وزارة التعليم العالي والبحث العلمي
جامعة البصرة/كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم علوم الحاسوب

المعالجات المايكروية
Microprocessors

م.م. عبد الكريم حسين عبد الكريم د. حميد عبد الكريم يونس
كلية علوم الحاسوب وتكنولوجيا المعلومات/جامعة البصرة
قسم علوم الحاسوب

تشرين الاول ٢٠٢٢

Microprocessors

Syllabus:

1. History of Microprocessor.
2. Input/Output.
3. Central Processing Unit (CPU).
4. Buses
   - Address Bus.
   - Data Bus.
   - Control Bus.
5. Hardware, Software and Firmware.
6. Intel 8086 (40 pins).
7. Architecture of the 8086 Processor Model.
8. Queue.
9. Read (Word, Byte, Address).
10. Register Organization of 8086.
11. The Fetch and Execute Cycle.
12. Memory Types.
    - RAM (SRAM (Cache), DRAM).
    - ROM.
    - PROM.
    - EPROM.
    - EEPROM.
13. Generating a Memory Address.
14. The Stack (Push, Pop algorithm).
15. Addressing Modes of The 8088/8086.
    - Register Addressing Mode.
    - Immediate Addressing Mode.
    - Direct Addressing Mode.

- Register Indirect Addressing Mode.
- Based Addressing Mode.
- Index Addressing Mode.
- Based Index Addressing Mode.

Reference:

1. **Peter Abel**, ***"IBM PC Assembly Language and Programming"***, 4th Edition, Principle Hall, USA, 1998.

2. **M. Rafiquzzaman & R. Chandra,** ***"Modern Computer Architecture",*** West Publishing Company, USA, 1988.

Microprocessor

A microprocessor is an electronic device which computes on the given input similar to CPU of a computer. It is made by fabricating millions (or billions) of transistors on a single chip.

History of Microprocessor

Microprocessor journey started with a 4-bit processor called *4004*; it was made by Intel Corporation in 1971. It was 1st single chip processor. Then the idea was extended to 8-bit processors like 8008, 8080 and then 8085 (all are Intel products). 8085 was a very successful one among the 8-bit processors; however its application is very limited because of its slower computing speed and other quality factors. Some years later Intel came up with its 1st 16-bit processors 8086. Using this the first portable calculator is designed. The following Table 1 shows the list of Intel microprocessors.

Table 1

Year	Name	Bit Size (bit)
1971	4004	4
1972	8008	8
1974	8080	8
1977	8085	8
1978	8086	16
1979	8089	16
1982	80286	32
1985	80386	32
1989	80486	32
1993	80586(Pentium)	32
1995	Pentium Pro	32

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

1997	Pentium II	32
1999	Celeron and Pentium III	32
2000	Pentium IV	32
2001	Titanium	64
2003	Pentium M processor	64
2005	Pentium IV and Xeon	64
2006	Pentium D 900	64

The different manufacturing companies are introduced different bit size microprocessors in the past decade is shown in the Table 2**.**

Table 2

Company Name	**Processor Name**
AMD	Athlon
Cypress	CY7C601
DEC	ALPHA
Fujitsu	MBL8086
Harris	CS80C286
LSI Logic	LR 30000
National Semiconductor	NS321016N
SGS-Thomson	ST6X86
SUN-Micro	SRP1030
Texas Instruments	TMS390
Toshiba	TC85R4000
Zilog	Z80
Motorola	68000

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

A microcomputer system just as any other computer system, include two principal components Hardware and Software. The hardware is a course the circuitry, cabinetry … etc and the software is the collection of programs which direct the computer while it performs its tasks.

The memory is used to store both data and instructions that are currently being used. It is normally broken into several modules, each module containing several thousand locations. Each location may contain part or all of a datum or instruction and is associated with an identifier called *a memory address.* The CPU does its work by successfully inputting, or fetching instructions from memory and carrying out the tasks detected them.

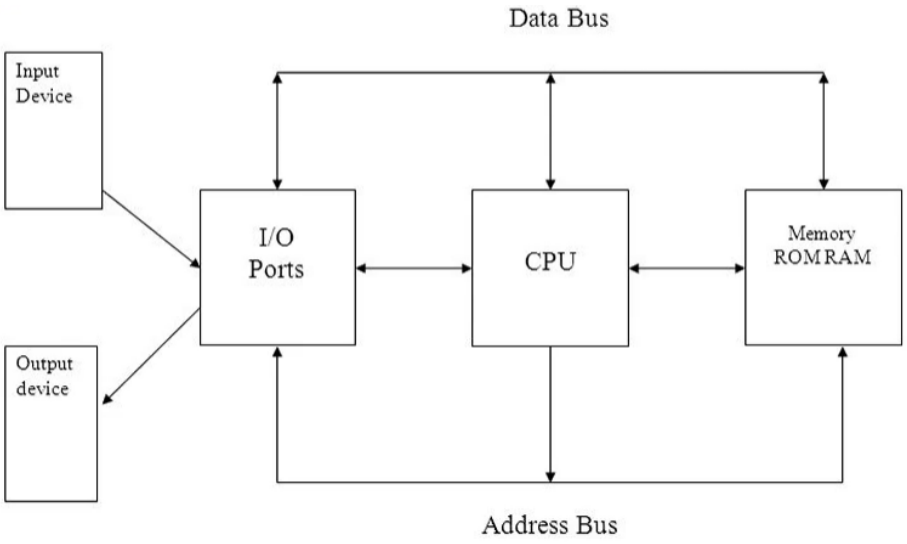


Figure 1: General architecture of a microcomputer system.

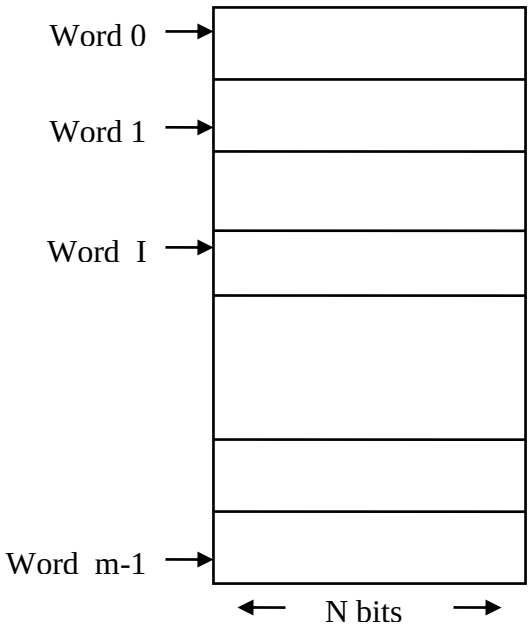
Figure 1 shows block diagram of a simple microcomputer. The major parts are the central processing unit or CPU, memory and the input and output. Connecting these parts are three sets of parallel line is called *buses and control bus*. In a microcomputer the CPU is a microprocessor and is often referred to as the microprocessor unit (MPU). Its purpose is to decode the instruction and use them to control the activity with in the system. It performs all arithmetic and logical computations.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

Memory: Memory section usually consists of a mixture of RAM and ROM. It may also magnetic floppy disks, magnetic hard disks or optical disks, to store the data.

