Chapter 1
Hardware

1 Basic Configuration of Computers

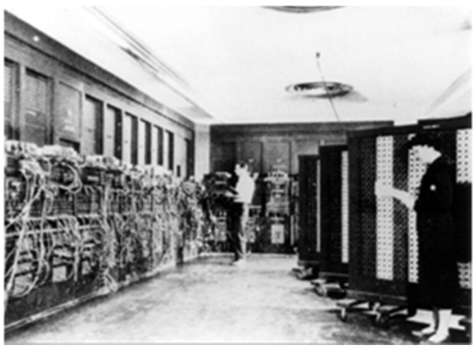
Various types of computers are extensively used around us. However, it has not even passed 100 years since the first computer was launched. Moreover, the basic configuration of computers has hardly changed since their initial launch. This chapter touches upon the history of computers from each generation and describes the five major units that form the basic configuration of a computer.

1 - 1 History of Computers

Generations of computers are divided according to the logic gates they used. A computer consists of components known as logic gates, which correspond to the brain of the computer that is used in arithmetic operations.

(1) 1st generation (1940s)

The world's first computer ENIAC was developed by J.W. Mauchly and J.P. Eckert in 1946. ENIAC used more than 18,000 vacuum tubes for its logic gates. Therefore, an enormous amount of heat was generated and the system consumed too much electricity for cooling, which even caused power outages. In those days, the computer was mainly used in ballistic calculations. However, it was necessary to replace circuit wiring according to the contents to be processed, and therefore, some people do not recognize it as a computer.
It was EDSAC, developed by M.V. Wilkes in 1949, which eliminated the replacement of circuit wiring required in ENIAC. EDSAC was a computer that used a stored-program system. The stored-program system stores the contents to be processed as a program inside the computer and then runs it, and computers based on this system are also known as Neumann computers from the name of its inventor (J. Von Neumann).



World's first computer: ENIAC

(2) 2nd generation (1950s)

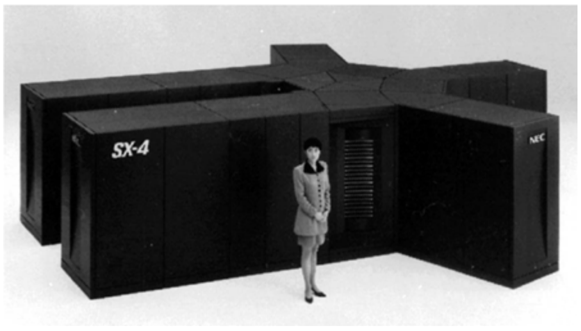
While 1st generation computers were mainly used in military and research and development, UNIVAC I was launched as a commercial computer in 1951. In this generation of computers, semiconductors like transistors were used as logic gates. Semiconductors have intermediate electrical properties between conductors that pass electricity and insulators that do not pass electricity. Besides transistors, semiconductor devices include the diode (rectifying device), which only passes the electric current in one direction, and LED (Light Emitting Diode), which emits light when voltage is applied in the forward direction. Semiconductors are smaller than vacuum tubes, and their failure occurrence rate is low. Therefore, computers became compact, and their reliability increased.

(3) 3rd generation (1960s)

In this generation of computers, IC (Integrated Circuit) was used as a logic gate. IC implements the processing capacity of several-hundred transistors in a square silicon chip of a few millimeters, thanks to the advancement of semiconductor device technology. Computers became extremely compact and fast thanks to IC technology, and several manufacturers announced various types of computers. A typical example is IBM/360 developed by the IBM Corporation. This computer was a general-purpose computer that was capable of handling any kind of processing without any restrictions on the usage purposes.

(4) 3.5th generation (1970s)

In this generation, IC technology made further progress, and manufacturers began using LSI (Large Scale Integration) as logic gates. LSI has higher integration density than IC, which enabled computers to become even more compact and faster. Since computers became compact, this led to the development of control computers that are used in industrial devices and microprocessors that are used in home appliance products. Moreover, their high-speed technology led to the development of super computers that are very useful for high-speed operations in the area of scientific and engineering computations. Moreover, microcomputers were also launched as small computers for individual use.



