Lab2 CPU&Memory

Computer Organization & Architecture

Nedaa Hussein Ahmed

nh1179@fayoum.edu.eg

Characters

- In addition to numbers, computers must be able to handle non numeric text information consisting of characters.
- Characters can be letters of the alphabet, decimal digits, punctuation marks, and so on. They are represented by codes that are usually eight bits long. One of the most widely used such codes is the (ASCII) code.

Characters cont...

- ASCII stands for American Standard Code for Information Interchange.
- Computers can only understand numbers, so an ASCII code is the numerical representation of a character such as 'a' or '@' or an action of some sort.

ASCII Code

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	00	NUL	32	20	SP	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(	72	48	H	104	68	h
9	09	HT	41	29	)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Notes

- The ASCII code for the small characters is = the ASCII code for the Capital characters + 20h.

ex:

The ASCII code for ”a”= The ASCII code for “A”+20h
= 41h+20h =61h

- The ASCII code for the decimal number=the decimal number +30h

ex:

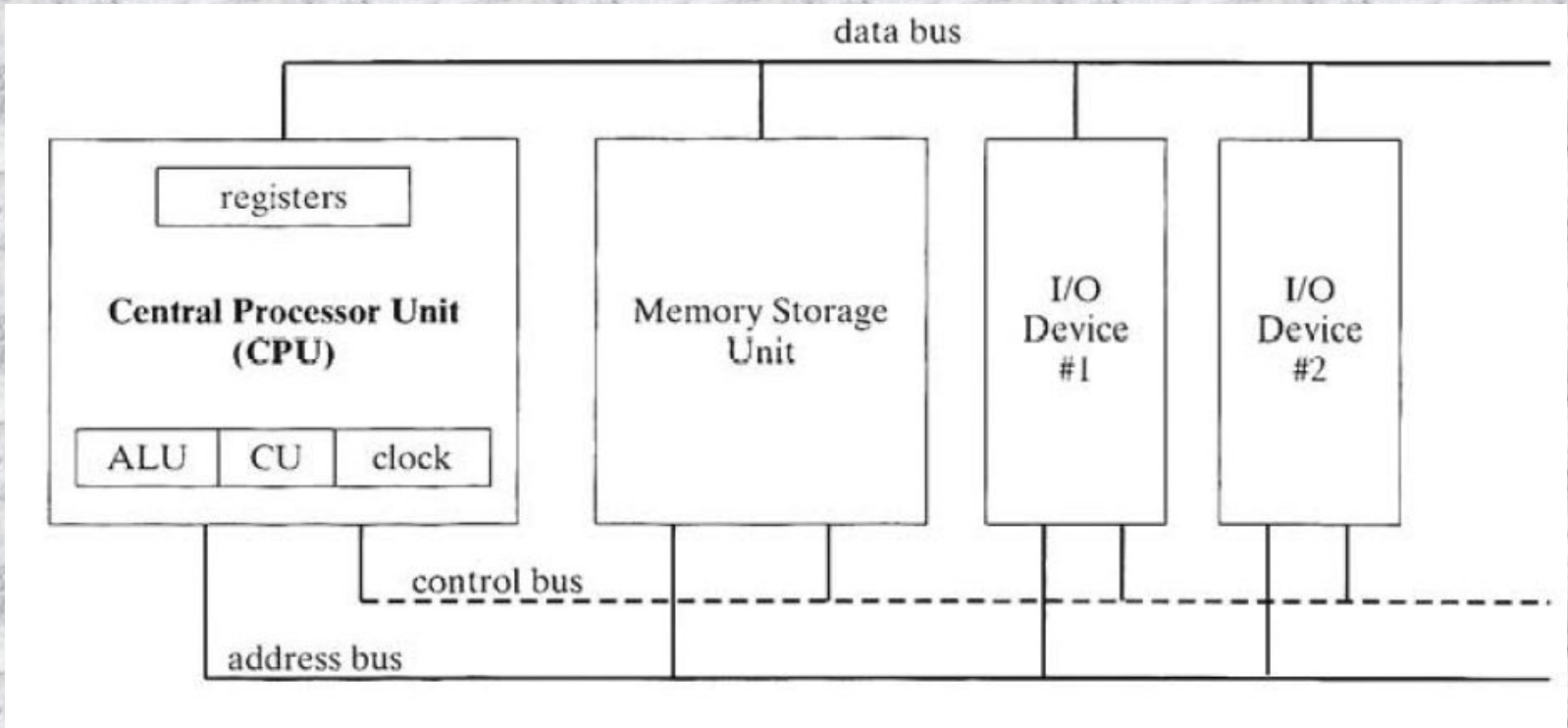
The ASCII code for number 7=7+30h=37h

Microprocessor

- The computer you are using to read this presentation uses a microprocessor to do its work.
- The microprocessor is the heart of any normal computer.
- A microprocessor -- also known as a CPU [central processing unit] is a complete computation engine that is fabricated on a single chip.