Where (program and data) are stored. The memory chip, or simple memory contains a large number of semiconductor storage cells, each capable of storing 1 bit of information (either full or empty) it is called *a **bit***. Which is short for **binary digit** (1 or 0). A group of eight bits is a **byte**. These cells are rarely read or written as individual cells, since 1 bit represents only a very small amount of information. The usual approach is to deal them in group of fixed size. For this purpose, the memory is organized so that a group of n bits can be stored or retrieve a single basic operation. Each group of n bits is referred to as a **word** of information, and n is called the ***word length (word size)***. To access the memory to store or retrieve a single word of information, It necessary to have **unique address** (A number identify the location). This type of addressing is known as **random access memory addressing.** The word random access memory describes the way memory cells are addressed. They are accessed randomly, rather than in a strictly serial manner as magnetic tape storage. This means that any memory location, regardless of its position, can be access in the same amount of time as any location. A more convenient unit of measure for memory uses **Kilobit** and **Kilobyte (KB)** means 1024 (2^{10}) bits or byte of memory. A larger unites a **Megabyte (MB)** is 1024 (2^{10}) KB. 2^{20} bytes, while Gigabyte is 1024 (2^{10}) MB, more important than a computers word size is the amount of memory (storage capacity or size). A computer with less than 16 KB of memory is limited to trivial application because it can execute only tiny programs, such machine might be able to play video game. The computers word size can be expressed in bytes as well as in bits. For example, a word size of 8 bits is also a word size of one byte, a word size of 16 bit is also a word size of two bytes. The word size also

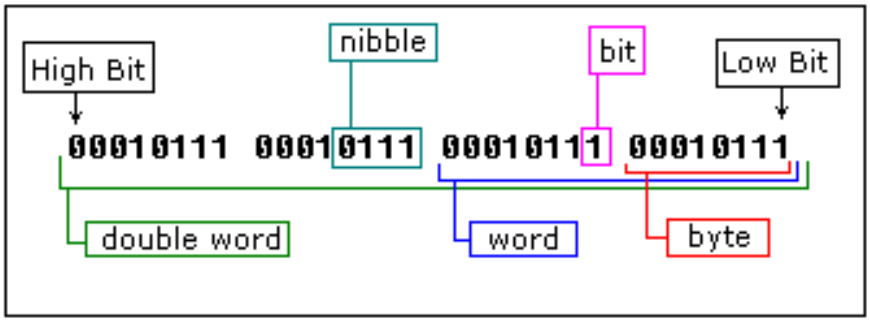
indicates the size of the data bus which carried data between the CPU and memory and between the CPU and I/O device.



The numbers from 0 to m-1 is used as the addresses of successive location in memory consisting of m words

*** Important Note**

- Each digit in a binary number is called *a **BIT***, 4 bits form a **NIBBLE**, 8 bits form a **BYTE**, two bytes form a **WORD**, two words form a **DOUBLE WORD** (rarely used):



Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

- Byte = 8 bits.

- Nibble= 4 bits.

- Word = 2 byte = 4 Nibbles = 16 bits.

- Double word = 2 Word = 4 byte = 8 Nibble = 32 bits.

- Kilo Byte (KB)=1024= 2^{10} Bytes.

- Mega Byte (MB) = 1024 KB = $2^{10} . 2^{10}$ Bytes = 2^{20} bytes =1,048,576 bytes.

- Gega Byte (GB) = 1024 MB = $2^{10} . 2^{20}$ bytes = 2^{30} byte =1,073,741,824 bytes.

- Tera Byte (TB) = 1024 GB = 20^{40} bytes.

- Memory Word Length or (memory location length) : it term use to refer to the number of bits in one memory location .

Table 3

Processor	Address Bus Size	Max Addressable Memory	In English!
8088	20	1,048,576	One Megabyte
8086	20	1,048,576	One Megabyte
80188	20	1,048,576	One Megabyte
80186	20	1,048,576	One Megabyte

80286	24	16,777,216	Sixteen Megabytes
80386sx	24	16,777,216	Sixteen Megabytes
80386dx	32	4,294,976,296	Four Gigabytes
80486	32	4,294,976,296	Four Gigabytes
80586 / Pentium (Pro)	32	4,294,976,296	Four Gigabytes

Input/output: The input/output section allows the computer to take in data from the outside world or send data to the outside world. Peripherals such as keyboards, video display terminals. Printers and modem are connected to the input/output section. These allow the user and computer to communicate with each other. The actual physical devices used to interface the computer buses to external systems are often called *ports*. An input/output port allows data from keyboard, an analog to digital converter (ADC) or some other source to be read into the computer under the control of the CPU. An output port is used to send data from the computer to some peripheral, such as, a video display terminal, a printer or a digital to analog converter (DAC).

Central processing Unit (CPU): CPU controls the operation of the computer. In a microcomputer the CPU is a microprocessor. The CPU fetches the binary coded instructions from memory, decodes the instructions into a series of simple action and carries out these actions in sequence of steps. CPU contains an a address counter or instruction pointer register which holds the address of the next instruction or data item to be fetched from memory,

general purpose register, which are used for temporary storage or binary data and circuitry, which generates the control bus signals.

Address bus: The address bus consists of 16, 20, 24 or 32 parallel lines. On these lines the CPU sends out the address of the memory locations that are to be written to or read from.

Data bus: It consists of 8, 16, 32 parallel signal lines. The data bus lines are bidirectional. This means that the CPU can read, data from memory or from a port on these lines, or it can send data out to memory or to port on these lines.

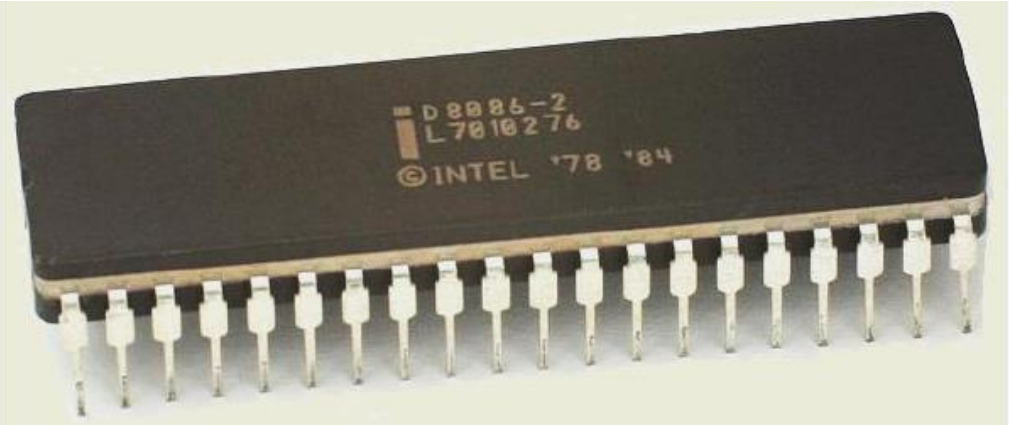
Control bus: The control bus consists of 4 to 10 parallel signals lines. The CPU sends out signals on the control bus enable the outputs of addressed memory devices or port devices. Typical control bus signal are memory read, memory write, I/O read and I/O write.

Hardware, software and Firmware: hardware is the given to the physical devices and circuitry of the computer. The job of software in a microcomputer system is to tells a computer what to do ? where to get data ? how to process the data ? and where to put the results when done ?

program is the sequence of instructions that are used to tell a computer what to do. Software refers to collection of programs written for the computer. Firmware is the term given programs stored in ROM's or in other devices which permanently keep their stored information.

Intel 8086

The 8086 is a 16-bit microprocessor chip designed by Intel corporation in between early 1976 and mid-1978. The release of Intel's 8086 microprocessor in 1978 was a watershed moment for personal computing.



Elements of the 8086 Microprocessor Architecture

The 8086 has:

- 16-bit internal data bus.
- 20-bit address bus: 2^{20} = 1,048,576 = 1 megabytes.
- Control bus.
- Execution Unit.
- Bus Interface Unit.

Among the on-chip peripherals are:

- 2 direct memory access controllers (DMA) .
- Three 16-bit programmable timers.
- Clock generator.
- Chip select unit.
- Programmable Control Registers.

The 8086 Processor Model

The simplified block diagram of the 80x86 processor model is organized as two separate processors:

1) Bus Interface Unit (BIU).

2) Execution Unit (EU).

Architecture

The internal architecture 8086 microprocessor is as shown in the Figure 2. The 8086CPU is divided into two independent functional parts, the Bus Interface Unit (BIU) and Execution Unit (EU).The Bus Interface Unit contains Bus Interface Logic, Segment registers, Memory addressing logic and a Six byte instruction object code queue. The execution unit contains the Data and Address registers, the Arithmetic and Logic Unit, the Control Unit and flags.