Super computer

(5) 4th generation (1980s)

Logic gate technology progressed further, and manufacturers began using VLSI (Very Large Scale Integration). Such exponential progress of hardware technology transformed the era of computers from "one computer in a company or facility" to "one computer per person." Manufacturers started developing and selling PC (Personal Computers) for individuals, which formed the foundation of the present-day information society.

Desktop computer

Laptop computer

In this era, the network environment was also developed, and it became common to use computers (servers) that provide services and terminals by connecting them through networks. In this usage, besides PCs, work stations that have higher performance than PCs were also used as terminals connected to networks. Moreover, the sizes of devices rapidly decreased with the development of the palm-sized PDA (Personal Digital Assistant) and smartphone; other personal digital assistants such as portable tablet terminals; and SoC (System on a Chip) and one chip microcomputers (single chip microcomputers) where functionality of computer is packed in one chip (LSI).

(6) 5th generation (?)

The progress of computer technology continues even today. For example, there is FPGA (Field Programmable Gate Array), which can be programmed after manufacture through simulation of a design diagram. FPGA is slower than a dedicated LSI, and it is also expensive. However, as compared to implementing similar functions in software, FPGA devices are high-speed and low-cost devices, which is why they are used in home appliance products, etc. With the advancement of various technologies, computers increasingly became compact and high performance, and concentrated efforts were made so that even beginners could use them easily. Moreover, as the next generation computers, computers (e.g., computers with inference function) closer to people and more closely linked to society are being developed. Furthermore, as part of a recent trend, computers with a focus on energy conservation are also being developed. Power consumption of computers is expressed in units of Watts (W) just like general electrical appliances. Each of the components used in computers consumes power. Therefore, researchers are working on design technologies combining components that have low power consumption.

1 - 2 Five Major Units of Computers

Computers are used in various fields, and there are various types of computers. However, the basic configuration elements are the same. This subsection describes the units that are used in computers.

In order to explain the configuration elements of computers, a list is created for the functions that are required for computers. The easiest way to understand this is to think in terms of human action and behavior. If we think about human behavior in solving the problem "3+6= $\boxed{}$", it becomes as follows:

1) Look at the problem and memorize it.

↓

2) Think about the meaning of "+".
…We know this is a sign that adds two values.

↓

3) Memorize "9" as a result of adding "3" and "6".

↓

4) Write "9" in the answer sheet.

Figure1-1
Action and behavior of humans

The five units of a computer perform this same action and behavior of humans. These units are called the five major units of a computer.

1) Input unit
   This is a unit that reads data to be processed by computer.
2) Output unit
   This is a unit that writes the processing results in a form that can be understood by humans.
3) Storage unit
   This is a unit that records data. There are main memory and auxiliary storage.
4) Arithmetic and logical unit
   This is a unit that performs arithmetic operations on data that is stored in the storage unit or makes a decision as per the instructions of the control unit.
5) Control unit
   This is a unit that interprets a command and gives instructions to the remaining four units.

Below is an explanation of these units by using a specific example. When the arithmetic operation "3+6= $\boxed{}$" is performed by a computer, a summary of the operation of each unit is as follows:

1) Enter "3+6" from the input unit, and store it in the storage unit.

↓

2) Control unit decodes the meaning of "+". As a result of decoding, it knows that "+" is a command for making addition. Therefore, it gives an instruction to the arithmetic and logical unit.

↓

3) The arithmetic and logical unit fetches "3" and "6" from the storage unit, and performs the calculation. The result "9" is stored in the storage unit.

↓

4) "9" is written from the storage unit to the output unit.

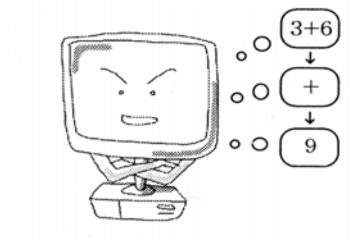


Figure 1-2
Functioning of a computer

Both input and output also operate on the basis of the instructions that are given by the control unit, which is not covered here. Therefore, the flow of data and control (instructions) between each unit is as shown in Figure 1-3 below.