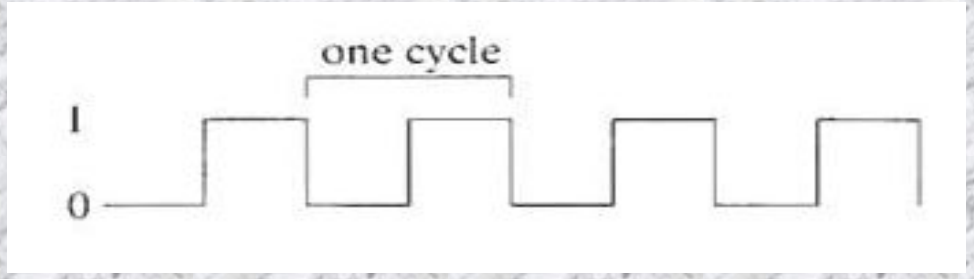
Basic Microcomputer Design

- The central processor unit (CPU) is where all the calculations and logic operations take place.
- It contains a limited number of storage locations called registers, a high-frequency clock, a control unit, and an arithmetic logic unit .



Basic Microcomputer Design cont...

- The clock: synchronizes the internal operations of the CPU with other system components.



- A machine instruction requires at least one clock cycle to execute, and a few require in excess of 50 clocks (ex. multiply).
- The control unit (CU): coordinates the sequencing of steps involved in executing machine instructions.
- The arithmetic logic unit (ALU): performs arithmetic operations such as addition and subtraction, and logical operations such as AND, OR, and NOT.

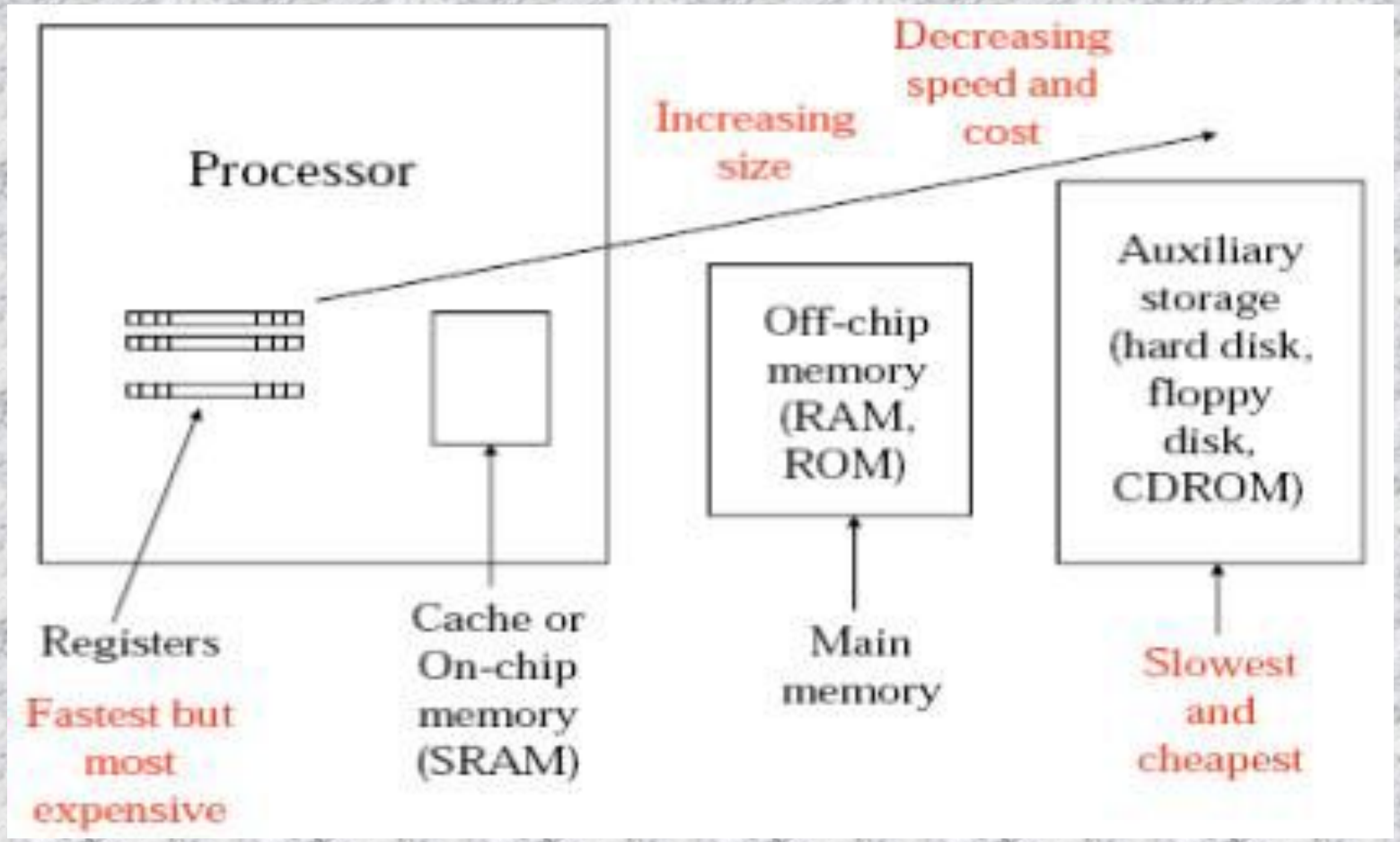
Basic Microcomputer Design cont...