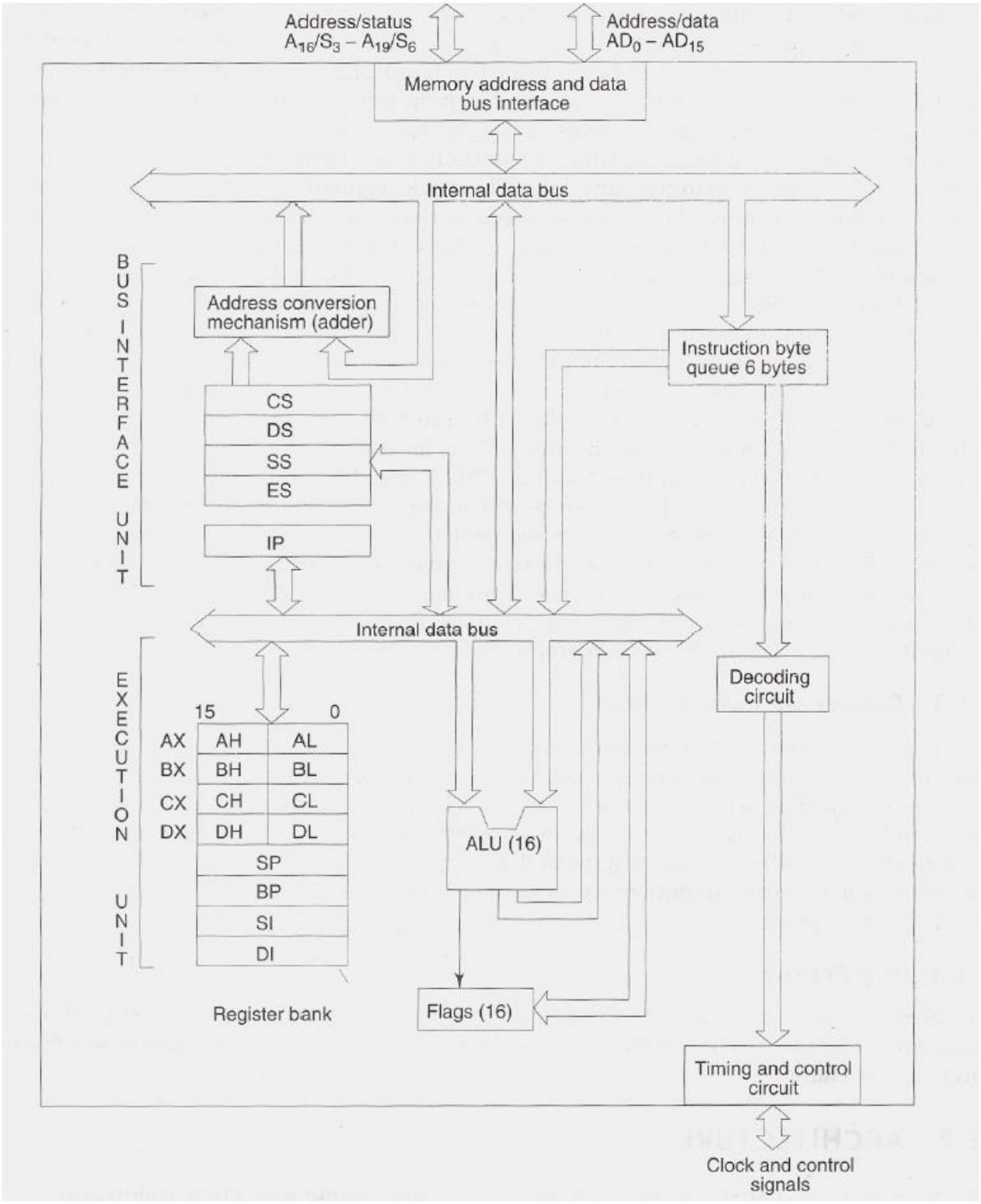


Figure 2: Internal architecture of 8086 Microprocessor.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

The BIU sends out address, fetches the instructions from memory, read data from ports and memory, and writes the data to ports and memory. In other words the BIU handles all transfers of data and addresses on the buses for the execution unit.

The execution unit (EU) of the 8086 tells the BIU where to fetch instructions or data from, decodes instructions and executes instruction. The EU contains control circuitry which directs internal operations. A decoder in the EU translates instructions fetched from memory into a series of actions which the EU carries out. The EU is has a 16-bit ALU which can add, subtract, AND, OR, XOR, increment, decrement, complement or shift binary numbers. The EU is decoding an instruction or executing an instruction which does not require use of the buses.

The Queue: The BIU fetches up to 6 instruction bytes for the following instructions. The BIU stores these pre-fetched bytes in first-in-first-out register set called *a queue*. When the EU is ready for its next instruction it simply reads the instruction byte(s) for the instruction from the queue in the BIU. This is much faster than sending out an address to the system memory and waiting for memory to send back the next instruction byte or bytes. Except in the case of JMP and CALL instructions, where the queue must be dumped and then reloaded starting from a new address, this prefetch-and-queue scheme greatly speeds up processing. Fetching the next instruction while the current instruction executes is called *pipelining*.

Word Read

Each of 1 MB memory address of 8086 represents a byte wide location.16-bit words will be stored in two consecutive memory locations.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

Register Organization of 8086

8086 has a powerful set of registers containing general purpose and special purpose registers. All the registers of 8086 are 16-bit registers. The general purpose registers, can be used either 8-bit registers or 16-bit registers. The general purpose registers are either used for holding the data, variables and intermediate result temporarily or for other purpose like counter or for storing offset address for some particular addressing modes etc. The special purpose registers are used as segment registers, pointers, index registers or as offset storage registers for particular addressing modes. Figure 3 shows register organization of 8086. We will categorize the register set into four groups as follows:

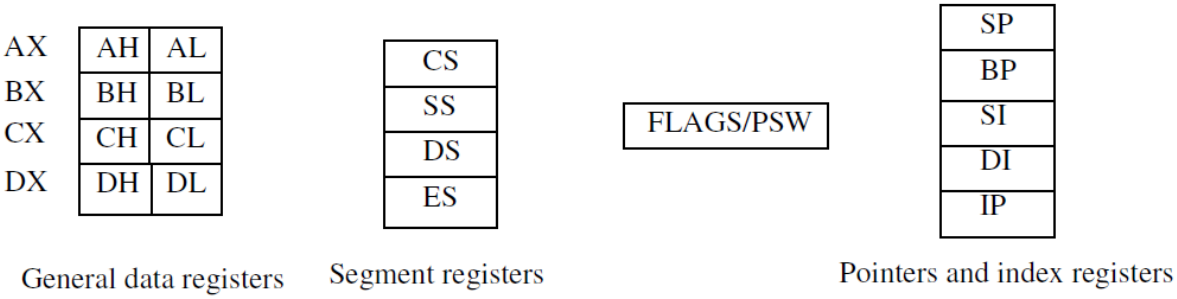


Figure 3: Register Organization of 8086.

General Purpose Registers

There are four 16-bit general purpose registers:

*AX

*BX

*CX

*DX

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

Each of these 16-bit registers are further subdivided into two 8-bit registers.

AX :	AH	AL	Accumulator
BX :	BH	BL	Base Register
CX :	CH	CL	Count Register
DX :	DH	DL	Data Register
	<--8 bits-->	<--8 bits-->	
	< 1 byte >	< 1 byte >	

AX Register: Accumulator register consists of two 8-bit registers AL and AH, which can be combined together and used as a 16- bit register AX. AL in this case contains the low-order byte of the word, and AH contains the high-order byte. Accumulator can be used for I/O operations, rotate and string manipulation.

BX Register: This register is mainly used as a **base register**. It holds the starting base

location of a memory region within a data segment. It is used as offset storage for forming physical address in case of certain addressing mode.

CX Register: It is used as default counter or **count register** in case of string and loop

instructions.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

DX Register: Data register can be used as a port number in I/O operations and implicit operand or destination in case of few instructions. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

Segment Registers

To complete 1Mbyte memory is divided into 16 logical segments. The complete 1Mbyte memory segmentation is as shown in fig 5. Each segment contains 64Kbyte of memory. There are four segment registers.

Code Segment (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during far jump, far call and far return instructions. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.

Stack Segment (SS) is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that 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. It is used for addressing stack segment of memory. The stack segment is that segment of memory, which is used to store stack data.

Data Segment (DS) is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general 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. It points to the data segment memory where the data is resided.

Extra Segment (ES) is a 16-bit register containing 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. It also refers to segment which essentially is another data segment of the memory. It also contains data.

