MOULE 1: Basic Structure of Computers
Machine Instructions and Pograms

A computer can be defined as a fast electronic calculating machine thatcomaccepts the (data) digitized input information process it as per the list of internally st red instructions and produces the resulting information.List of instructions are called programs & internl storage is called computer memory.

Computer Architecture in general covers three aspects of computer design namely:

Computer Hardware, Instruction set Architecture and Computer Organization.

Computer hardware consists of electronic circuits, displays, magnetic and optical storage media and communication facilities.

Instruction set Architecture is programmer visible machine interface such as instruction set, registers, memory organization and exception handling

Computer Types

Computer is a fast electronic calculating machine which accepts digital input, processes it according to the internally stored instructions (Programs) and produces the result on the output device. The internal operation of the computer can be as depicted in the figure below:

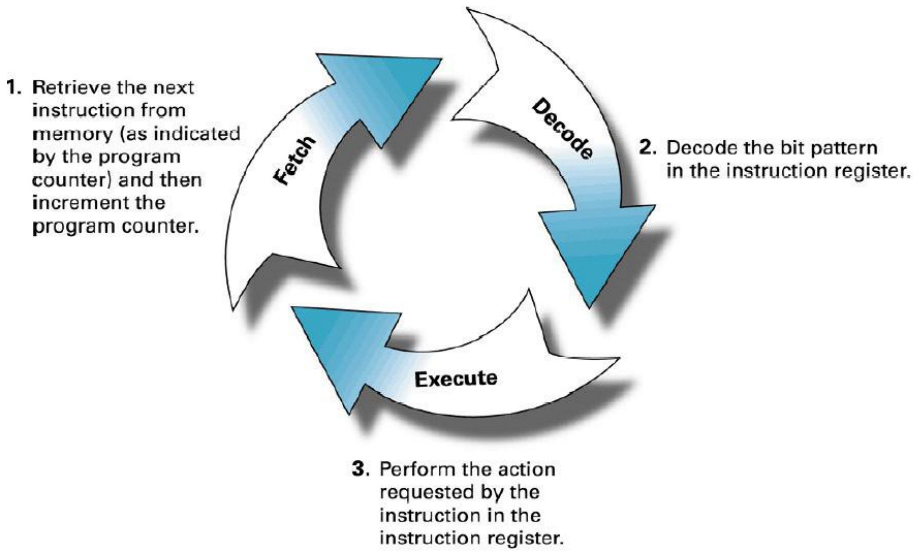

1. Retrieve the next instruction from memory (as indicated by the program counter) and then increment the program counter.
Fetch
Decode
2. Decode the bit pattern in the instruction register.
Execute
3. Perform the action requested by the instruction in the instruction register.

Figure 1: Fetch, Decode and Execute steps in a Computer System

The computers can be classified into various categoriesas given below:

- Micro Computer
- Laptop Computer
- Work Station
- Super Computer
- Main Frame
- Hand Held
- Multi core

Micro Computer: A personal computer; designed to meet the computer needs of an individual. Provides access to a wide variety of computing applications, such as word processing, photo editing, e-mail, and internet.

Laptop Computer: A portable, compact computer that can run on power supply or a battery unit. All components are integrated as one compact unit. It is generally more expensive than a comparable desktop. It is also called a Notebook.

Work Station: Powerful desktop computer designed for specialized tasks. Generally used for tasks that requires a lot of processing speed. Can also be an ordinary personal computer attached to a LAN (local area network).

Super Computer: A computer that is considered to be fastest in the world. Used to execute tasks that would take lot of time for other computers. For Ex: Modeling weather systems, genome sequence, etc (Refer site: http://www.top500.org/)

Main Frame: Large expensive computer capable of simultaneously processing data for hundreds or thousands of users. Used to store, manage, and process large amounts of data that need to be reliable, secure, and centralized.

Hand Held: It is also called a PDA (Personal Digital Assistant). A computer that fits into a pocket, runs on batteries, and is used while holding the unit in your hand. Typically used as an appointment book, address book, calculator and notepad.

Multi Core: Have Multiple Cores – parallel computing platforms. Many Cores or computing elements in a single chip. Typical Examples: Sony Play station, Core 2 Duo, i3, i7 etc.

Functional Unit

A computer in its simplest form comprises five functional units namely input unit, output unit memory unit, arithmetic & logic unit and control unit. Figure 2 depicts the functional units of a computer system.

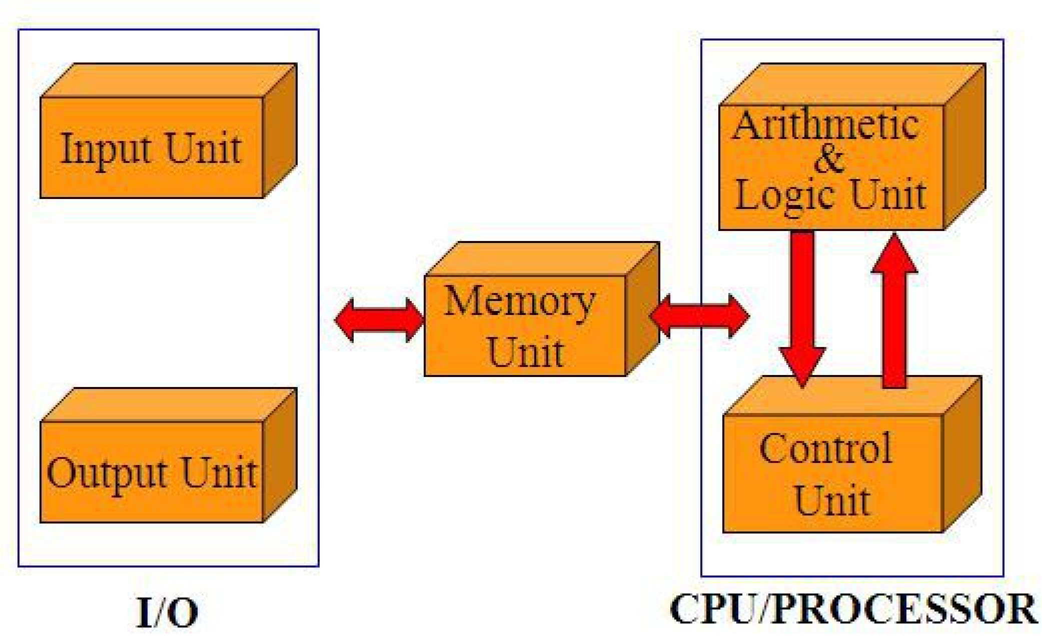


Figure 2: Basic functional units of a computer

Let us discuss about each of them in brief:

1. **Input Unit:** Computer accepts encoded information through input unit. The standard input device is a keyboard. Whenever a key is pressed, keyboard controller sends the code to CPU/Memory.

