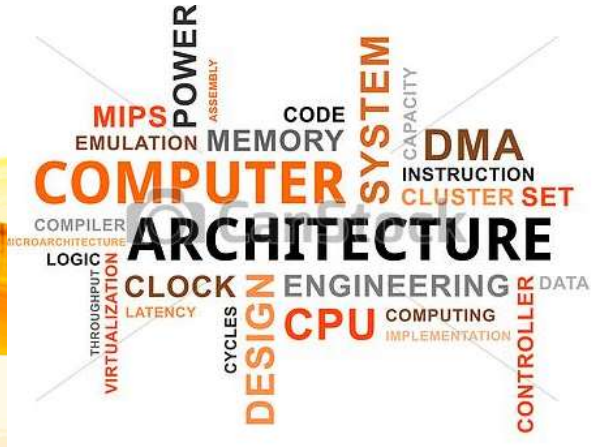
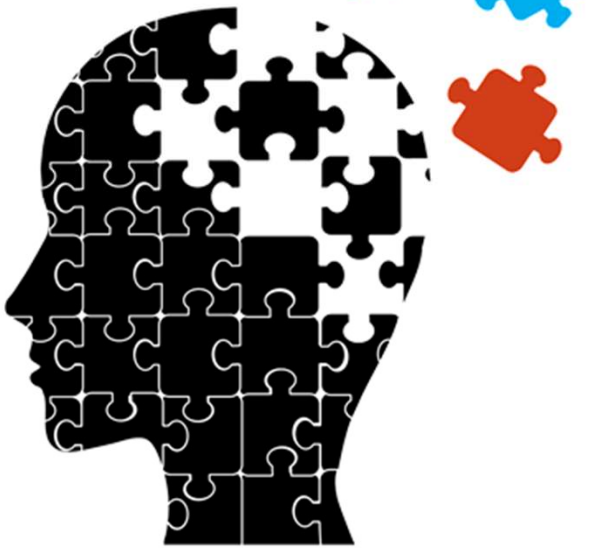
UNIT I
BASIC STRUCTURE OF COMPUTERS

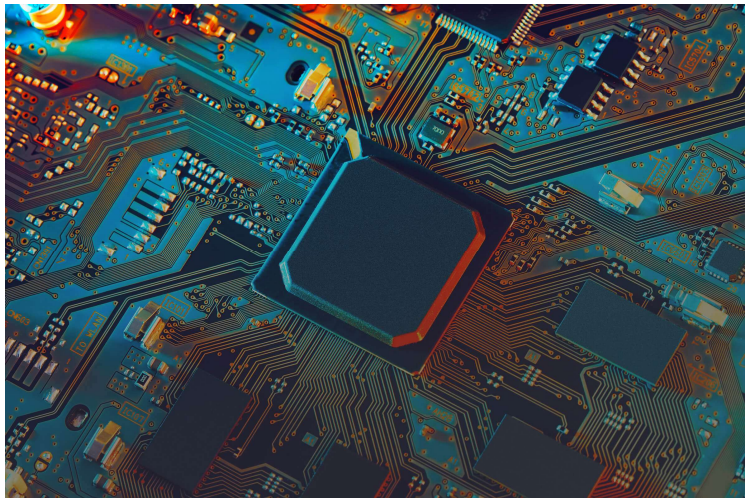
Functional units – Basic operational concepts – Bus Structures – **Performance** – **Memory locations and addresses** – Memory operations – Instruction and Instruction sequencing –– Addressing modes – Assembly language – Case study : RISC and CISC Architecture.



Recall the previous class concepts

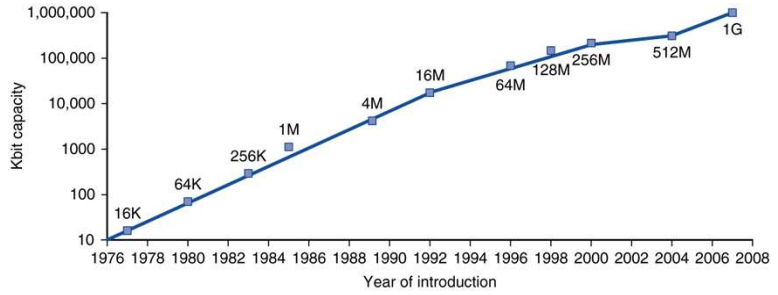


Technology & Performance



Technology Trends

- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost



DRAM capacity

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2005	Ultra large scale IC	6,200,000,000

Response Time and Throughput

Response time : How long it takes to do a task

Throughput : Total work done per unit time

Example : tasks/transactions/… per hour

How are response time and throughput affected by

- Replacing the processor with a faster version?
- Adding more processors?