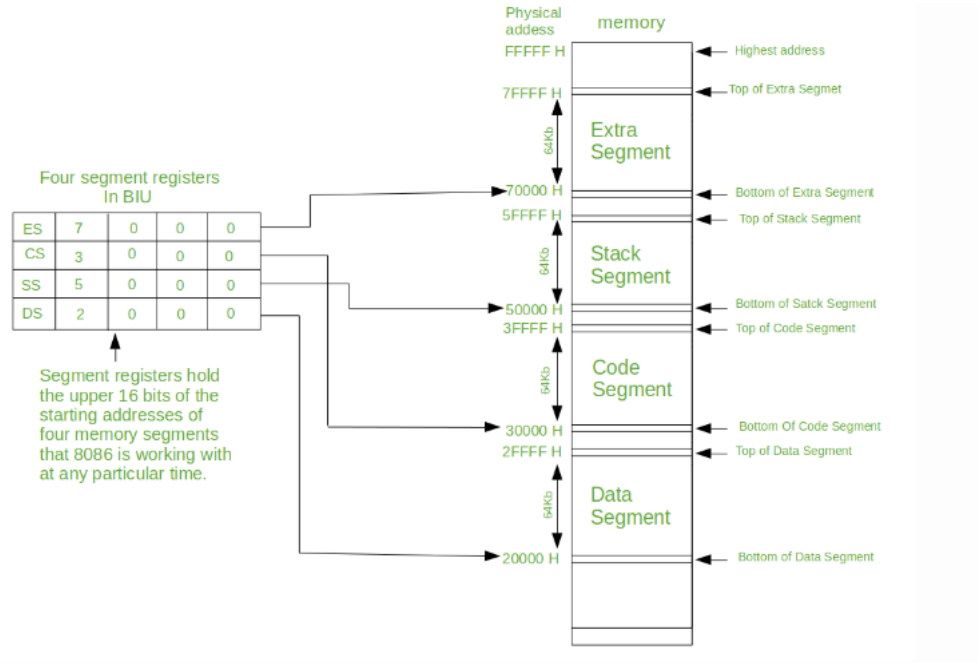


Figure 4: Memory Segmentation.

Pointers and Index Registers

The pointers contain within the particular segments. The pointers IP, BP, SP usually contain offsets within the code, data and stack segments respectively

Stack Pointer (SP) is a 16-bit register pointing to program stack in stack segment.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

Base Pointer (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.

Source Index (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.

Destination Index (DI) is a 16-bit register. DI is used for indexed, based indexed and

register indirect addressing, as well as a destination data address in string manipulation instructions.

Instruction Pointer (IP) IPR address the next instruction in a section of memory defined as instruction pointer of code segment.

Flag Register

Note: - 1 → SET , 0 → RESET

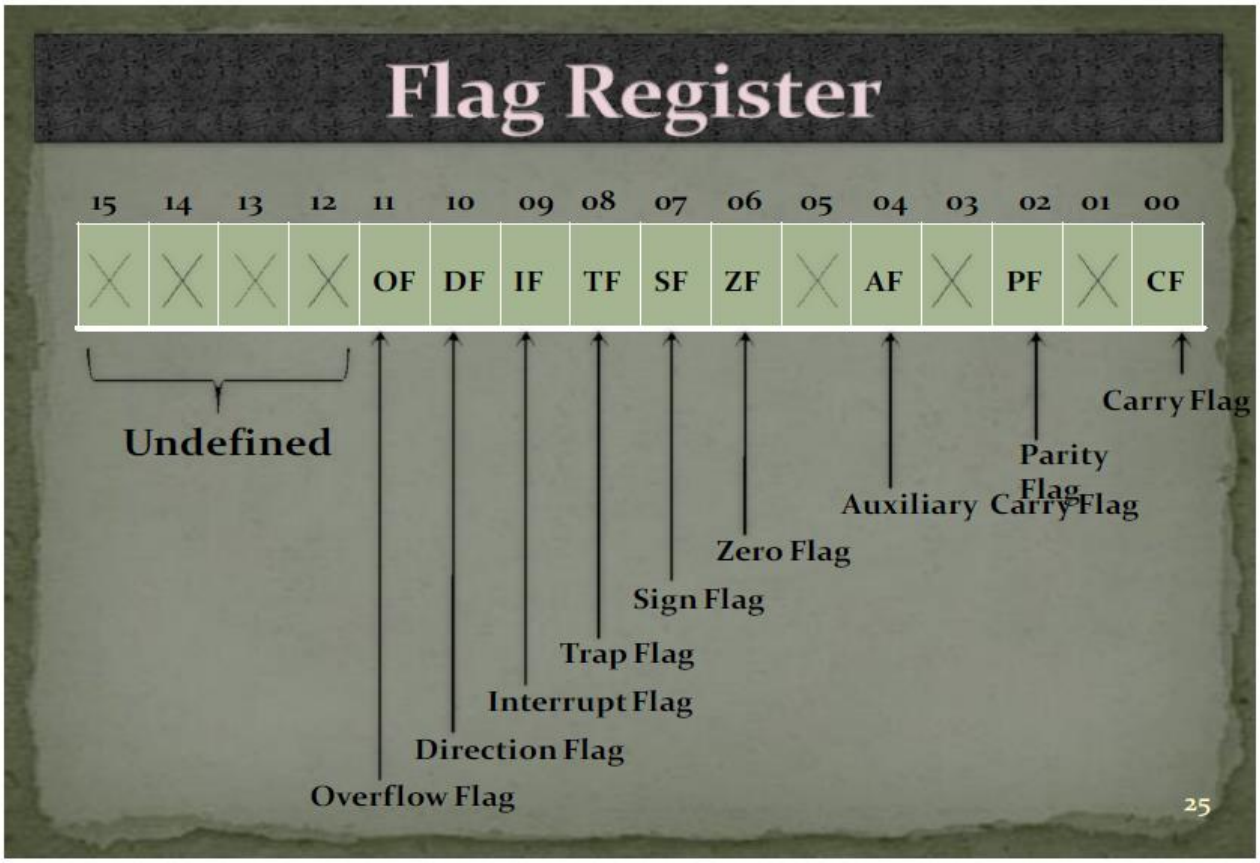


Figure 5: Flag Register of 8086.

Flags Register determines the current state of the processor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program.

The 8086 flag register as shown in the Figure 5. 8086 has 9 active flags and they are

divided into two categories:

1. Conditional Flags

2. Control Flags

Conditional Flags

Conditional flags are as follows:

Carry Flag (CF): This flag indicates an overflow condition for unsigned integer arithmetic. It is also used in multiple-precision arithmetic.

Auxiliary Flag (AF): If an operation performed in ALU generates a carry/barrow from lower nibble (i.e., D0 – D3) to upper nibble (i.e., D4 – D7), the AC flag is set i.e., carry given by D3 bit to D4 is AF flag. This is not a general-purpose flag, it is used internally by the Processor to perform Binary to BCD conversion.

Parity Flag (PF): This flag is used to indicate the parity of result. If lower order 8-bits of the result contains even number of 1's, the Parity Flag is set and for odd number of 1's, the Parity flag is reset.

Zero Flag (ZF): It is set; if the result of arithmetic or logical operation is zero else it is reset.

Sign Flag (SF): In sign magnitude format the sign of number is indicated by MSB bit. If the result of operation is negative, sign flag is set.

The Fetch and Execute Cycles

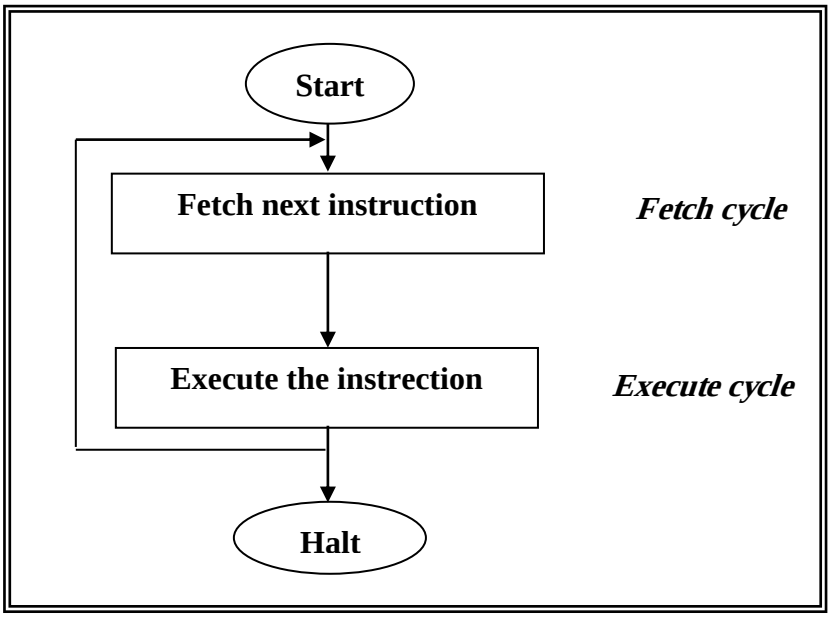


Figure 6: fetch and execute cycle.