   Examples include Mouse, Joystick, Tracker ball, Light pen, Digitizer, Scanner etc.

2. **Memory Unit:** Memory unit stores the program instructions (Code), data and results of computations etc. Memory unit is classified as:
   - Primary /Main Memory
   - Secondary /Auxiliary Memory

Primary memory is a semiconductor memory that provides access at high speed. Run time program instructions and operands are stored in the main memory. Main memory is classified again as ROM and RAM. ROM holds system programs and firmware routines such as BIOS, POST, I/O Drivers that are essential to manage the hardware of a computer. RAM is termed as Read/Write memory or user memory that holds run time program instruction and data. While primary storage is essential, it is volatile in nature and expensive. Additional requirement of memory could be supplied as auxiliary memory at cheaper cost. **Secondary memories** are non volatile in nature.

3. **Arithmetic and logic unit:** ALU consist of necessary logic circuits like adder, comparator etc., to perform operations of addition, multiplication, comparison of two numbers etc.

4. **Output Unit:** Computer after computation returns the computed results, error messages, etc. via output unit. The standard output device is a video monitor, LCD/TFT monitor. Other output devices are printers, plotters etc.

5. **Control Unit:** Control unit co-ordinates activities of all units by issuing control signals. Control signals issued by control unit govern the data transfers and then appropriate operations take place. Control unit interprets or decides the operation/action to be performed.

The operations of a computer can be summarized as follows:

1. A set of instructions called a program reside in the main memory of computer.

2. The CPU fetches those instructions sequentially one-by-one from the main memory, decodes them and performs the specified operation on associated data operands in ALU.

3. Processed data and results will be displayed on an output unit.

4. All activities pertaining to processing and data movement inside the computer machine are governed by control unit.

Basic Operational Concepts

An Instruction consists of two parts, an Operation code and operand/s as shown below:

OPCODE	OPERAND/s

Let us see a typical instruction

ADD LOCA, R0

This instruction is an addition operation. The following are the steps to execute the instruction:

Step 1: Fetch the instruction from main memory into the processor
Step 2: Fetch the operand at location LOCA from main memory into the processor

Step 3: Add the memory operand (i.e. fetched contents of LOCA) to the contents of register

R0 Step 4: Store the result (sum) in R0.

The same instruction can be realized using two instructions as

Load LOCA, R1

Add R1,R0

The steps to execute the instructions can be enumerated as below:

Step 1: Fetch the instruction from main memory into the processor
Step 2: Fetch the operand at location LOCA from main memory into
the processor Register R1
Step 3: Add the content of Register R1 and the contents of register R0
Step 4: Store the result (sum) in R0.

Figure 3 below shows how the memory and the processor are connected. As shown in the diagram, in addition to the ALU and the control circuitry, the processor contains a number of registers used for several different purposes. The instruction register holds the instruction that is currently being executed. The program counter keeps track of the execution of the program. It contains the memory address of the next instruction to be fetched and executed. There are n general purpose registers R0 to R_{n-1} which can be used by the programmers during writing programs.

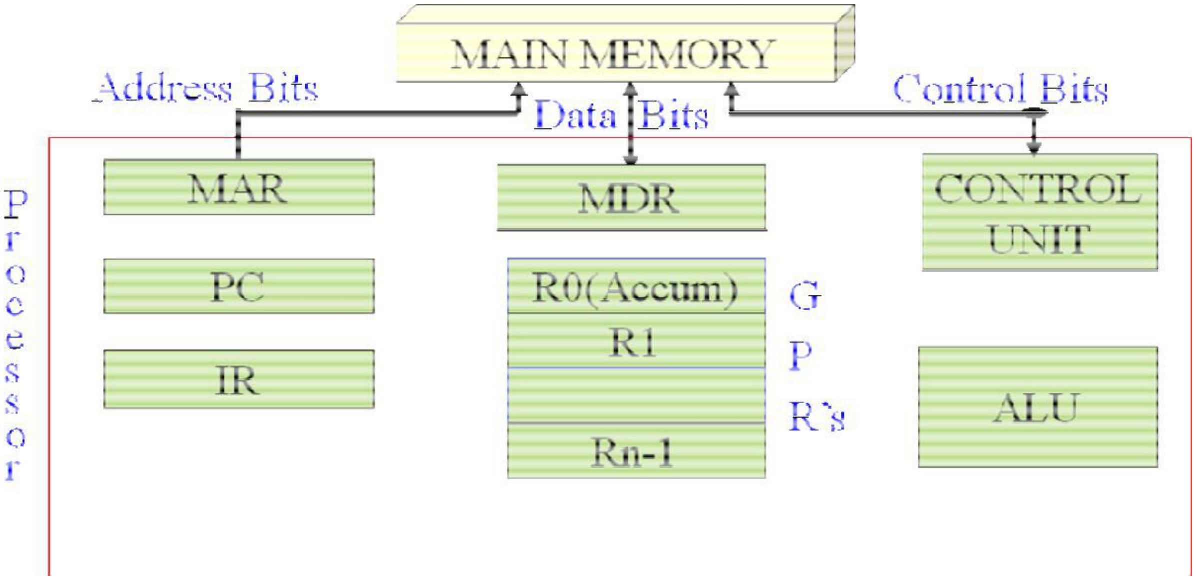


Figure 3: Connections between the processor and the memory

The interaction between the processor and the memory and the direction of flow of information is as shown in the diagram below:

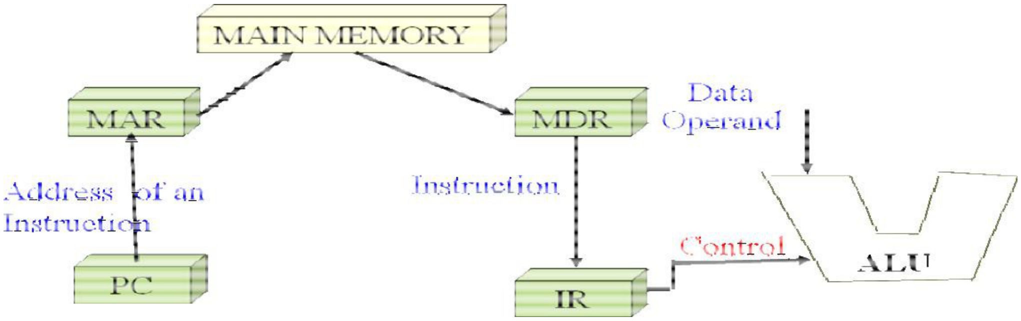


Figure 4: Interaction between the memory and the ALU

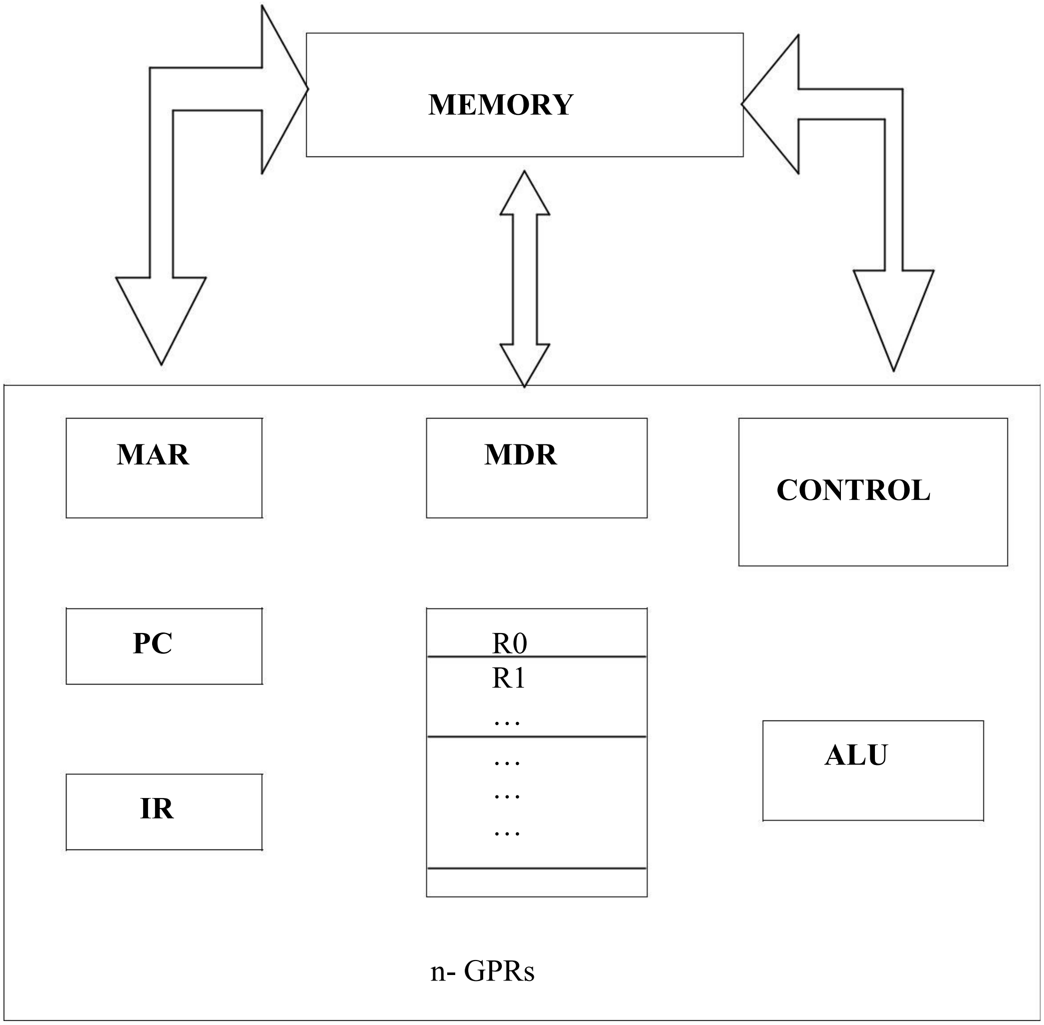


Fig b : Connections between the processor and the memory

The fig shows how memory & the processor can be connected. In addition to the ALU & the control c rcuitry, the processor contains a number of registers used for several different purposes.

The instruction register (IR):- Holds the instructions that is currently being executed. Its output is available for the control circuits which generates the timing signals that control the various processing elements in one execution of instruction.

The program counter PC:-

This is another specialized register that keeps track of execution of a program. It contains the memory address of the next instruction to be fetched and executed.

Besides IR and PC, there are n-general purpose registers R0 through R_{n-1}.

The other two registers which facilitate communication with memory are: -

6. **MAR – (Memory Address Register):-** It holds the address of the location to be accessed.
7. **MDR – (Memory Data Register):-** It contains the data to be written into or read out of the address location.

Operating steps are

5. Programs reside in the memory & usually get these through the I/P unit.
6. Execution of the program starts when the PC is set to point at the first instruction of the program.
7. Contents of PC are transferred to MAR and a Read Control Signal is sent to the memory.
8. After the time required to access the memory elapses, the address word is read out of the memory and loaded into the MDR.
9. Now contents of MDR are transferred to the IR & now the instruction is ready to be decoded and executed.
10. If the instruction involves an operation by the ALU, it is necessary to obtain the required operands.
11. An operand in the memory is fetched by sending its address to MAR & Initiating a read cycle.
12. When the operand has been read from the memory to the MDR, it is transferred from MDR to the ALU.
13. After one or two such repeated cycles, the ALU can perform the desired operation.

14. If the result of this operation is to be stored in the memory, the result is sent to MDR.

15. Address of location where the result is stored is sent to MAR & a write cycle is initiated.
16. The contents of PC are incremented so that PC points to the next instruction that is to be executed.

Normal execution of a program may be preempted (temporarily interrupted) if some devices require urgent servicing, to do this one device raises an Interrupt signal.

An interrupt is a request signal from an I/O device for service by the processor. The processor provides the requested service by executing an appropriate interrupt service routine.

If a user wants to enter and run an application program, he/she needs a System Software.

System Software is a collection of programs that are executed as needed to perform functions such as:

- Receiving and interpreting user commands
- Entering and editing application programs and storing then as files in secondary storage devices
- Running standard application programs such as word processors, spread sheets, games etc…

Operating system - is key system software component which helps the user to exploit the below underlying hardware

with the programs.