- The memory storage unit : where instructions and data are held while a computer program is running. The storage unit receives requests for data from the CPU, transfers data from random access memory (RAM) to the CPU, and transfers data from the CPU into memory.

Basic Microcomputer Design cont...

- A bus is a group of parallel wires that transfer data from one part of the computer to another. The system bus of a computer usually consists of three different busses :
  1. The data bus, transfers instructions and data between the CPU and memory.
  2. The control bus, uses binary signals to synchronize the actions of all devices attached to the system bus (read or write).
  3. The address bus, holds the addresses of instructions and data when the currently executing instruction transfers data between the CPU and memory.

Memory hierarchy



Instruction Execution Cycle

- The execution of a single machine instruction can be divided into a sequence of individual operations called the instruction execution cycle.
- When the CPU executes an instruction using a memory operand, it must calculate the address of the operand, place the address on the address bus, wait for memory to get the operand and so on.
- When the CPU executes a single machine instruction, three primary operations are always necessary: fetch , decode, and execute. Two more steps are required when the instruction uses a memory operand: fetch operand, and store output operand.

Instruction Execution Cycle cont...

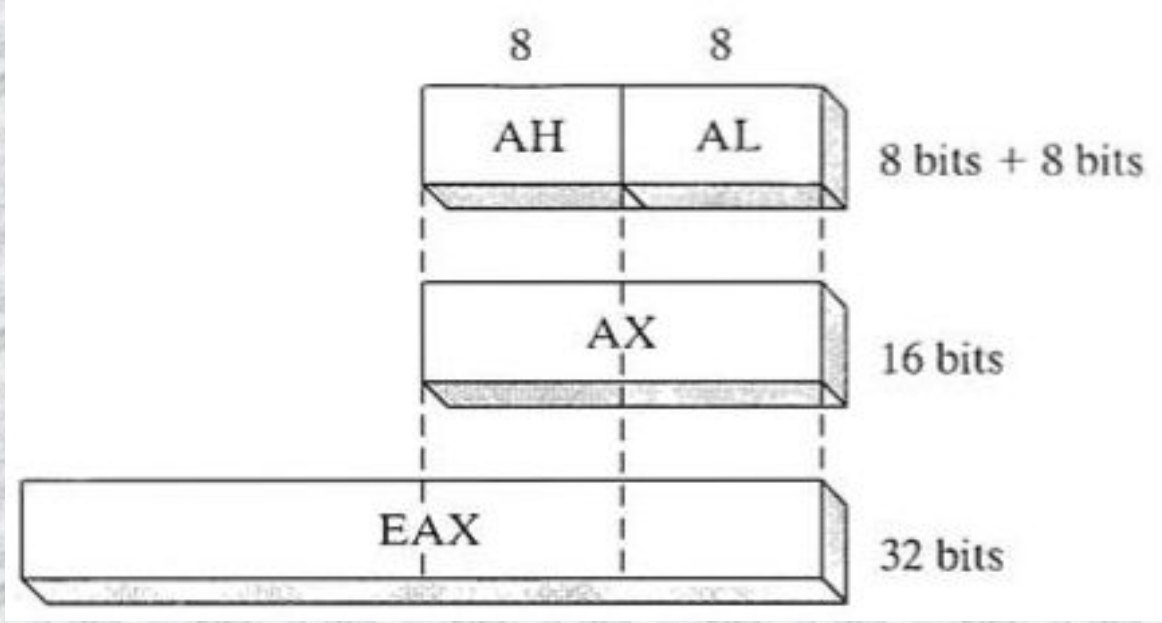
- Fetch: The control unit fetches the instruction , copying it from memory into the CPU and increments the program counter (PC).
- Decode: The control unit determines the type of instruction to be executed . It passes zero or more operands to the arithmetic logic unit (ALU) and sends signals to the ALU that indicate the type of operation to be performed.

Instruction Execution Cycle cont...

- Fetch operands: If a memory operand is used , the control unit initiates a read operation to retrieve the input operand from memory.
- Execute: The arithmetic logic unit executes the instruction, sends its data to the output operand, and updates status flags providing information about the output.
- Store output operand: If the output operand is in memory, the control unit initiates a write operation to store the data.