The instruction cycle include the following steps:

1- Fetch instruction

The CPU read instruction from memory.

2- Interpret instruction

The instruction decoded to determine what action is required.

3- Fetch data

The execution of an instruction requires reading data from memory or an I/O module.

4- Process data

The execution of an instruction requires performing some arithmetic or logical operation on data.

5- Write data

The results of an instruction require writing data to memory or an I/O module.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

The following diagram state the instruction cycle:

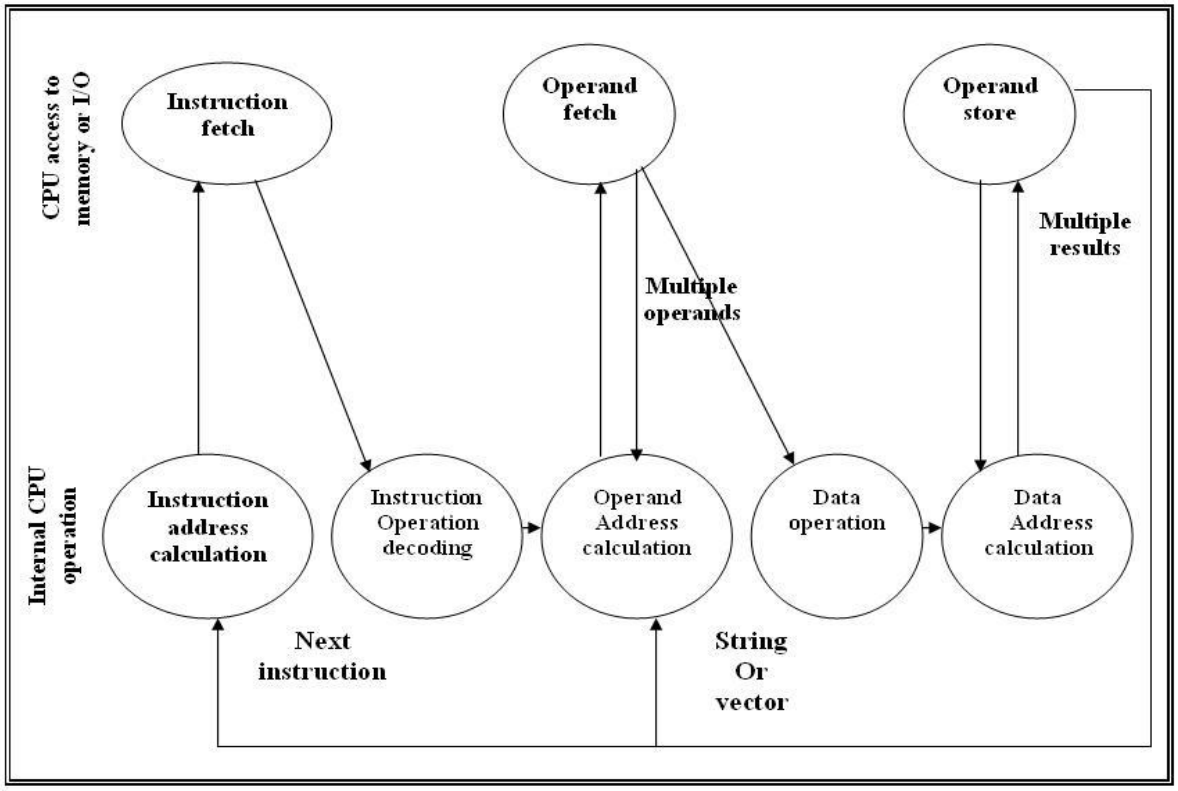


Figure 7: Instruction cycle.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

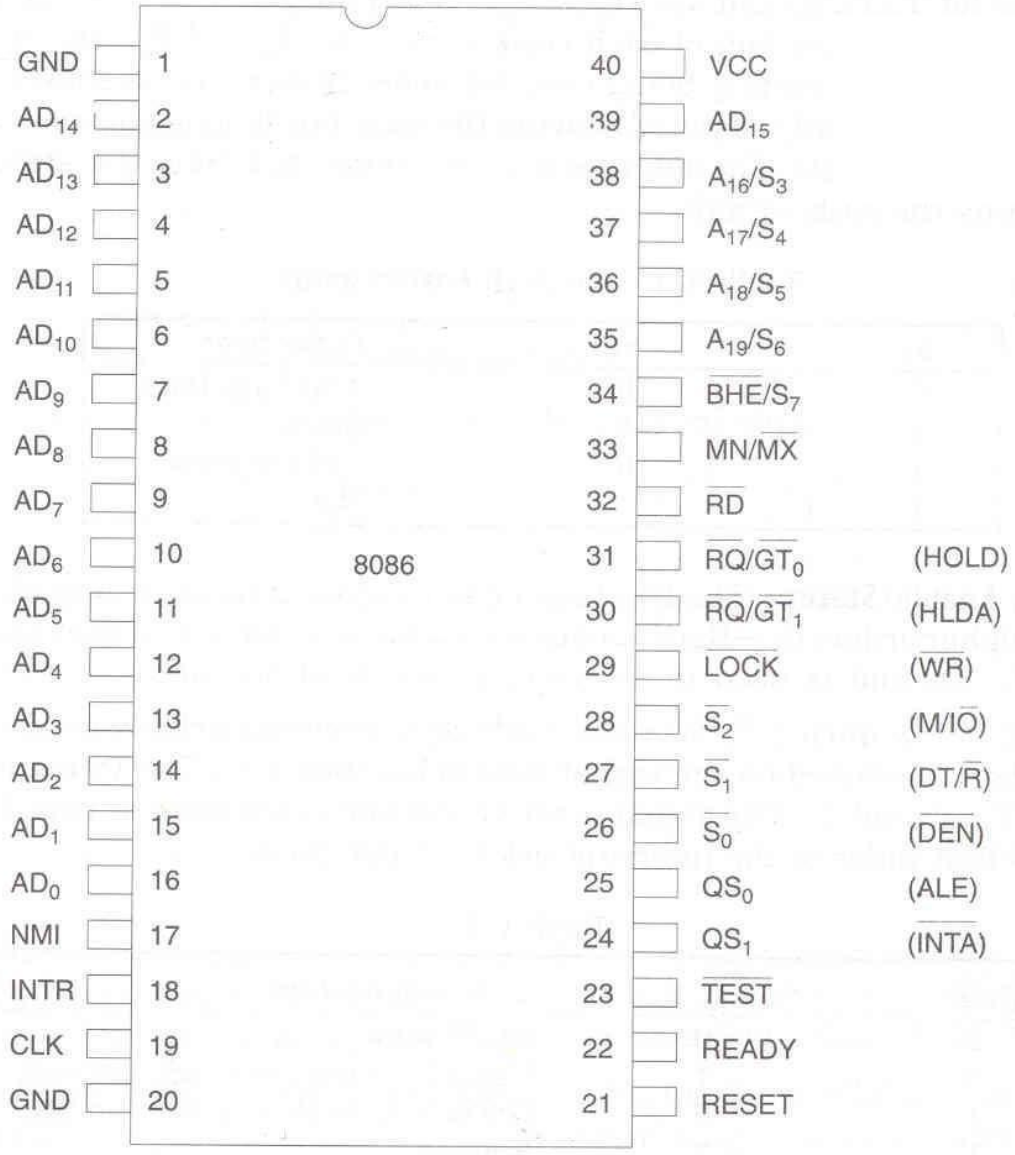


Figure 8 : Pin configuration of 8086 processor.

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

MEMORY TYPES

Random Access Memory (RAM)

The short-term memory. Data and programs are stored temporarily, while the CPU processes the data according to program's instructions, once the program has completes its task, the newly processed data can be stored in the computer's long-term-memory such as a hard disk. The RAM can be used for another processing task. RAM is what we have allows the computer to store information quickly for latter reference, so that RAM holds

RAM is a **volatile** even a short interruption in the computers power supply erase RAM. When a program instruction read the data its gets a copy of the data? This is called *a nondestructive read* because the content of the memory address are not changed. Sending data to a RAM memory address is called a *destructive write* because the new data erases whatever was there before RAM can be divided as:

- **Static RAM (SRAM)**

Semiconductor memory devices in which the stored data will remain permanently stored as long as power is supplied, without the need for periodically rewriting the data into memory. Example of SRAM cache memory.