USER PROGRAM and OS ROUTINE INTERACTION

Let’s assume computer with 1 processor, 1 disk and 1 printer and application program is in machine code on disk. The various tasks are performed in a coordinated fashion, which is called multitasking. t0, t1 …t5 are the instances of time and the interaction during various instances as given below:

t0: the OS loads the program from the disk to memory

t1: program executes

t2: program accesses disk

t3: program executes some more

t4: program accesses printer

t5: program terminates

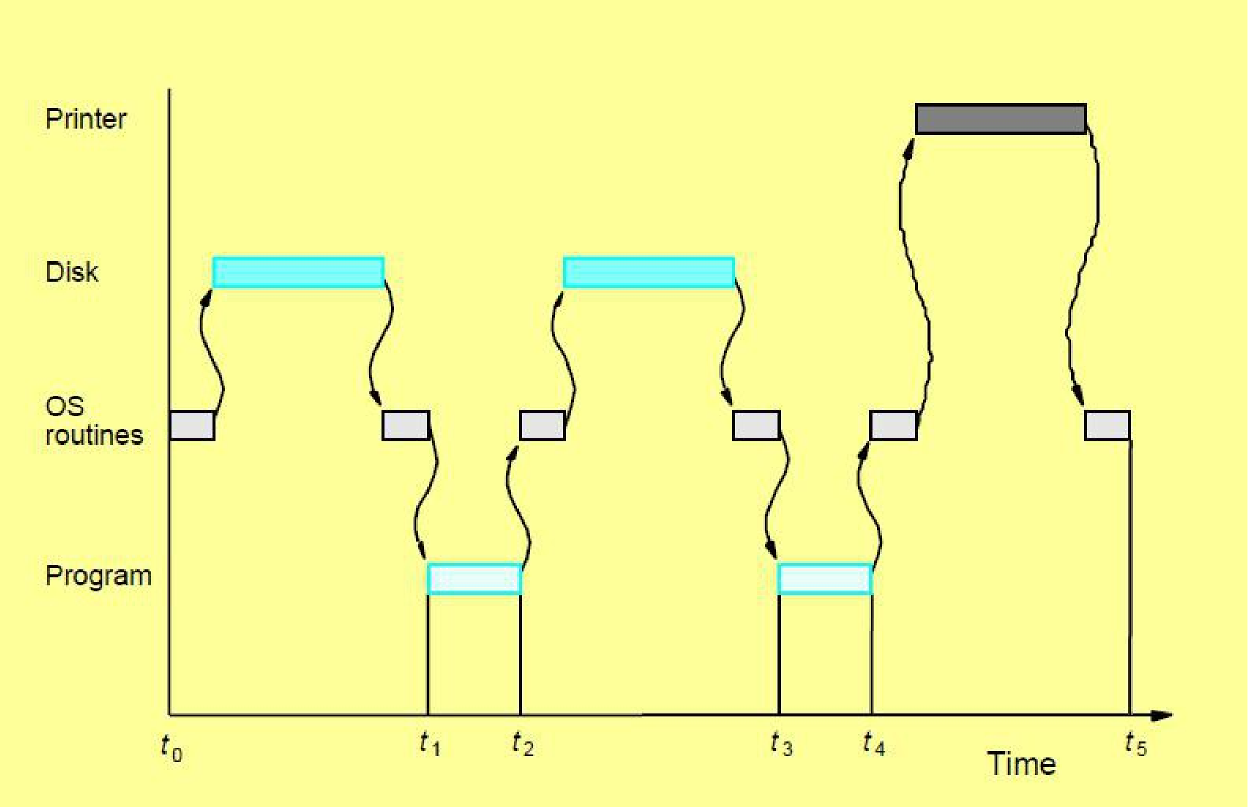


Figure 6 :User program and OS routine sharing of the processor

PERFORMANCE

The total time required to execute a program is the most important measure of performance for a computer. (t0-t5 of earlier example). Compiler, instruction set and hardware architecture, program all have impact on performance.

Performance

The most important measure of the performance of a computer is how quickly it can execute programs. The speed with which a computer executes program is affected by the design of its hardware. For best performance, it is necessary to design the compiles, the machine instruction set, and the hardware in a coordinated way.

The total time required to execute the program is elapsed time is a measure of the performance of the entire computer system. It is affected by the speed of the processor, the disk and the printer. The time needed to execute a instruction is called the processor time.

Just as the elapsed time for the execution of a program depends on all units in a computer system, the processor time depends on the hardware involved in the execution of individual machine instructions. This hardware comprises the processor and the memory which are usually connected by the bus as shown in the fig c.

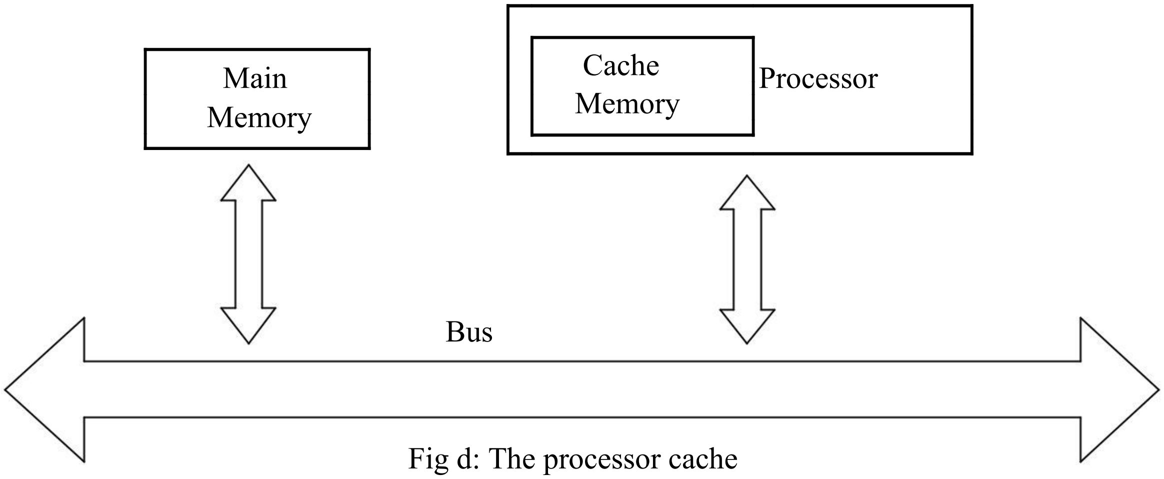


Fig d: The processor cache

The pertinent parts of the fig. c are repeated in fig. d which includes the cache memory as part of the processor unit.

Processor clock

Processor circuits are controlled by a timing signal called clock. The clock designer the regular time intervals called clock cycles. To execute a machine instruction the processor divides the action to be performed into a sequence of basic steps that each step can be completed in one clock cycle. The length P of one clock cycle is an important parameter that affects the processor performance.

Processor used in today's personal computer and work station have a clock rates that range from a few hundred million to over a billion cycles per second

Basic Performance Equation: The basic performance equation is given by

$$\mathbf{T = (N * S) / R}$$

where

T=execution time,
N=number of instructions,
S=average cycles per instruction,
R=clock rate in cycles per second

CACHING

Commonly used data are copied to on-processor memory (cache) to reduce access time.
Small memories can be made with higher speed than large ones. In a computer, we need both.

