn consist of multiple clock cycles.

Summary

- **Clock Cycle**: Basic timing unit (single oscillation of the clock).
- **Machine Cycle**: Sequence of operations to execute a machine-level instruction (composed of multiple clock cycles).
- **Instruction Cycle**: Complete process of fetching, decoding, and executing an instruction (composed of multiple machine cycles).