- **Dynamic RAM (DRAM)**

Semiconductor memory devices in which the stored data will NOT remain permanently stored, even with power applied, unless the data are periodically rewriting the data into memory. The later operation is called *a refresh operation.* Example of DRAM.

Read-Only Memory (ROM)

As the name suggests can only be read, it cannot be written to under normal circumstance. The contents of ROM remain intact even after the

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

power is turned off; it is therefore known as **non-volatile** or permanent memory. ROM contains basic program which are:

- Basic I/O system (BIOS).
- The start up portion of the operating system (OS), because BIOS is the most important part in ROM, so ROM is called *BIOS ROM*.

Types of ROMs:-

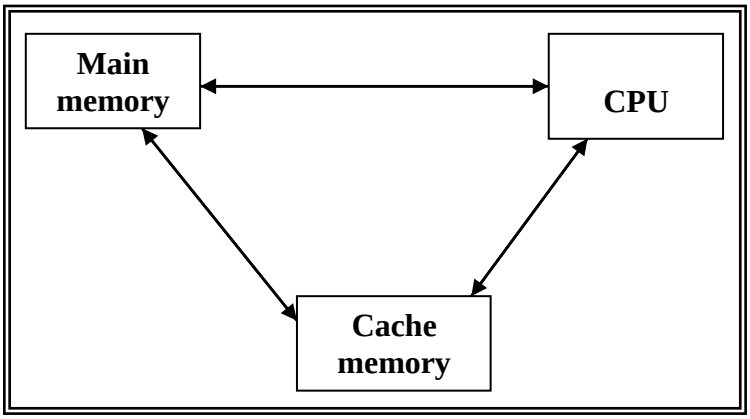
1- **Programmable–Read Only Memory (PROM)**: it's a ROM that can be electrically programmable by the user with special circuitry; it can not be erased and programmed.

2- **Erasable-Programmable-Read-Only-Memory (EPROM)**: A ROM that can be electrically programmable by the user with special circuitry; it can be erased with Ultra violet light and programmed as often as required. EPROM is considered non volatile but can lose their content with age.

3- **EEPROM (Electrically-Erasable-Programmable-Read-Only-Memory)**: A ROM that can be electrically programmed; it can be erased with electric charges and programmed as often as required.

Cache Memory

Is a special high-memory area that the CPU can access quickly? Cache can be located on the microprocessor chip or else where on the mother board. The cache memory is the only block of memory that communicates directly with the CPU at the high speed. The most frequently used instruction are kept in cache so the CPU can look there first; this allow the CPU to run faster because it doesn't have to take time to swap instruction in and out of memory.



Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

GENERATING A MEMORY ADDRESS

Physical Address Formation (Calculation)

The 8086 addresses a segmented memory. The complete physical address which is 20-bits long is generated using segment and offset registers each of the size 16-bit. The content of a segment register also called as *segment address*, and content of an offset register also called as *offset address*. To get total physical address, put the lower nibble 0H to segment address and add offset address. The Figure 9 shows formation of 20-bit physical address.

P.A. = segment register contents shifted to left with 4 zeros + offset

or

P.A. = segment register × 10h + offset

E.x.:

- Instruction fetch P.A. = CS×10h + IP
- Writing data in memory P.A. = DS×10h + DI
- Read data from memory P.A. = DS×10h + SI

Suppose:

IP register contains (xxxx xxxx xxxx xxxx)

CS register contains (yyyy yyyy yyyy yyyy)

The physical address =

0000 xxxx xxxx xxxx

yyyy yyyy yyyy 0000

---

yyyy ssss ssss xxxx

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

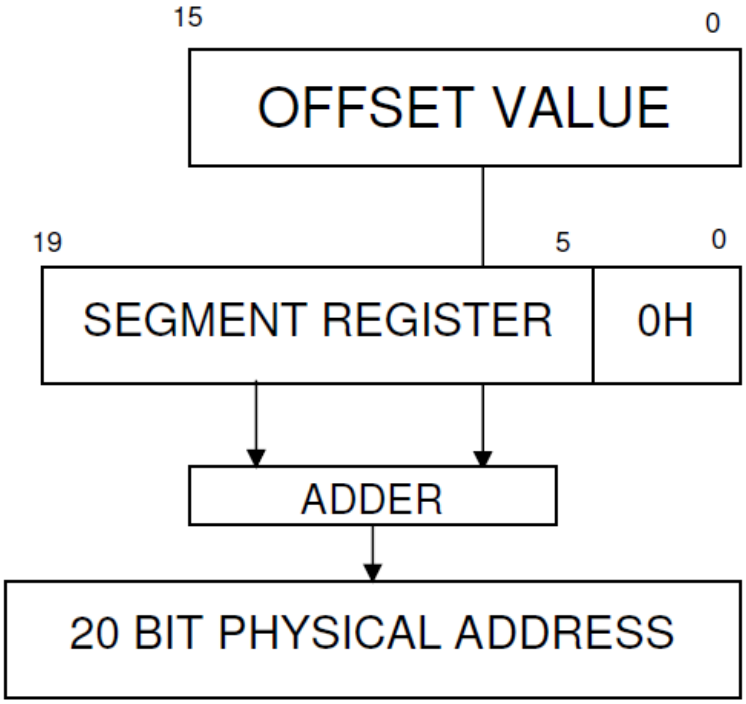


Figure 9: Physical address formation (calculation).

Logical Addresses: In the 8088 / 8086 CPU, the address of any memory location can be written in the form (**segment : offset**), so the address divided into two parts :

- **Segment address:** which held in the segment registers and have 16- bit length.
- **Offset:** which represent the distance the location which we want to reach from the beginning of the segment. The length of the offset 16 – bit also. And the value of the offset store in one of the pointer register, therefore, it will be clear that the length of one segment not more than 64KB.

Segment ⟶ (CS , DS , SS , ES)

Offset ⟶ (IP , SI , DI , SP ,BP)

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

Physical Address (P.A.): used to access memory are 20 bits in length is generated by combining 16 bit offset and 16 bit base value that is to located in one segment registers.

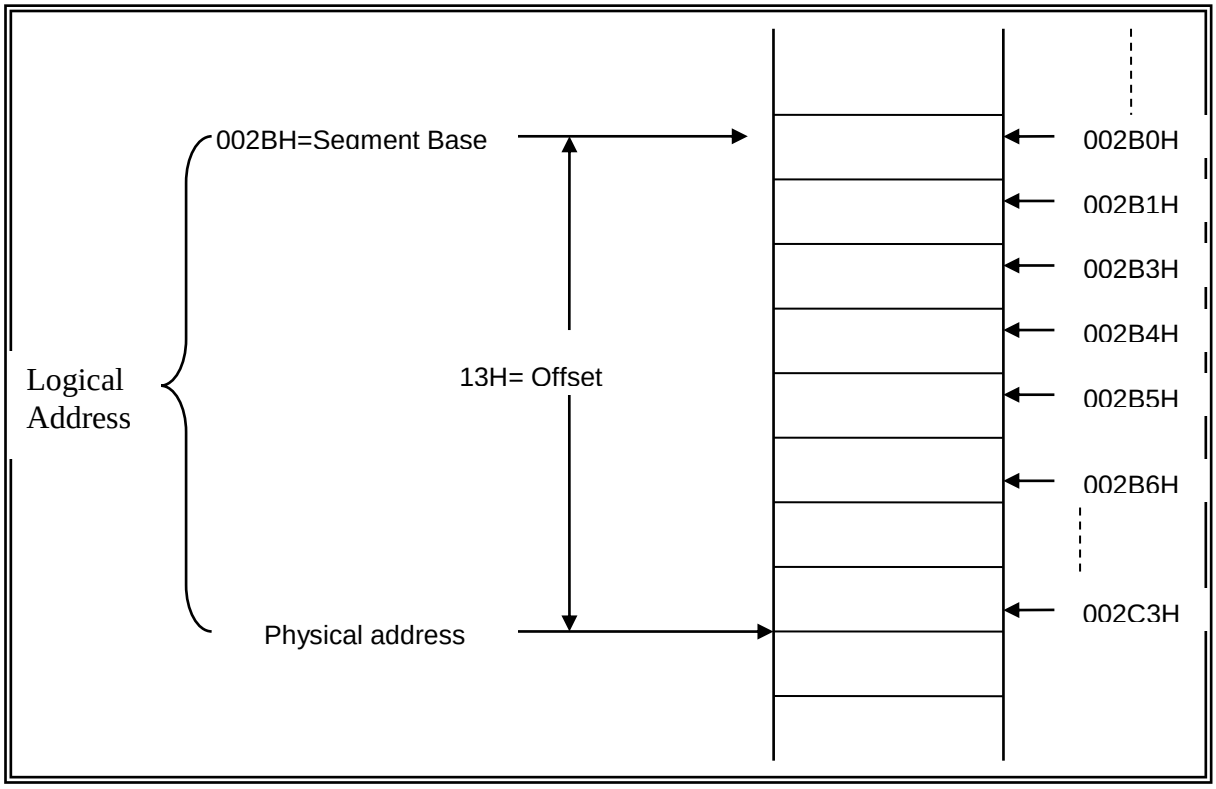


Figure 10: The Logical and Physical address.