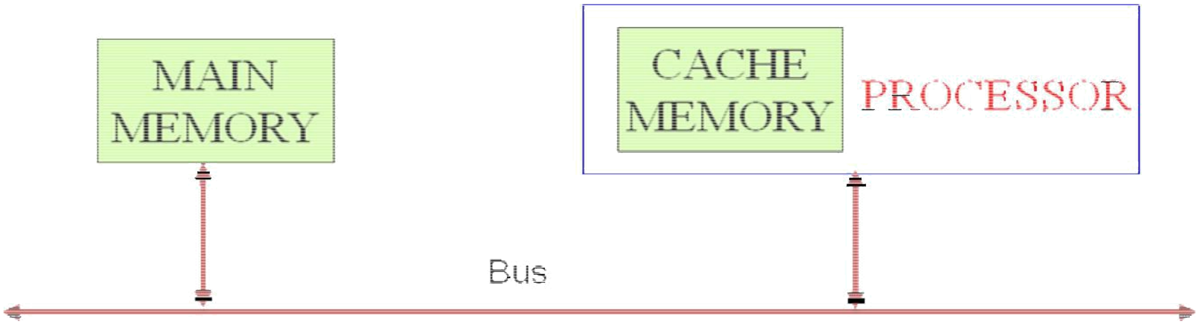


Figure 7: The processor cache

Clock rate

These are two possibilities for increasing the clock rate 'R'.

- Improving the IC technology makes logical circuit faster, which reduces the time of execution of basic steps. This allows the clock period P, to be reduced and the clock rate R to be increased.
- Reducing the amount of processing done in one basic step also makes it possible to reduce the clock period P. however if the actions that have to be performed by an instructions remain the same, the number of basic steps needed may increase.

Increase in the value 'R' that are entirely caused by improvements in IC technology affects all aspects of the processor's operation equally with the exception of the time it takes to access the main memory. In the presence of cache the percentage of accesses to the main memory is small. Hence much of the performance gain excepted from the use of faster technology can be realized.

PERFORMANCE MEASUREMENT

Benchmark refers to standard task used to measure how well a processor operates. To evaluate the performance of Computers, a non-profit organization known as SPEC-System Performance Evaluation Corporation employs agreed-upon application programs of real world for benchmarks. Accordingly, it gives performance measure for a computer as the time required to execute a given benchmark program. The SPEC rating is computed as follows

$$\text{SPEC Rating} - \frac{\text{Running time on a Reference Computer}}{\text{Running time on a test Computer}}$$

If the SPEC rating = 50 Means that the computer under test is 50 times as fast as the ultra sparc 10. This is repeated for all the programs in the SPEC suit, and the geometric mean of the result is computed.

MACHINE INSTRUCTIONS and PROGRAMS

NUMBER, ARITHMETIC OPERATIONS and CHARACTERS

Binary numbers (0, 1) are used in computers. Various number systems are used in computers. Numbers in binary are represented as vectors

$$B=bn-1\ldots b1b0$$

Unsigned numbers are in range 0 to 2n-1 and are represented by

$$V\ (B)=bn-1\times 2n-1+\ldots+b1\times 21+b0\times 2^{0}$$

Negative Numbers: They can be represented in various ways given below.

- Sign-and-magnitude

  Most significant bit determines sign, remaining unsigned bits represent magnitude

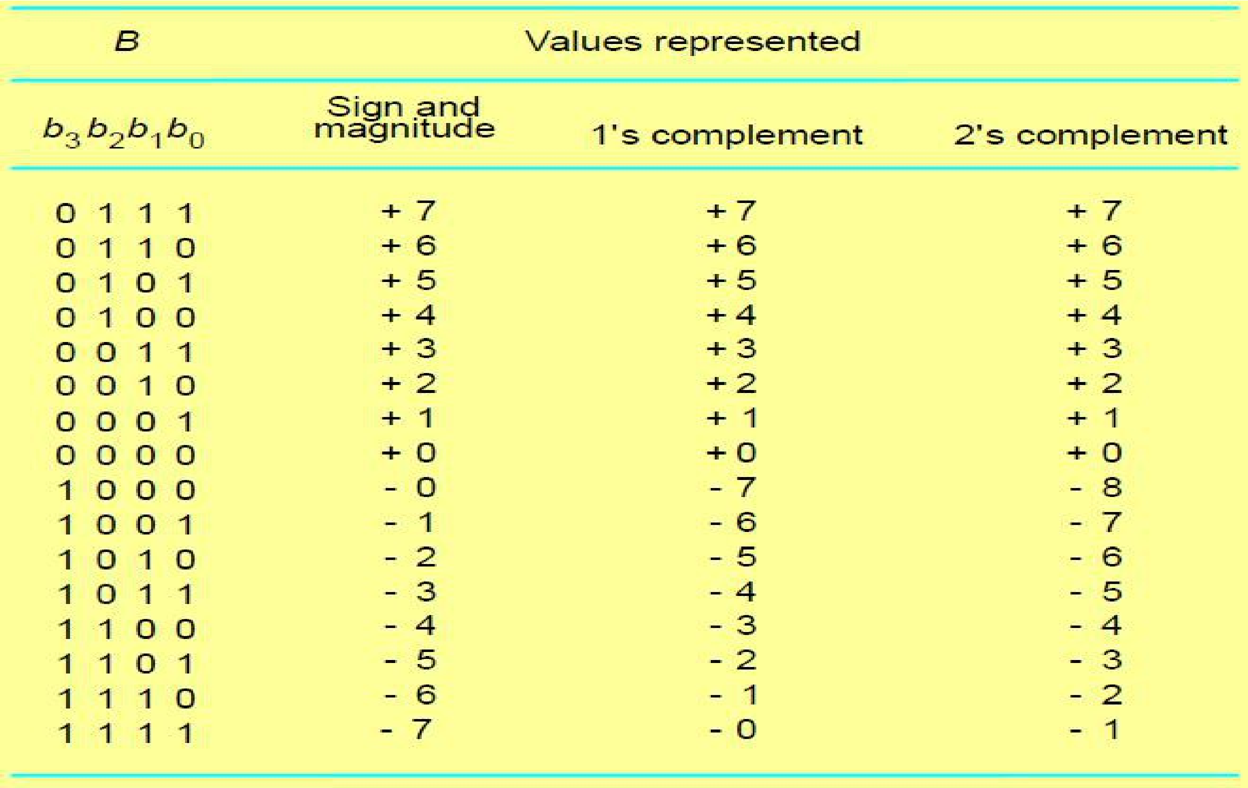
- 1’s complement

  Most significant bit determines sign. To change sign from unsigned to negative, invert all the bits

  (-3 is obtained by complementing each bit in vector 0011 to yield 1100).