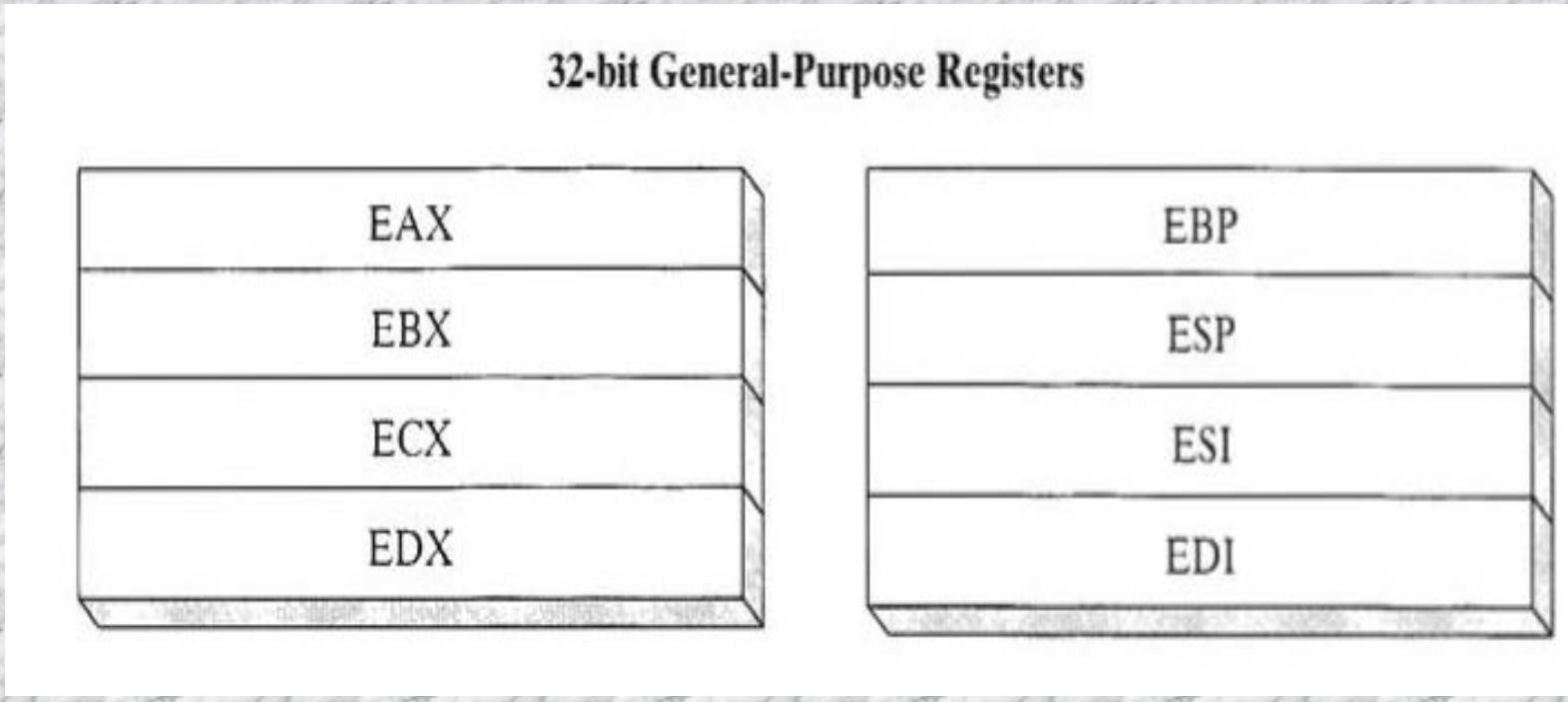
Basic Program Execution Registers

- Registers are high-speed storage locations directly inside the CPU, designed to be accessed at much higher speed than conventional memory.
- There are eight general-purpose registers, six segment registers, a register that holds processor status flags (EFLAGS) and an instruction pointer (EIP).



General-purpose registers

- General-purpose registers are primarily used for arithmetic and data movement.