Ex.1: find the physical address if you have offset = 50_h and segment address=3572_h.

The Physical address = segment address × 10h + offset

= 3572h × 10h + 50h

= 35720h + 50h

= 35770h (20 bits)

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

Another solution:

Offset	0000	0000	0000	0101	0000
CS	0011	0101	0111	0010	0000
PA	0011	0101	0111	0111	0000
	(3	5	7	7	0) h

Ex.2: Find P.A. if segment address = 1234_{16} and offset address = 0022_{16}?

P.A. = 1234×10h + 0022

= 12362_{16} (address in memory)

Ex.3: If CS =0200, IP = 1234, SP = 0022 find P.A .?

P.A. = 0200×10h + 1234 = 03234_{16}

Ex.4: If DS = 3000, SI = 2001, find P.A.?

P.A. = 3000×10h + 2001 = 32001_{16}

Ex.5: If SS = 2005, SP = 0201 find P.A.?

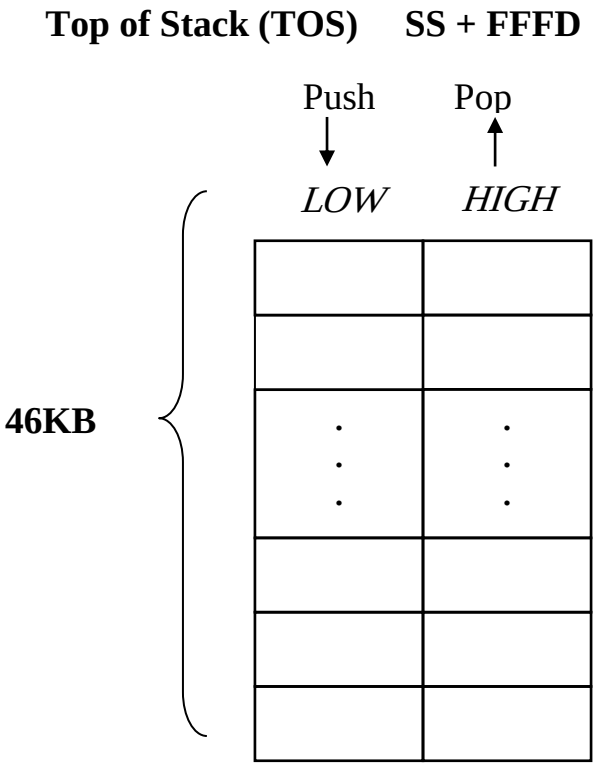
P.A. = 2005×10h + 0201 = 20251_{16}

THE STACK

It is 64 KB of memory organized as word, used to store important register's contents, that return address while call instruction address is executed.

The principle of the Stack operation is **L**east-**I**n-**F**irst-**O**ut (**LIFO**)

i.e., the last stored value is the first one returned from the stack. The stack operated by limited number of instruction such as: PUSH, POP, CALL, and RET.



P.A. stack = SS + SP

PUSH Algorithm

1. Sp ← Sp - 2
2. [Sp] ← register (the content of source register is stored in the stack)

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

```
PUSH  AX
```

CALL Algorithm

1. Sp ← Sp – 2
2. [Sp] ← IP
3. IP ← the address of subroutine.

```
CALL  fact  ; ….. fact is subroutine
```

POP Algorithm

1. register ←[Sp] (the content of stack [word] is returned from the stack into the destination register)
2. Sp ←SP + 2

```
POP  CX
```

RET Algorithm

1. IP ←[Sp]
2. Sp ←SP + 2

```
RET
```

At initializing a microprocessor, offset (SP) = $FFFF_{16}$

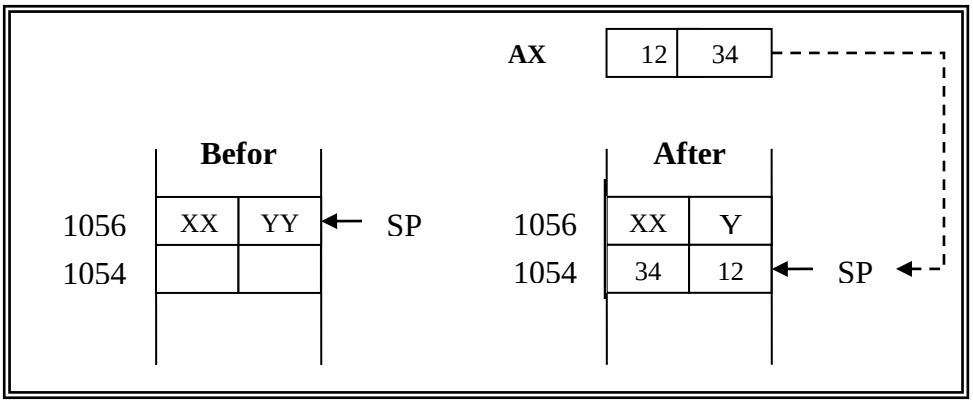
Bottom of stack (BOS) = SS + SP = SS + FFFF

Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

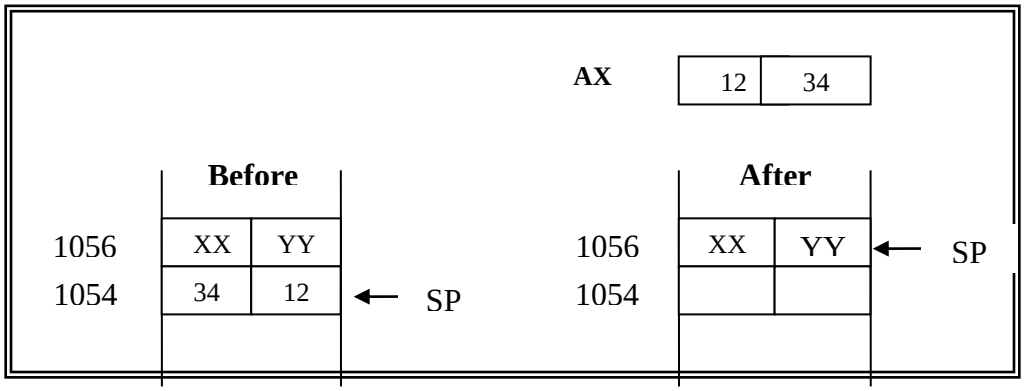
* Any number of stack may exist in an 8088 microcomputer. A new stack can be brought changing the value in the SS register.
* Only one stack is active at a time.
* The 8088 pushes data and address to the stack one word at a time.

Ex:

PUSH AX ; where AX=1234 , SS= 0105 , SP= 0006



POP AX ; where Ax=1234 , SS= 0105 , SP= 0004



Abdulkareem Hussien Abdulkareem & Dr. Hameed Abdulkareem Younis
Basrah Univ., Computer Science and Information Technology College, Computer Scie. Dept.

Addressing Modes

The manner in which a processor determines the addresses of the operands involved in that instruction is called *addressing mode*.

- There are many modes for addressing the operands, these modes will discuss briefly:-

1. Implied Mode: In this mode, the operands are specified implicitly in the definition of the instruction.

- Zero-address instructions in a stack-organization are implied mode instructions, since the operands are implied to be on top of the stack.

2. Immediate Mode: In this mode, an instruction contains the operand value.

- This mode has an operand field rather than an address field .

Example: ADD R_1, 25;$R_1 \longleftarrow R_1 + 25$

3. Register Mode: In this mode, the operand values are held in the CPU registers.

Example: ADD R_2, R_1; $R_2 \longleftarrow R_1 + R_2$

4. Register Indirect Mode: In this mode, an instruction specifies the address of a CPU register that holds the address of operand.