- 2’s complement

  Most significant bit determines sign. To change sign from unsigned to negative, invert all the bits and add 1. This is equivalent to subtracting the positive number from 2n. The representations are as given in the table below



B	Values represented		
$b_3 b_2 b_1 b_0$	Sign and magnitude	1's complement	2's complement
0 1 1 1	+ 7	+ 7	+ 7
0 1 1 0	+ 6	+ 6	+ 6
0 1 0 1	+ 5	+ 5	+ 5
0 1 0 0	+ 4	+ 4	+ 4
0 0 1 1	+ 3	+ 3	+ 3
0 0 1 0	+ 2	+ 2	+ 2
0 0 0 1	+ 1	+ 1	+ 1
0 0 0 0	+ 0	+ 0	+ 0
1 0 0 0	- 0	- 7	- 8
1 0 0 1	- 1	- 6	- 7
1 0 1 0	- 2	- 5	- 6
1 0 1 1	- 3	- 4	- 5
1 1 0 0	- 4	- 3	- 4
1 1 0 1	- 5	- 2	- 3
1 1 1 0	- 6	- 1	- 2
1 1 1 1	- 7	- 0	- 1

Figure 1: Binary, signed integer representation

Addition & Subtraction of Signed Numbers: 3 systems of representing signed numbers

These systems differ only in the way they represent negative number

Sign and magnitude system – simplest representation

Most awkward for addition and subtraction. 1's complement method is somewhat better. 2's complement is the most efficient method.

Circle representation of Integer Mod N:

This is a graphical technique to compute (a+b) mod 16. This can be also used for addition involving signed numbers. Both the cases are shown below

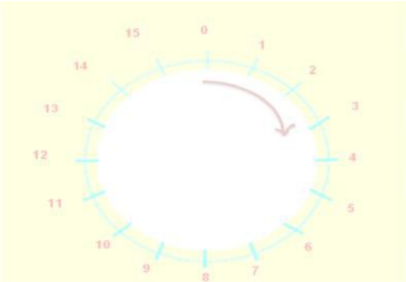
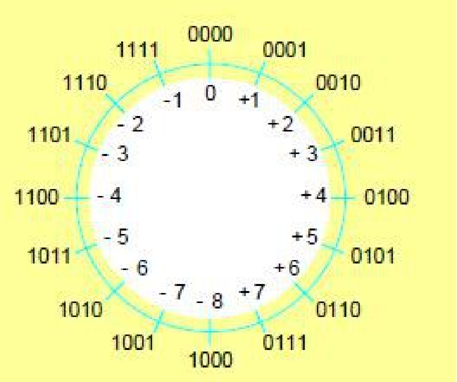


Figure 2:Circle representation of integer Mod 16

1) The operation (7+4) mod 16 yields the value 11.

To perform this graphically using the above representation locate 7 on the circle and then move 4 units in the clock wise direction to arrive at the answer 11.

2) consider adding +7 to -3. The representation is as shown below

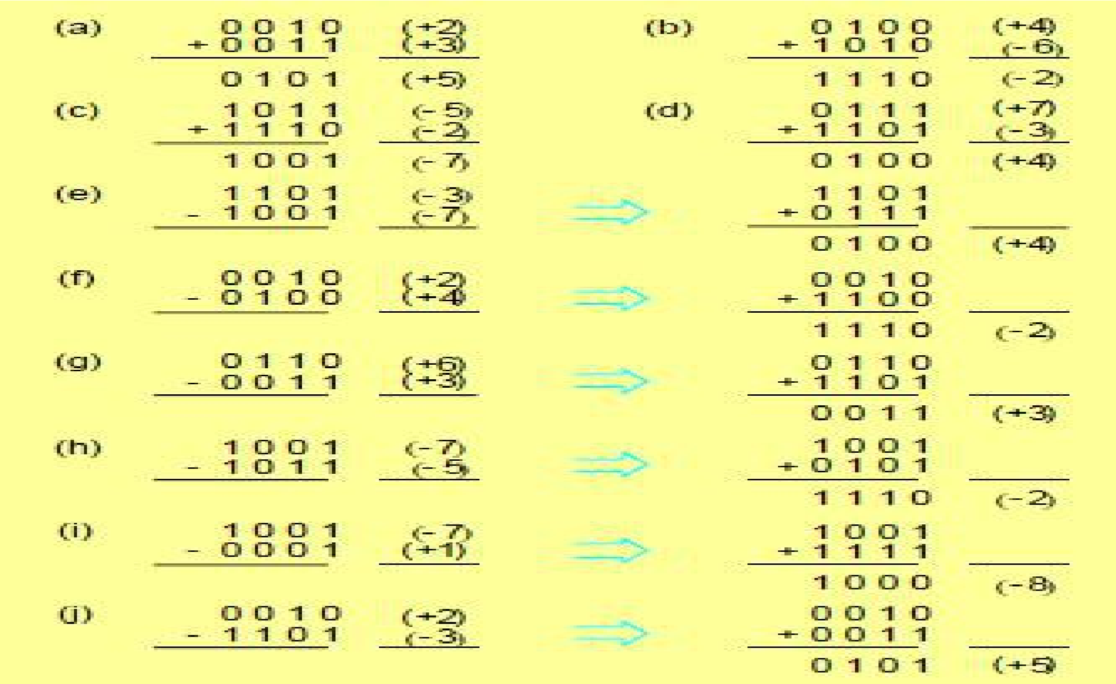


Mod 16 system for 2's-complement numbers

Figure 3: Mod 16 system for 2's complement numbers

2's complement representation for 7 is 0111 and -3 is 1101. Thus locate 0111 and then move 1101 (13 steps) in clockwise direction to arrive at 0100 = +4.