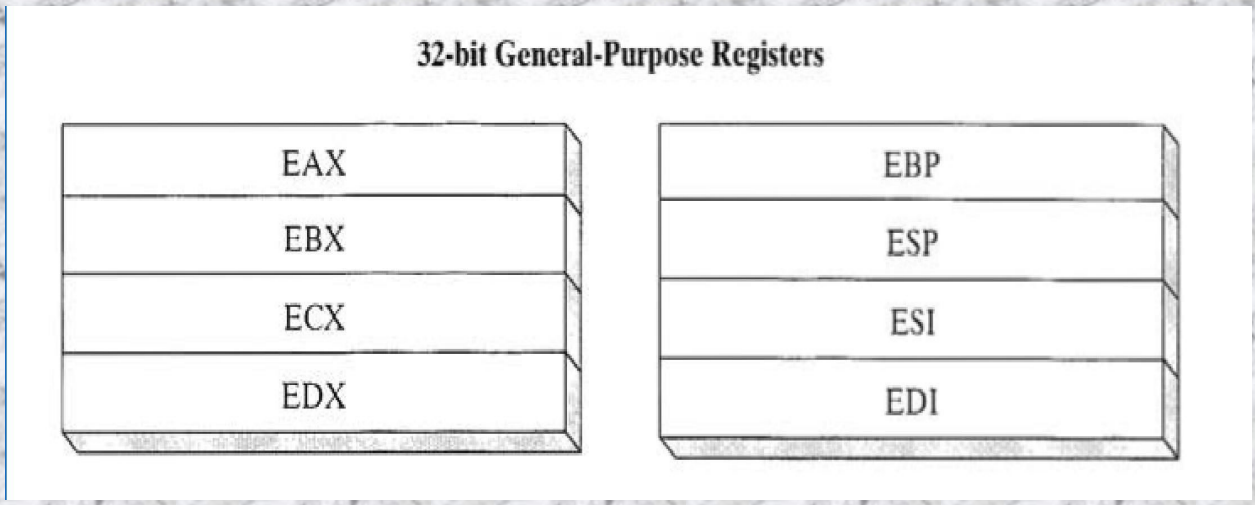


General-purpose registers cont...

- **Specialized Uses Some general-purpose registers have specialized uses:**

  ➢EAX(Extended Accumulator) is automatically used by multiplication and division instructions.

  ➢The CPU automatically uses ECX(Extended Count) as a loop counter.



General-purpose registers cont...

- **Specialized Uses Some general-purpose registers have specialized uses:**

  ➢ESP(Extended Stack Pointer) addresses data on the stack (a system memory structure). It should never be used for ordinary arithmetic or data transfer.

  ➢ESI and EDI(Extended Source Index and Extended Destination Index) are used by high-speed memory transfer instructions.

Segment Registers & IP

- The segment registers are used as base locations for pre-assigned memory areas called segments.
  - ✓CS (code segment) - points at the segment containing the current program.
  - ✓DS (Data Segment) - generally points at segment where variables 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.
- The EIP, or instruction pointer register contains the address of the next instruction to be executed

EFLAGS Register

The EFLAGS (or just Flags) register consists of individual binary bits that either control the operation of the CPU or reflect the outcome of some CPU operation.

There are machine instructions that can test and manipulate the processor flags.

A flag is set when it equals 1: it is clear (or reset) when it equals 0.

EFLAGS Register cont...

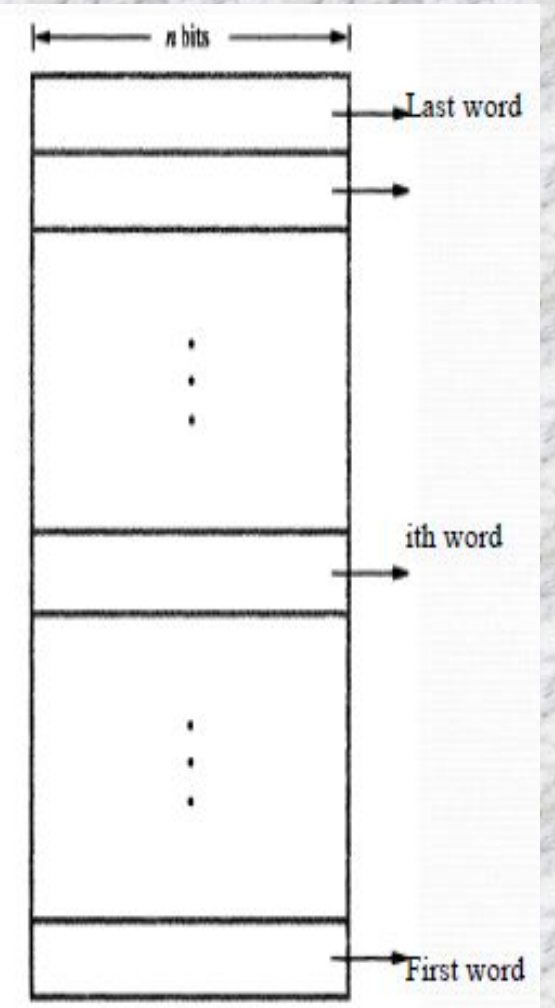
- The Carry flag (CF) is set when the result of an unsigned arithmetic operation is too large to fit into the destination .
- The Overflow flag (OF) is set when the result of a signed arithmetic operation is either too large or too small to fit into the destination.
- The Sign flag (SF) is set when the result of an arithmetic or logical operation generates a negative result.
- The Zero flag (ZF) is set when the result of an arithmetic or logical operation generates a result of zero.
- The Auxiliary Carry flag (AC) is set when an arithmetic operation causes a carry from for example bit 3 to bit 4 in an 8-bit operand .
- The Parity flag (PC) sums the number of bits that are set in a number, and indicates whether the sum is odd or even.

Memory locations and addresses

- Number and character operands, as well as instructions, are stored in the memory of a computer.