Example: MUL R_2, [R_1]; $R_2 \longleftarrow R_2$*[R_1]

- The selected register contains the address of the operand rather than the operand itself.
- The advantage of this mode is that the address field of the instruction uses fewer bits to select a registers than would have been required to specify a memory address directly.
- **The Effective Address (EA) or (FA):** Is the memory address obtained from the computation of the given addressing mode.

5. Direct Addressing Mode: In this mode, the effective address (PA) is equal to the address part of the instruction. The operand resides in memory and its address is given directly by the address field of the instruction.

Example: SUB R_1, X

6. Indirect Addressing Mode: In this mode, the address field of the instruction gives the address where the effective address (PA) is stored in memory.

Example: ADD R_1, [X]

Figure (1) shows some of addressing modes.

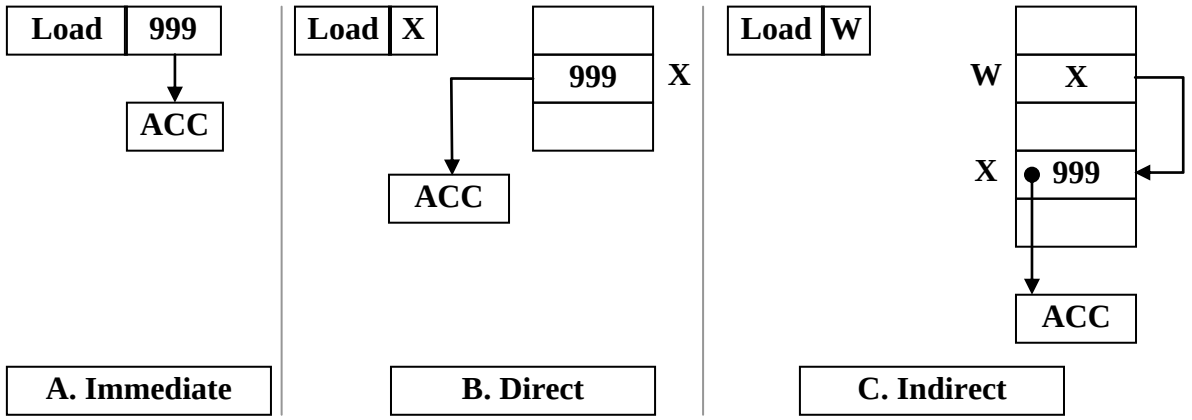


Figure (11): Some of addressing modes.

EA (or PA) = R+D

Depending on the register (R) there are three relative addressing modes:-

7. Relative Addressing Mode: In this mode, the content of the *program counter* (PC) is added to the address part of the instruction, in order to obtain the effective address (PA). The position of the effective address in memory is relative to the address of the next instruction.

Example: Let PC=825, and the address part of the instruction contains the number 24. To calculate the effective address:

1- Increment the PC after reading the current instruction. PC =PC + 1 = 826

2- Effective address = PC + 24 = 826 + 24 = 850

- Relative addressing is often used with branch–type instructions when the branch address is in the area surrounding the instruction word itself.

8. Indexed Addressing Mode: In this mode, the content of an *index register* (*XR or IR)* is added to the address part of the instruction to obtain the effective address (PA). The index register contains the index value. The address field of the instruction defines by the beginning address of a data array in memory.

Example: Let the indexed items X(0), X(1),… , X(K) are stored in consecutive addresses in memory. The instruction-address field D contains the address of the first item X(0). While the index register (XR) contains the index i, so the address of item X(i) is D+XR, as shown in Figure (2).

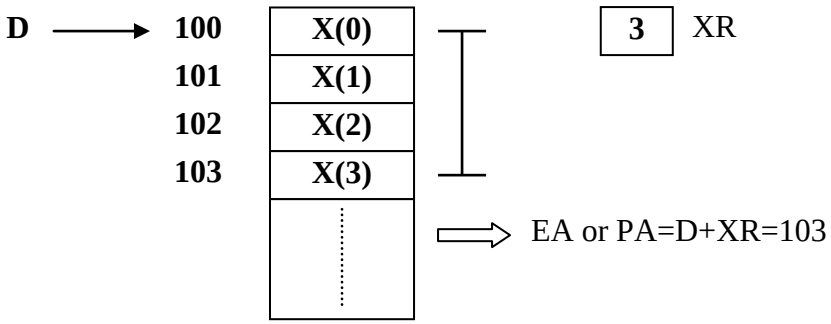


Figure (12): Use of the indexed addressing mode in accessing arrays

9. Base Register Addressing Mode: In this mode, the content of a *base register (BR)* is added to the address part of the instruction to obtain the effective address (PA). A base register held the base address, and address field gives a displacement relative to this base address.

- The base register addressing mode is used in computers to facilitate the relocation of programs in memory. When programs and data are moved from one segment of memory to another, the address values of the displacement (D) values of instructions do not have to change. Only the values of the base register requires updating to reflect the new memory segment.

Numerical Example

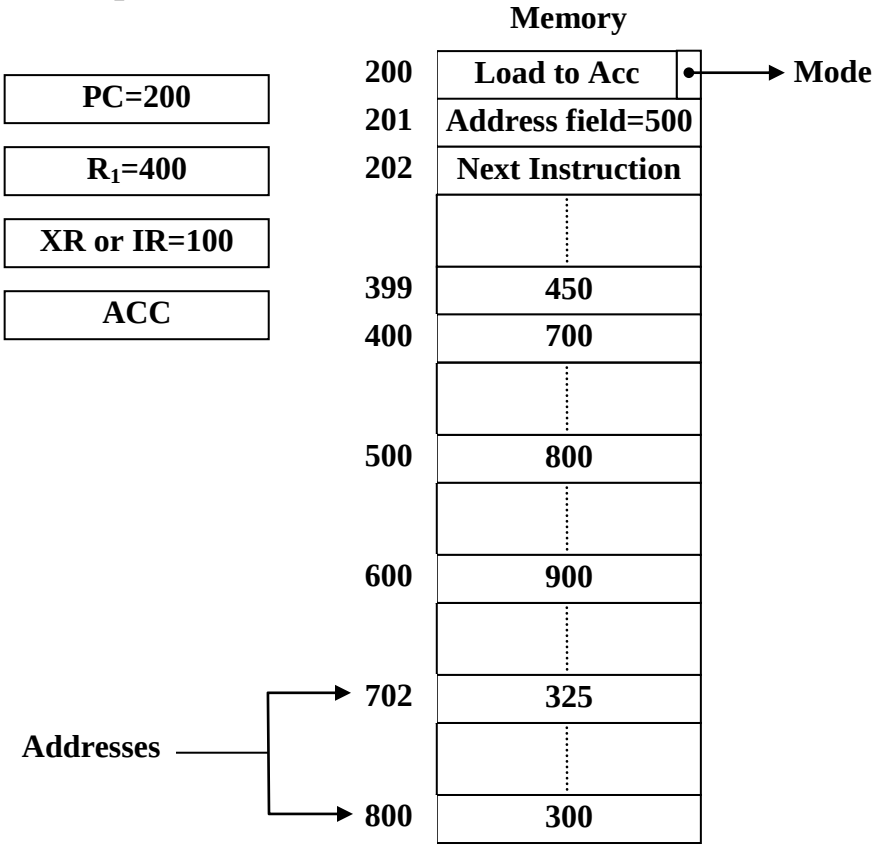


Figure (13): Numerical example.

In this example, we show the effective of the addressing (PA) mode on the above instruction, where the two-word instruction at address 200 and 201 is "Load to Acc" with on address field=500.

For each possible mode, we calculate the effective address (PA) and the operand that must be loaded in to Acc.

1. Immediate Mode:

.Effective address =201.

.Operand=500.

2. Direct Address Mode:

.Effective address =500.

.Operand=800.

3. Indirect Address Mode:

.Effective address is stored in memory at address 500, So

.Effective address =800.

.Operand=300.

4. Relative Mode:

.Effective address =(PC+2)+500=702.

.Operand=325.

5. Index Mode:

.Effective address =XR+500=100+500=600.

.Operand=900.

6. Register Mode:

.Operand is in R1 and 400 is loaded to Acc.

7. Register Indirect:

.Effective address =400.

.Operand=700.