Some more examples of 2's complement add and substract operations are as shown in figure below:



```
(a)      0 0 1 0    (+2)            (b)      0 1 0 0    (+4)
       + 0 0 1 1    (+3)                   + 1 0 1 0    (-6)
         0 1 0 1    (+5)                     1 1 1 0    (-2)
(c)      1 0 1 1    (-5)            (d)      0 1 1 1    (+7)
       + 1 1 1 0    (-2)                   + 1 1 0 1    (-3)
         1 0 0 1    (-7)                     0 1 0 0    (+4)
(e)      1 1 0 1    (-3)                     1 1 0 1
       - 1 0 0 1    (-7)      ⇒            + 0 1 1 1
                                             0 1 0 0    (+4)
(f)      0 0 1 0    (+2)                     0 0 1 0
       - 0 1 0 0    (+4)      ⇒            + 1 1 0 0
                                             1 1 1 0    (-2)
(g)      0 1 1 0    (+6)                     0 1 1 0
       - 0 0 1 1    (+3)      ⇒            + 1 1 0 1
                                             0 0 1 1    (+3)
(h)      1 0 0 1    (-7)                     1 0 0 1
       - 1 0 1 1    (-5)      ⇒            + 0 1 0 1
                                             1 1 1 0    (-2)
(i)      1 0 0 1    (-7)                     1 0 0 1
       - 0 0 0 1    (+1)      ⇒            + 1 1 1 1
                                             1 0 0 0    (-8)
(j)      0 0 1 0    (+2)                     0 0 1 0
       - 1 1 0 1    (-3)      ⇒            + 0 0 1 1
                                             0 1 0 1    (+5)
```

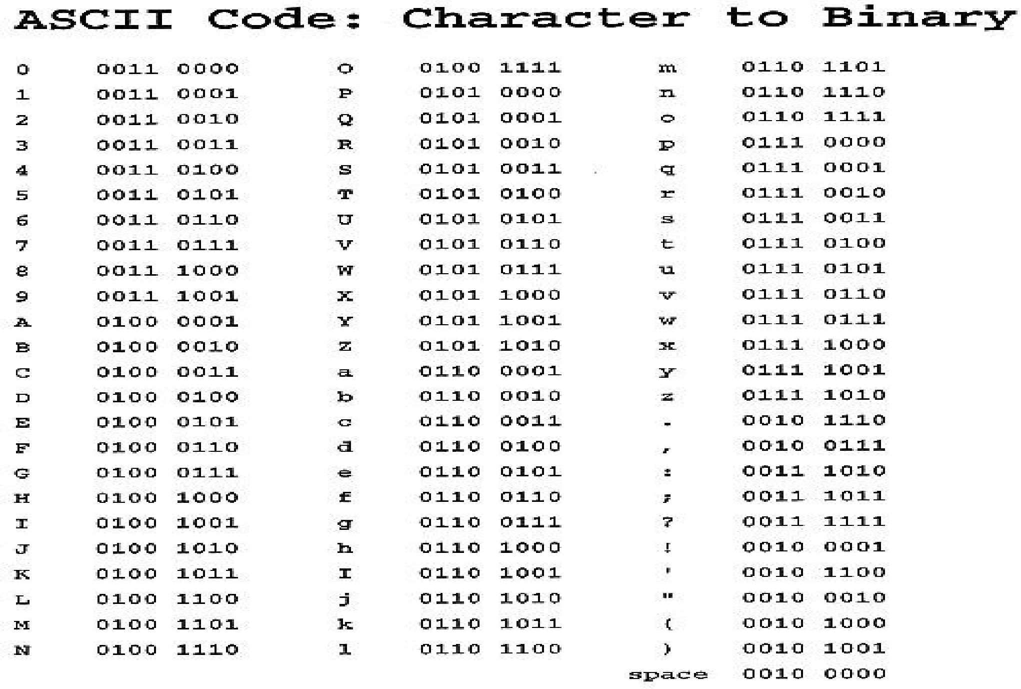
Figure 4: 2’s complement add and substract operations

Overflow in integer arithematic: In 2’s complement arithmetic addition of opposite sign numbers will never result in overflow . If the numbers are the same sign and the result is the opposite sign, overflow has occurred.

E.g. 0100 + 0111 =1011 (but 1011 is -5)

In case of unsigned numbers carry out signals that an overflow has occured

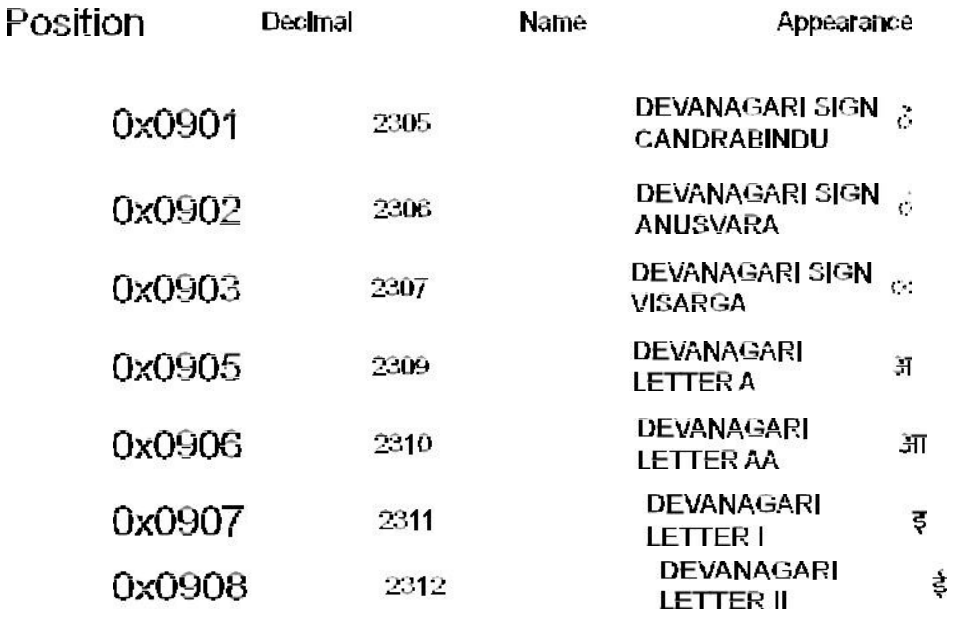
Characters: Apart from numbers computers must be able to handle alphanumeric text information consisting of characters . Characters can be letters of alphabets, decimal digits , punctuation marks etc. Most widely used code was ASCII and now a days unicode is being used widely.



ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	1	0110 1100	)	0010 1001
				space	0010 0000

Figure 5: ASCII Codes



Position	Decimal	Name	Appearance
0x0901	2305	DEVANAGARI SIGN CANDRABINDU	ँ
0x0902	2306	DEVANAGARI SIGN ANUSVARA	ं
0x0903	2307	DEVANAGARI SIGN VISARGA	ः
0x0905	2309	DEVANAGARI LETTER A	अ
0x0906	2310	DEVANAGARI LETTER AA	आ
0x0907	2311	DEVANAGARI LETTER I	इ
0x0908	2312	DEVANAGARI LETTER II	ई

Figure 6: Example of Uni Code

MEMORY LOCATIONS and ADDRESSES

Memory consists of storage cells. They store the bits 0 or 1. We can deal with them in n-bit groups called words (typically 8, 16, 32 or 64 bits). Usually refer to memory size in bytes e.g. we say we have 128MB memory and rarely use words as the unit. We use addresses to store or retrieve item of information For some k, memory consists of 2k unique addresses which range from 0 to 2k -1. The possible addresses are the address space of the computer.