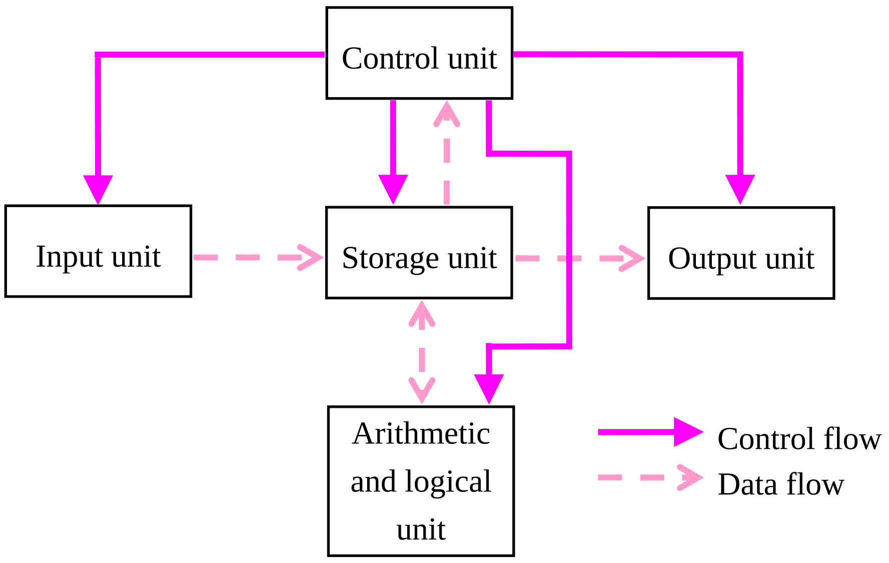
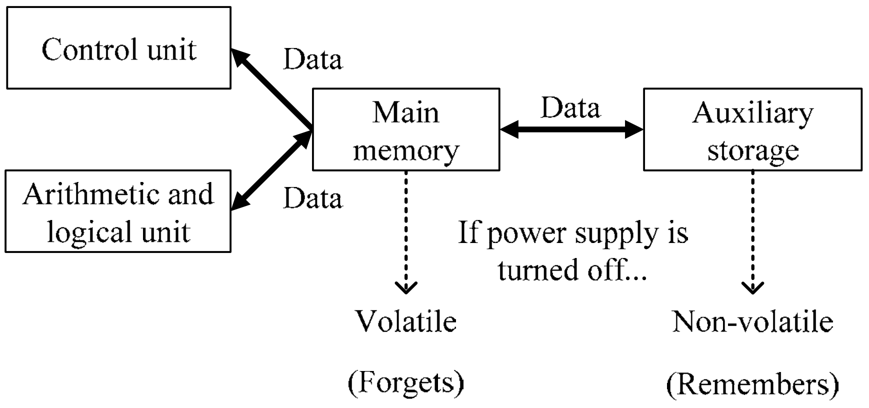


Figure1-3　Flow of control and data in five major units

In the storage unit, there is main memory and auxiliary storage. (In Figure 1-3, what is mentioned as the storage unit is main memory.) The differences between these two are described below.

- Main memory
  This unit can directly exchange data with the control unit and the arithmetic and logical unit, and it has a volatility characteristic (contents are lost if power supply is turned off).
- Auxiliary storage
  This unit stores data that cannot be accommodated in main memory, and it has a non-volatility characteristic (contents are not lost even if power supply is turned off).



Moreover, the control unit and the arithmetic and logical unit are collectively referred to as the CPU (Central Processing Unit) or the processor. The input unit, the output unit, and

the auxiliary storage outside the processor are also referred to as peripheral devices.
Present-day computers use microprocessors (or MPU (Micro Processing Unit)), where CPU (processor) functions are consolidated into one LSI. Microprocessors are also used in home electrical appliances other than PCs.
On the other hand, a semiconductor chip (LSI) where all required functions (system) including memory are integrated is referred to as SoC (System on a Chip). SoC has advantages such as high speed and low power consumption. However, it suffers from the disadvantage of high development risk. Therefore, it is also used in combination with SiP (System in a Package), which integrates multiple semiconductor chips into one package. Moreover, in a one chip microcomputer (single chip microcomputer) used in home electrical appliances, not only CPU and memory functions but also input and output functions are integrated into one semiconductor chip.
In addition, there are co-processors that perform only specific processes for assisting the processor, and dedicated processors that are focused only on specific processes, unlike general purpose processors that perform various processes like in PCs.