Decreasing a response Time

Relative Performance

Define **Performance = 1/Execution Time**

"X is *n* time faster than Y"

$$\text{Performance}_X / \text{Performance}_Y = \text{Execution time}_Y / \text{Execution time}_X = n$$

If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

Performance ratio n = 15 /10 = 1.5

A is therefore 1.5 times faster than B.

Measuring Execution Time

Elapsed time : Total response time - Determines system performance

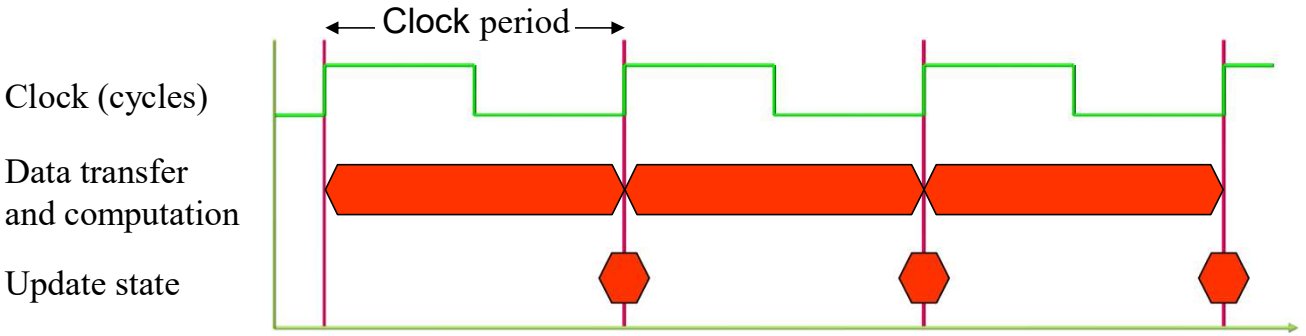
Example : Processing, I/O, OS overhead, idle time

CPU time: Time spent processing a given job . Comprises user CPU time and system CPU time

Different programs are affected differently by CPU and system performance

CPU Clocking

Operation of digital hardware governed by a constant-rate clock



Clock period: duration of a clock cycle

Clock frequency (rate): cycles per second

CPU Performance & its Factor

$$\text{CPU Execution Time} = \text{CPU Clock Cycles} \times \text{Clock Cycle Time}$$

$$= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}$$

$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$$

Performance improved by

- Reducing number of clock cycles
- Increasing clock rate
- Hardware designer must often trade off clock rate against cycle count

Improving Performance

Formula

Our favorite program runs in **10 seconds on computer A**, which has a **2 GHz clock**. We are trying to help a computer designer build a **computer, B, which will run this program in 6 seconds.** The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing **computer B to require 1.2 times as many clock cycles as computer A for this program**. What clock rate should we tell the designer to target?

FORMULA

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A}$$

$$10 \text{ seconds} = \frac{\text{CPU clock cycles}_A}{2 \times 10^9 \frac{\text{cycles}}{\text{second}}}$$

20×10^9 cycles

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B}$$

$$6 \text{ seconds} = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{\text{Clock rate}_B}$$

$\text{Clock Rate}_B = 4$ GHz

To run the program in 6 seconds , B must have twice the clock rate of A

Instruction Performance

$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$$

Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500 ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much?

CPU clock cycles $_A$ = I x 2.0
CPU clock cycles $_B$ = I x 1.2

Compute the CPU time for each computer

CPU time $_A$ = CPU clock cycles $_A$ × Clock cycle time

CPU time $_A$ = 500 x I ps
CPU time $_B$ = 600 x I ps

$$\frac{\text{CPU performance}_A}{\text{CPU performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = \frac{600 \times I \text{ ps}}{500 \times I \text{ ps}}$$

Computer A is 1.2 times as fast as computer B

Classical CPU Performance Equation

$$\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$$

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$$

$$= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$$

Comparing Code Segments

A compiler designer is trying to decide between two code sequences for a particular computer. The hardware designers have supplied the following facts:

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

Sequence 1: IC = 5

Clock Cycles = 10
(2×1 + 1×2 + 2×3)

CPI = clock cycle / IC = 10/5 = 2.0

Sequence 2: IC = 6

Clock Cycles = 9
(4×1 + 1×2 + 1×3)

CPI = 9/6 = 1.5

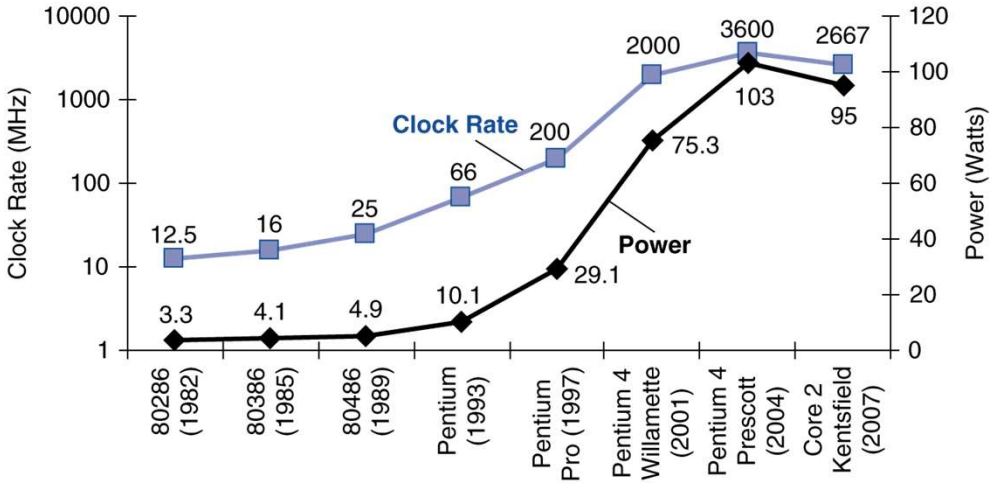
Performance Summary

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$$

$$\text{CPU Time} = s \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

- Performance depends on
  - Algorithm: affects IC, possibly CPI
  - Programming language: affects IC, CPI
  - Compiler: affects IC, CPI
  - Instruction set architecture: affects IC, CPI, T_c

Power Trends



In CMOS IC technology **Complementary Metal-Oxide Semiconductor**

$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

×30 | 5V → 1V | ×1000

Memory locations and addresses

- Determine how the computer's **memory is organized so that the user can efficiently store or retrieve information** from the computer.
- Memory consists of many **millions of storage cells (flip-flops).**
- Each cell can **store a bit of information i.e. 0 or 1**

Memory locations and addresses

- Each group of n bits is referred to as a word of information, and n is called the word length.
- The word length can vary from 8 to 64 bits.
- A unit of 8 bits is called a byte.
- Accessing the memory to store or retrieve a single item of information (word/byte) requires distinct addresses for each item location. (0 through 2^k-1 as the addresses of successive-locations in the memory).