E.g.

24-bit address has 2^{24} (16,777,216) locations.

Information quantities: bit, byte, word where Byte=8 bits,

word typically varies 16-64 bits. Most machines address memory in units of bytes.

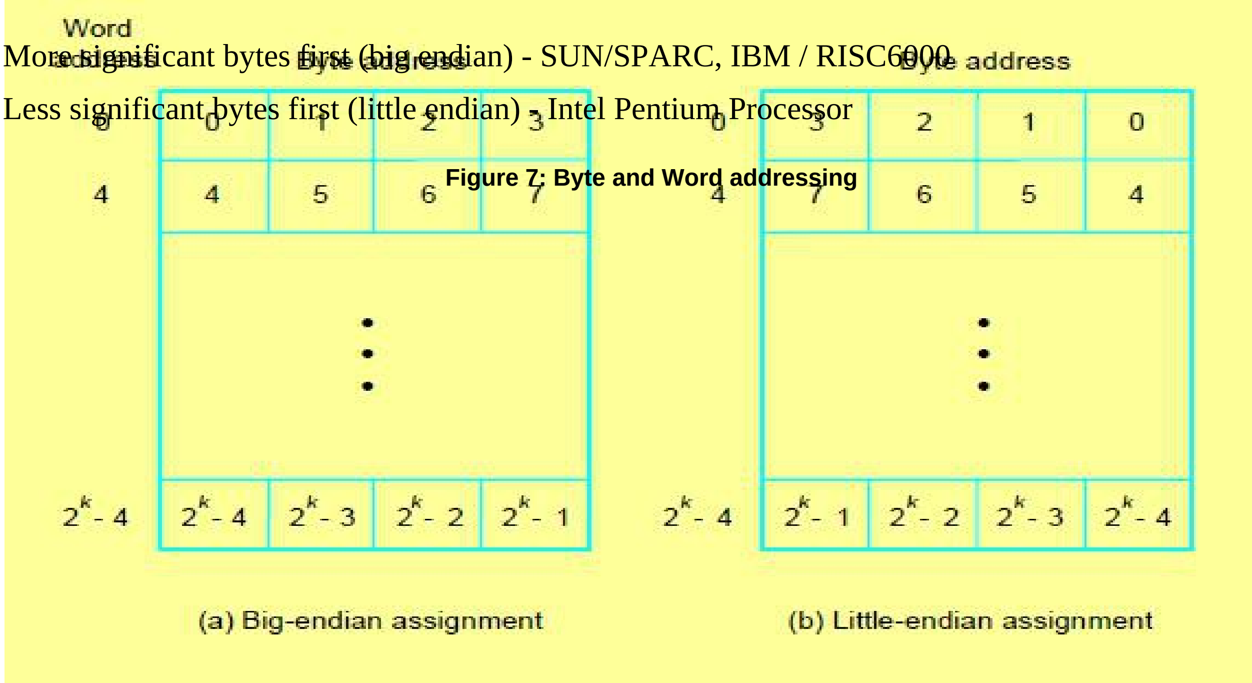
For a 32-bit machine, successive words are at address 0, 4, 8, 12 and so on.

Significant Bytes: Consider the hexadecimal (base 16) 32-bit number 34123F2A. This number is made up of four bytes 34, 12, 3F, 2A (4x8=32-bits). Bytes/bits with higher weighting are “more significant” i.e. the byte 34 is more significant than 2A. Bytes/bits with lower weighting are “less significant” i.e. 2A. Two ways byte addresses can be assigned across words

– More significant bytes first (big endian) - SUN/SPARC, IBM / RISC6000

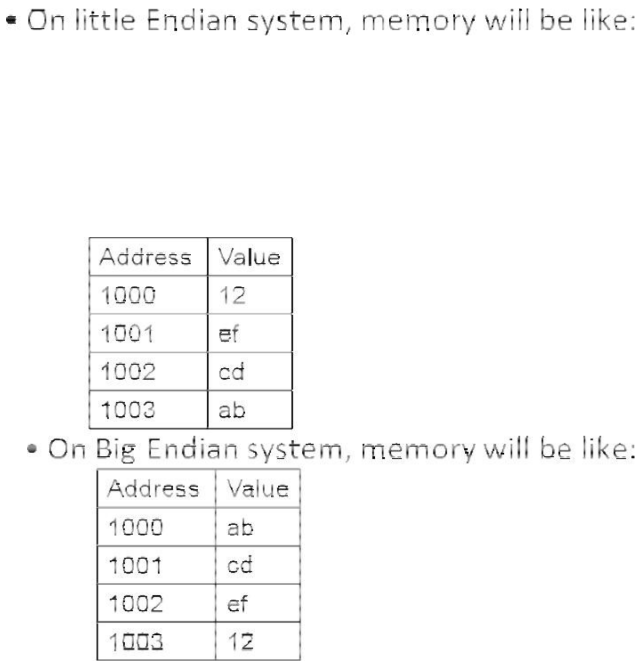
– Less significant bytes first (little endian) - Intel Pentium Processor

Figure 7: Byte and Word addressing


Word address
Byte address
0
0 1 2 3
4
4 5 6 7
2^k - 4
2^k - 4 2^k - 3 2^k - 2 2^k - 1
(a) Big-endian assignment
Byte address
0
3 2 1 0
4
7 6 5 4
2^k - 4
2^k - 1 2^k - 2 2^k - 3 2^k - 4
(b) Little-endian assignment

Big Endian and Little Endian:

Consider a 32 bit integer (in hex): 0xabcdef12. It consists of 4 bytes: ab, cd, ef, and 12. Hence this integer will occupy 4 bytes in memory. Say we store it at memory address starting 1000. There are 24 different orderings possible to store these 4 bytes in 4 locations (1000 - 1003). 2 among these 24 possibilities are very popular. These are called as little endian and big endian.



- On little Endian system, memory will be like:

Address	Value
1000	12
1001	ef
1002	cd
1003	ab

- On Big Endian system, memory will be like:

Address	Value
1000	ab
1001	cd
1002	ef
1003	12