Arithmetic and Logic Unit
Most computer operations are executed in the arithmetic and logic unit (ALU) of the processor. Any arithmetic or logic operation, such as addition, subtraction, multiplication division, or comparison of numbers, is initiated by bringing the required operands into the processor, where the operation is performed by the ALU.
When operands are brought into the processor, they are stored in high-speed storage elements called registers. Each register can store one word of data. Access times to registers are even shorter than access times to the cache unit on the processor chip.

Output Unit
Output unit function is to send processed results to the outside world. A familiar example of such a device is a printer. Most printers employ either photocopying techniques, as in laser printers, or ink jet streams. Such printers may generate output at speeds of 20 or more pages per minute. However, printers are mechanical devices, and as such are quite slow compared to the electronic speed of a processor. Some units, such as graphic displays, provide both an output function, showing text and graphics, and an input function, through touchscreen capability. The dual role of such units is the reason for using the single name input/output (I/O) unit in many cases.

Control Unit
The memory, arithmetic and logic, and I/O units store and process information and perform input and output operations. The operation of these units must be coordinated in some way. This is the responsibility of the control unit. The control unit is effectively the nerve center that sends control signals to other units and senses their states.
I/O transfers, consisting of input and output operations, are controlled by program instructions that identify the devices involved and the information to be transferred.
Control circuits are responsible for generating the timing signals that govern the transfers. They determine when a given action is to take place. Data transfers between the processor and the memory are also managed by the control unit through timing signals. A large set of control lines (wires) carries the signals used for timing and synchronization of events in all units.

The operation of a computer can be summarized as follows:
• The computer accepts information in the form of programs and data through an input unit and stores it in the memory.
• Information stored in the memory is fetched under program control into an arithmetic and logic unit, where it is processed.
• Processed information leaves the computer through an output unit.
• All activities in the computer are directed by the control unit.

Von Neumann architecture
In the 1940s, a mathematician called John Von Neumann described the basic arrangement (or architecture) of a computer. Most computers today follow the concept that he described although there are other types of architecture. A Von Neumann-based computer is a computer that:
Uses a single processor.
Uses one memory for both instructions and data. A von Neumann computer cannot distinguish between data and instructions in a memory location! It ‘knows’ only because of the location of a particular bit pattern in RAM.
Executes programs by doing one instruction after the next in a serial manner using a fetch-decode-execute cycle.

Number Representation and Arithmetic Operations
1.4.1 Integers
Consider an n-bit vector : B = bn−1 . . . b1b0 where bi = 0 or 1 for 0 ≤ i ≤ n − 1.

$$V(B) = b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

We need to represent both positive and negative numbers. Three systems are used for representing such numbers:
• Sign-and-magnitude
• 1's-complement
• 2's-complement

In all three systems, the leftmost bit is 0 for positive numbers and 1 for negative numbers.

B	Values represented		
$b_3b_2b_1b_0$	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

In *1's-complement* representation, negative values are obtained by complementing each bit of the corresponding positive number. Thus, the representation for −3 is obtained by complementing each bit in the vector 0011 to yield 1100.

In the *2's-complement* system, forming the 2's-complement of an *n*-bit number is done by subtracting the number from 2*n*. Hence, the 2's-complement of a number is obtained by adding 1 to the 1's-complement of that number.

There are distinct representations for +0 and −0 in both the sign-and magnitude and 1's-complement systems, but the 2's-complement system has only one representation for 0.

Addition of Unsigned Integers
The sum of 1 and 1 is the 2-bit vector 10, which represents the value 2. We say that the *sum* is 0 and the *carry-out* is 1. We add bit pairs starting from the low-order (right) end of the bit vectors, propagating carries toward the high-order (left) end. The carry-out from a bit pair becomes the *carry-in* to the next bit pair to the left.

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

Carry-out

Addition and Subtraction of Signed Integers
The 2's-complement system is the most efficient method for performing addition and subtraction operations.
Unsigned integers mod *N* is a circle with the values 0 through *N* − 1. The decimal values 0 through 15 are represented by their 4-bit binary values 0000 through 1111.