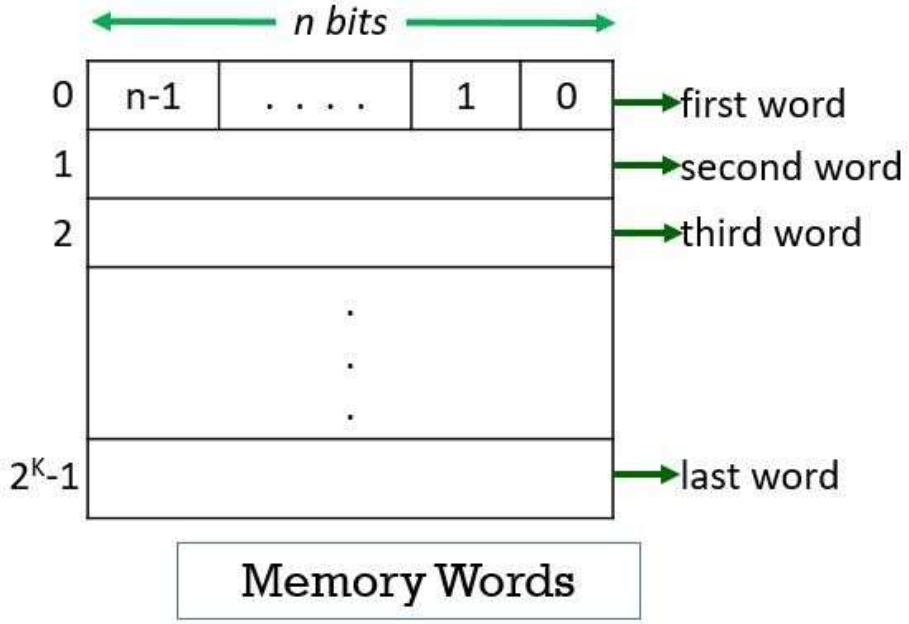
Memory locations and addresses

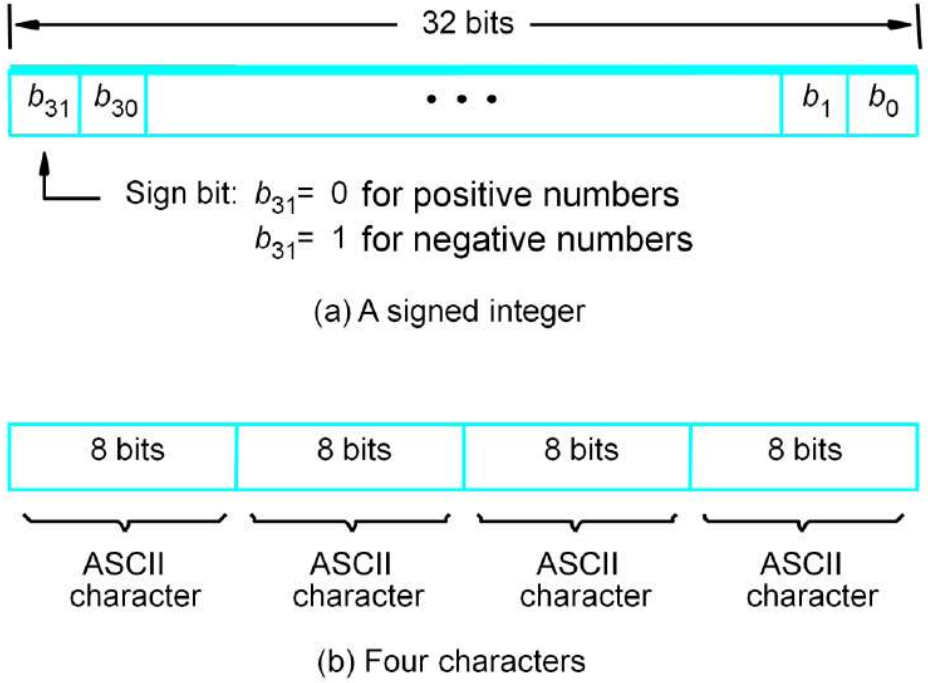
- If 2^{k} = no. of addressable locations; then 2^k addresses constitute the address-space of the computer.
- For example, a 24-bit address generates an address-space of 2^24 locations (16 MB).
- A 32-bit address creates an address space of 2^{32} or 4G (4 giga) locations.

Byte Addressability

- A byte is always 8 bits, but the word length typically ranges from 16 to 64 bits.
- In byte-addressable memory, successive addresses refer to successive byte locations in the memory.
- Byte locations have addresses 0, 1, 2. . . . .
- If the word-length is 32 bits, successive words are located at addresses 0, 4, 8. . with each word having 4 bytes.

Byte Addressability





ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

Big-Endian and Little-Endian Assignments in Byte Addresses

- Big-Endian – is used when lower byte addresses are used for the more significant bytes of the word
- Little- Endian – is used when lower byte addresses are used for the less significant bytes of the word