2 Data Representation in Computers

In order to make a computer work, it is necessary to store a program with a processing sequence and data to be processed inside the computer in advance. This section describes the methods of recording (representing) information inside computers.

2 - 1 Data Representation

(1) Unit of representation

Inside computers, data is recorded as electrical signals. Electric signals can basically represent only two states.

• Is electric current flowing?	⟷	Is it not flowing?
• Is voltage high?	⟷	Is it low?

0 and 1 are linked to these two states, and they are recorded and saved as data inside computers. In other words, data recorded by computers is represented with 0 or 1. The minimum unit that represents this 0 or 1 is called a bit. Various meanings are formed by combining 0 and 1 represented by this bit, and they are recorded as data. At this time, one unit formed by collecting 8 bits is called a byte.

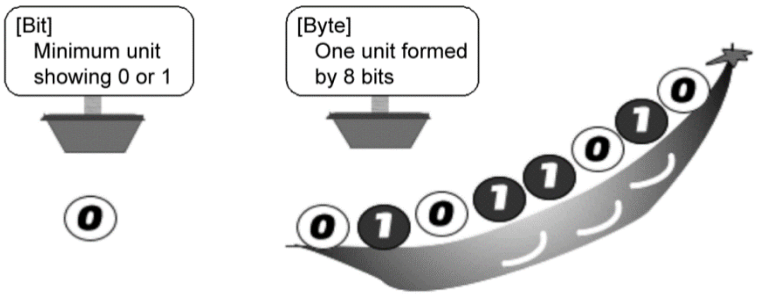


Figure 1-4 Bit and byte

Moreover, there is a unit called word, which is formed by collecting more bits than in a byte. Word is the unit of processing inside computers, and there are 16 bits, 32 bits, 64 bits, etc. according to the model of computer used. Needless to say that if more bits can be processed at a time, then more information can be processed in a certain time. Therefore, the higher the number of bits in one word, the higher the processing speed of the computer. In most present-day computers, 1 word is 32 bits or 64 bits.

(2) Information amount

There are two types (0, 1) of information that can be represented with 1 bit, and the number of bit combinations is referred to as the information amount. The information amount that can be represented with 2 bits is of 4 types (00, 01, 10, 11), the information amount that can be represented with 3 bits is of 8 types (000, 001, 010, 011, 100, 101, 110, 111), ... and so on.

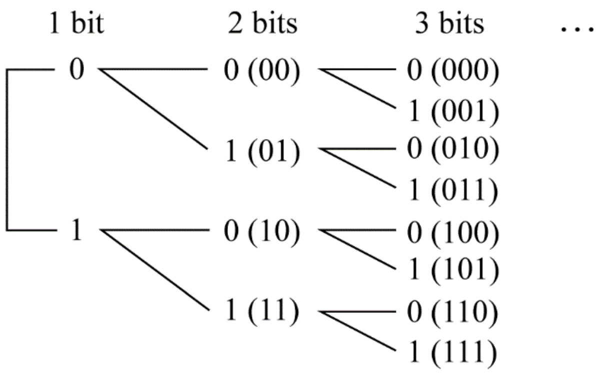


Figure1-5 Number of bits and information amount

As shown in Figure 1-5, branches increase with the increase of every one bit used, and the information amount that can be represented keeps on doubling. In other words, the information amount that can be represented with *n* bits is of 2^n types (2^n shows the power of 2, and means multiplying 2 by itself *n* times). Using this concept, a summary of the information amount that can be represented with byte and word (16 bits) is given below.

* Information amount of *n* bits = 2^n types

- Information amount that can be represented with 1 byte (=8 bits)
  = 2^8 types = 256 types (00000000 through 11111111)
- Information amount that can be represented with 1 word (=16 bits)
  = 2^{16} types = 65,536 types (0000000000000000 through 1111111111111111)

[Information amount in information theory]

In the field of information theory, when event x occurs with occurrence probability (probability that a certain event occurs) $P(x)$, the information amount that can be obtained is defined as $I(x)$. According to this definition, the smaller $P(x)$ is, the larger $I(x)$ is; and the larger $P(x)$ is, the smaller $I(x)$ is. (This is easy to understand if the information amount is considered as the degree of surprise when an event occurs.)
To explain the relation between bit and information amount, 1 bit represents the information of 0 or 1. In other words, the occurrence probability $P(x)$ of event x, where either 0 or 1 appears, is 1/2, and so the information amount of occurrence probability 1/2 is 1 bit. Therefore, information amount n bit is an event with occurrence probability $1/2^n$; in other words, it means information of 2^n types.