We will now consider how the memory is organized:

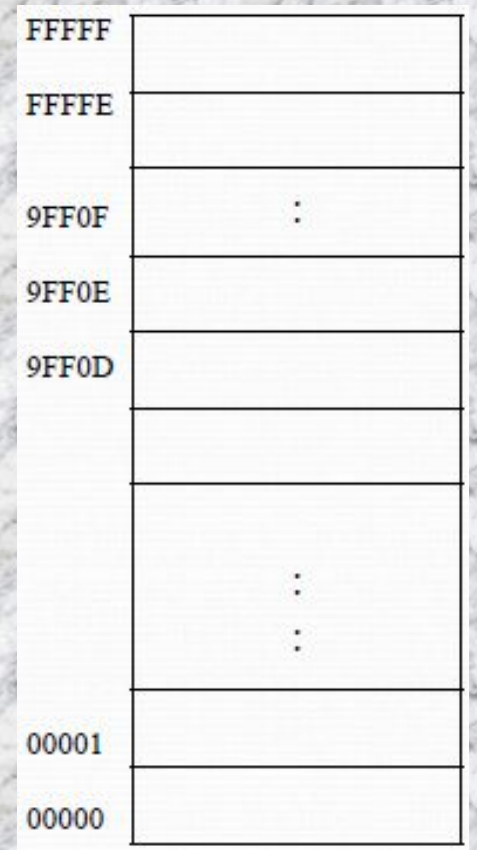
- The memory consists of many millions of storage cells, each of which can store a bit of information having the value 0 or 1.
- The memory is organized where a group of n bits can be stored or retrieved in a single basic operation.
- Each group of n bits is referred to as a word of information, and n is called the word length. The memory of a computer can be schematically represented as a collection of words.



Byte	8 bit
word	From 16 to 64 bits

Memory locations and addresses cont...

- Accessing the memory to store or retrieve a single item of information, either a word or a byte, requires distinct names or addresses for each item location.
- It is customary to use numbers from 0 through 2k - 1, for some suitable value of k, as the addresses of successive locations in the memory.
- The memory can have up to 2k addressable locations.
- For example, a 24-bit address generates an address space of (16,777,216) locations. This number is usually written as 16M (16 mega)
- A 32-bit address creates an address space of 232 or 4G (4 Giga) locations.



1 Kilo	2^{10}	1024
1 Mega	2^{20}	1,048,576
1 Giga	2^{30}	1,073,741,824
1 Tera	2^{40}	1,099,511,627,776

IA-32 Memory Management

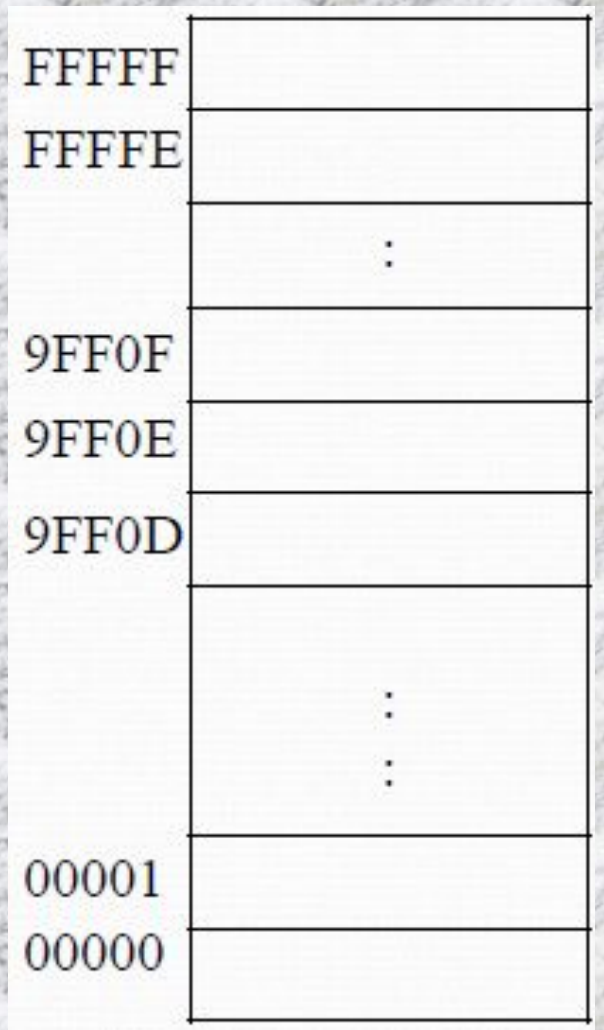
➢**IA-32 manages memory according to the basic modes of operation:**

1. Real-Address mode

➢ 1 MB (1,048,576 byte) RAM maximum addressable using 20-bit addresses from hexadecimal 00000 to FFFFF.