Storage of the value $D7C4_{16}$

Big Endian

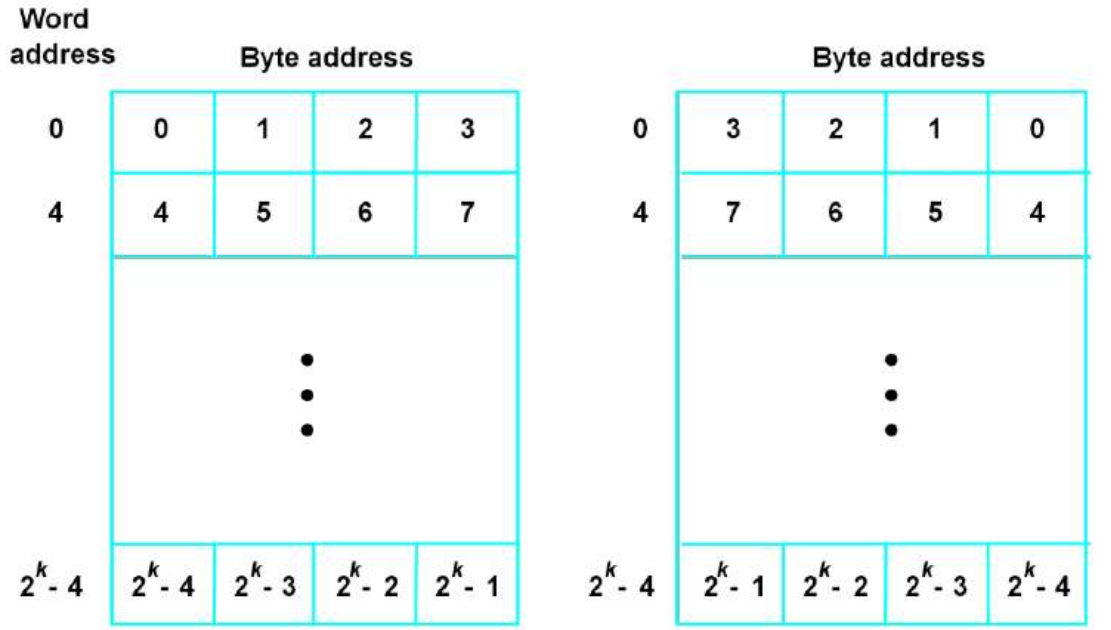
Motorola Processors: 68000, 68030, etc...

0	1
D7	C4

Little Endian

Intel Processors: 80386, Pentium, etc...

0	1
C4	D7



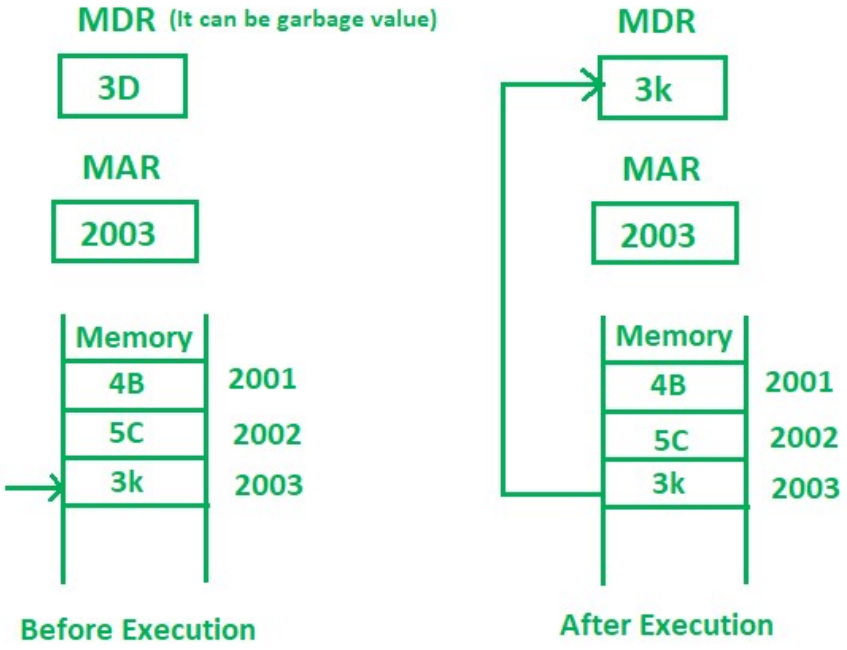
(a) Big-endian assignment (b) Little-endian assignment

Memory Operations

- Two basic operations are:
  - ✓ Load (or Read or Fetch)
  - ✓ Store (or Write)