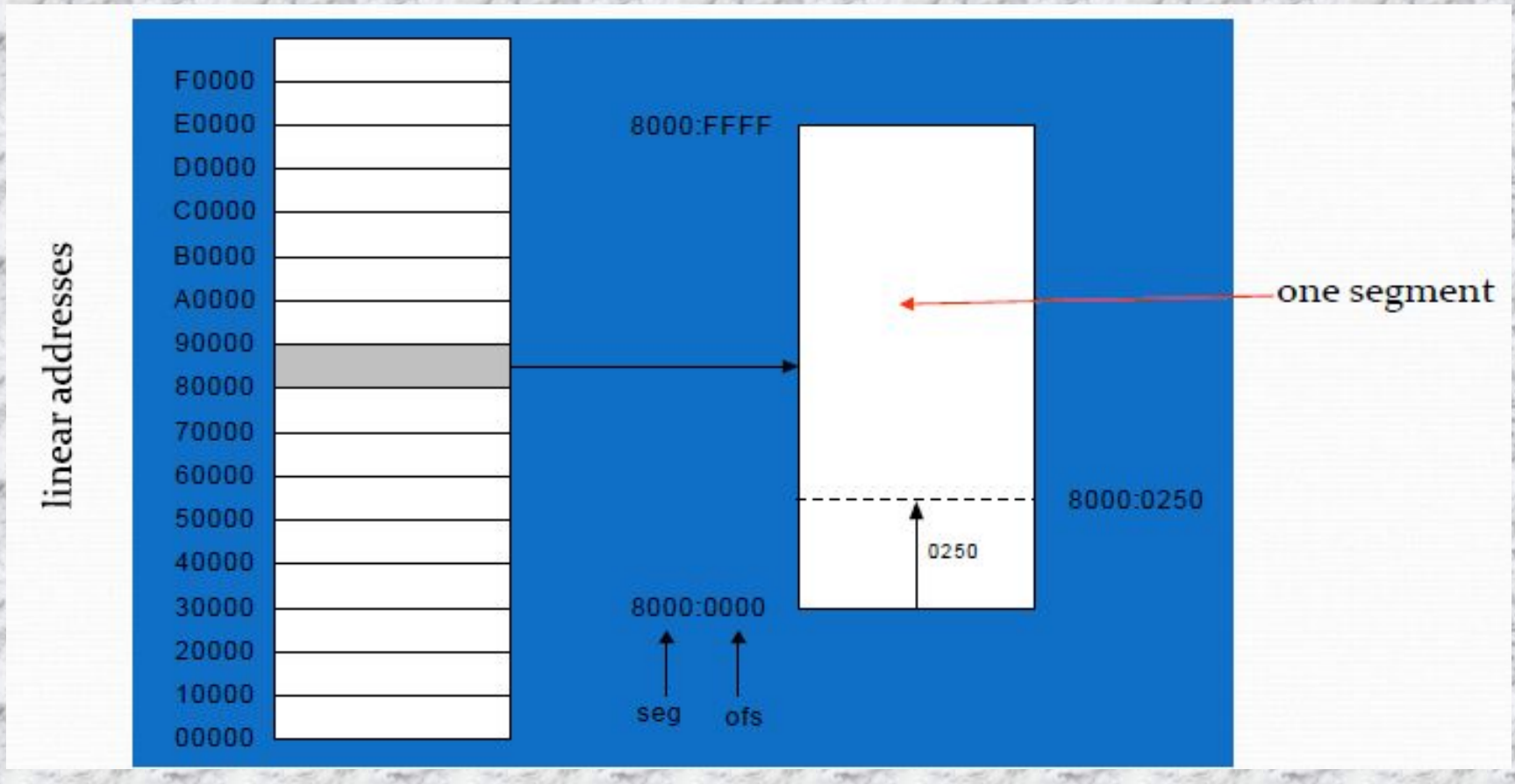
➢ But the original 8086 processor had only 16-bit registers so it was impossible to directly represent a 20 -bit address.

➢They came up with a scheme known as segmented memory.



Segmented Memory

- Segmented memory addressing: absolute (linear) address is a combination of a 16-bit segment value added to a 16-bit offset.
- All of memory is divided into 64-kilobyte units called segments.



Calculating Linear Addresses

- Given a segment address, multiply it by 10h (add a hexadecimal zero), and add it to the offset
- **Example:**
  - convert **08F1:0100** to a linear address

Adjusted Segment value:	**0 8 F 1 0**
Add the offset:	**+ 0 1 0 0**

Linear address:	**0 9 0 1 0**

Calculating Linear Addresses cont...

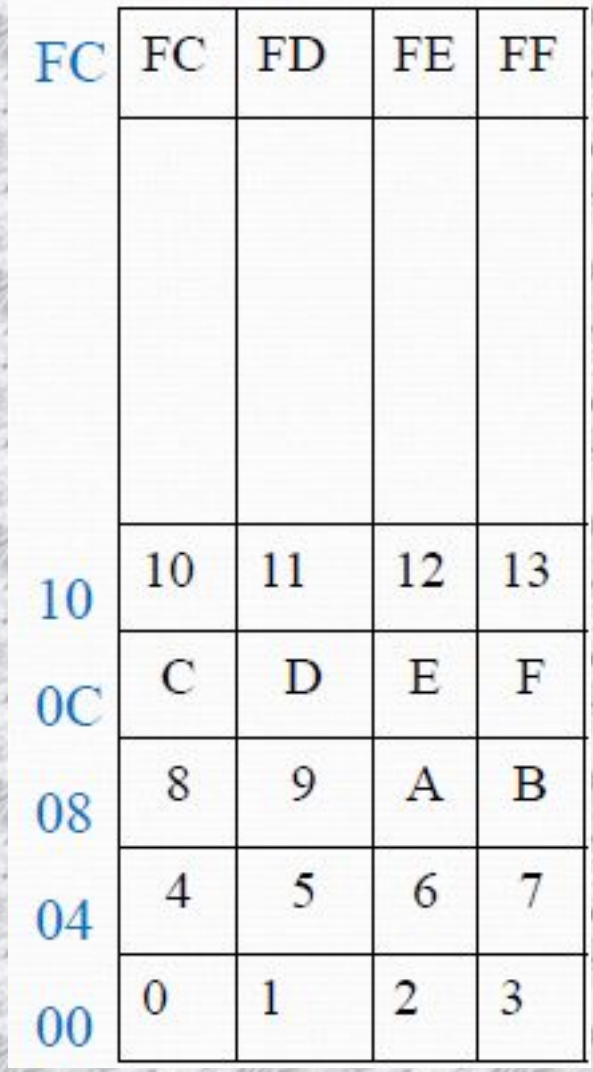
What linear address corresponds to the segment/offset address 028F:0030?