Load Operations

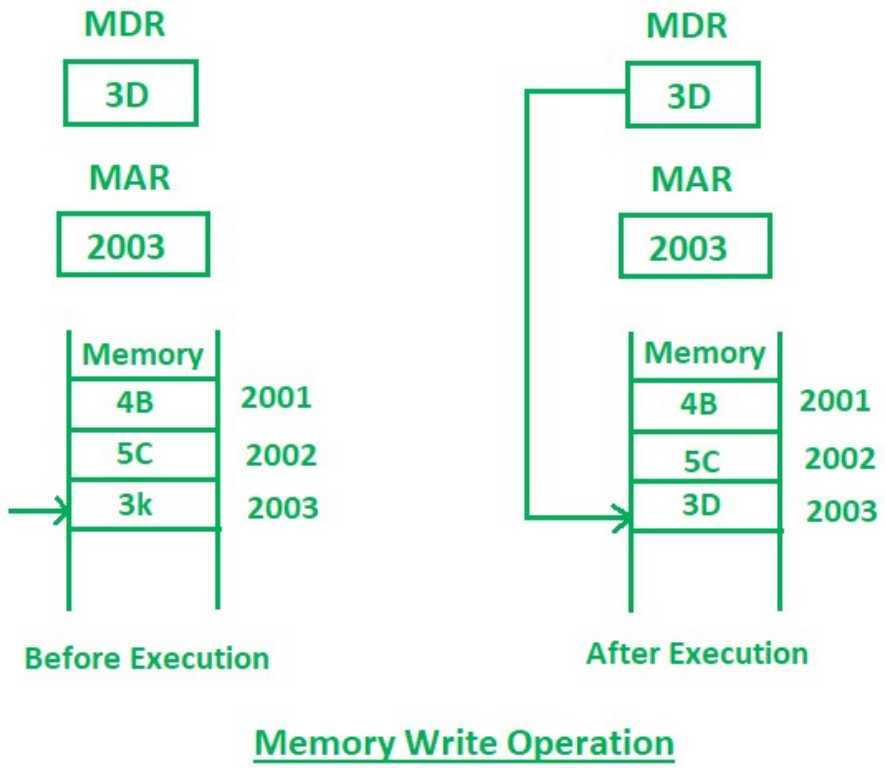
- Transfers a copy of the contents of a specific memory location to the processor
- Memory contents remain unchanged
- Processor sends the address of the desired location to the memory and request that its contents be read
- Memory reads the data stored at that address and sends them to the processor



Memory Read Operation

Store Operations

- Transfers an item of information from the processor to a specific memory location, destroying the former contents of that location
- Processor sends the address of the desired location to the memory, together with the data to be written into that location



Memory Write Operation

QUIZ

1. What is computer architecture?

   a) set of categories and methods that specify the functioning, organization, and implementation of computer systems

   b) set of principles and methods that specify the functioning, organization, and implementation of computer systems

   c) set of functions and methods that specify the functioning, organization, and implementation of computer systems

   d) None of the mentioned

QUIZ

2. To reduce the memory access time we generally make use of _____

a) SDRAM's

b) Heaps

c) Cache's

d) Higher capacity RAM's

QUIZ

3. Who developed the basic architecture of computer?

a) Blaise Pascal

b) Charles Babbage

c) John Von Neumann

d) None of the above

QUIZ

4. Which of the following allows simultaneous write and read operations?

a) ROM

b) EROM

c) RAM

d) None of the above

QUIZ

5. Computer address bus is

a) Multidirectional

b) Bidirectional

c) Unidirectional

d) None of the above

QUIZ

6. Which of the following is a way in which the components of a computer are connected to each other?

a) Computer parts

b) Computer architecture

c) Computer hardware

d) None of the above

QUIZ

7. Which of the following circuit convert the binary data into a decimal?

a) Decoder

b) Encoder

c) Code converter

d) Multiplexer

QUIZ

8. The address in the main memory is known as -

a) Logical address

b) Physical address

c) Memory address

d) None of the above

QUIZ

9. The collection of 8-bits is called as -

a) Byte

b) Nibble

c) Word

d) Record

QUIZ

10. Which of the following register can interact with the secondary storage?

a) PC

b) MAR

c) MDR

d) IR

QUIZ

11. Which of the following register keeps track of the instructions stored in the program stored in memory?

a) Accumulator

b) Address Register

c) Program Counter

d) Index Register

QUIZ

12. Which of the following is correct about memory and storage?

a) Memory is temporary, Storage is temporary

b) Memory is temporary, Storage is permanent

c) Memory is permanent, Storage is temporary

d) Memory is slow, Storage is Fast


Thank You