028F0 + 0030 = 02920

Always use hexadecimal notation for addresses.

Memory locations and addresses cont...

- **Byte addressability:**
  - The most practical assignment is to have successive addresses refer to successive byte locations in the memory.
  - The term byte-addressable memory is used for this assignment. Byte locations have addresses 0, 1, 2, ... Thus, if the word length of the machine is 32 bits, successive words are located at addresses 0, 4, 8, . .., with each word consisting of four bytes.

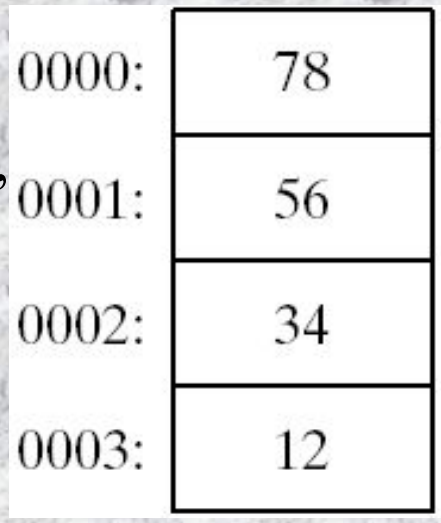


Little Endian Order

- Intel processors store and retrieve data from memory using what is referred to as little endian order. This means that the least significant byte of a variable is stored at the lowest address. The remaining bytes are stored in the next consecutive memory positions.
- Example:

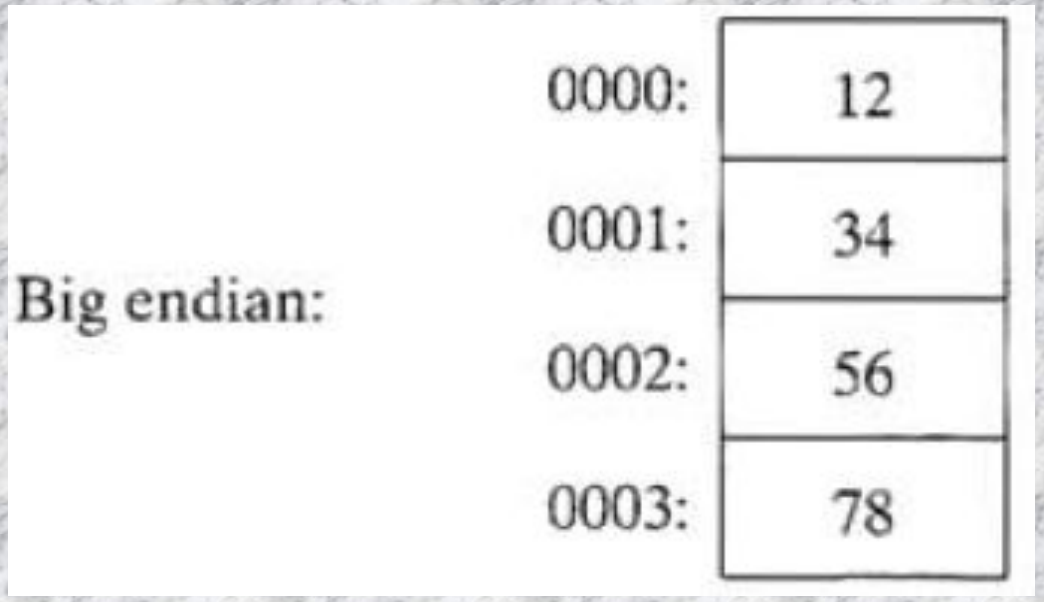
  Consider the doubleword **12345678h**. If placed in memory at offset 0

  - 78h would be stored in the first byte.
  - 56h would be stored in the second byte, and
  - the remaining bytes would be at offsets 3 and 4, as the following diagram shows:



Big Endian Order

➢Big endian order (high to low). The following figure Shows an example of **12345678h** stored in big endian order at offset 0:




Thank You