(3) Prefix

When data is handled, rather large numbers are handled as they are. However, it is very difficult to handle very large numbers or very small numbers without modification. Therefore, they are represented in combination with a prefix (auxiliary unit) representing a certain value.

For example, even if a nurse holding a newborn baby says “He is a healthy baby boy of 3,000g,” no one would say “I lately got a little fat and became 75,000g.” Normally, one would say “75 kg.” This “k” is the prefix. “k” is a prefix that means the value “10^3”, and it can represent values like “$75\text{kg} = 75\times10^3\text{g}$”.

The readings and values of major prefixes are summarized in the table below.

[Prefixes used for representing large numbers]

Symbol	Reading	Decimal value	Binary value
k	Kilo	10^3	2^{10}
M	Mega	10^6	2^{20}
G	Giga	10^9	2^{30}
T	Tera	10^{12}	2^{40}
P	Peta	10^{15}	2^{50}

[Prefixes used for representing small numbers]

Symbol	Reading	Decimal value
m	Milli	10^{-3}
M	Micro	10^{-6}
n	Nano	10^{-9}
p	Pico	10^{-12}

Here, for prefixes that represent large numbers, binary values are also shown in the table. This is because inside computers, for representing information in binary form by using two numbers of 0 and 1, prefixes are also represented in binary form when numbers related to computers are represented (details of binary numbers are explained later).
For examples, data like "1k calorie" is not related to computers. Therefore, it means

1k calorie = 1×10^3 calories = 1,000 calories.

On other hand, data like "1k byte" is related to computers, and therefore it means

1k byte = 1×2^{10} bytes = 1,024 bytes.

However, it is almost equally related to "$10^3\approx2^{10}$, $10^6\fallingdotseq2^{20}$, ...", and therefore it is often treated as 1k byte = 1,000 bytes these days. Memorizing this relation helps when arithmetic calculation problems are solved.

2 - 2 Radix and Radix Conversion

Decimal numbers that we normally use are numbers that are obtained by raising each digit to the power of 10 by using ten numbers from 0 through 9.

$$(362.9)_{10} = 3\times\underline{10}^{2} + 6\times\underline{10}^{1} + 2\times\underline{10}^{0} + 9\times\underline{10}^{-1}$$
$$= (362.9)_{10}$$

*10 in $(\ )_{10}$ shows that it is a decimal number.

Here, the weight (10) of the decimal digit is called radix, and numbers represented with radix *n* are referred to as *n*-adic numbers. *n*-adic numbers are represented by using *n* numbers from 0 through (*n*−1) that are carried over when *n* is reached.

(1) Binary numbers

Binary numbers are represented by using two numbers 0 and 1 that are carried over when 2 is reached.
The smallest unit of information handled inside computers is a bit, and only two numbers of 0 and 1 can be used. Therefore, computers use binary numbers that represent numerical values

with two numbers, 0 and 1.

$(101.1)_2 = 1\times\underline{2}^{2} + 0\times\underline{2}^{1} + 1\times\underline{2}^{0} + 1\times\underline{2}^{-1}$
$= (5.5)_{10}$ * 2 in $(\ )_2$ shows that it is a binary number.

Decimal numbers and their corresponding binary numbers are as shown below. Binary numbers can use only two numbers of 0 and 1 for one digit. Therefore, the number of digits quickly becomes large, compared with decimal numbers.

Decimal numbers	0	1	2	3	4	5	6	7	8	9	10	...
Binary numbers	0	1	10	11	100	101	110	111	1000	1001	1010	...

Figure 1-6 Decimal numbers and corresponding binary numbers

(2) Octal numbers

Octal numbers are numerical values that are represented by using eight numbers from 0 through 7 and that are carried over when 8 is reached.

$(317.5)_8 = 3\times\underline{8}^{2} + 1\times\underline{8}^{1} + 7\times\underline{8}^{0} + 5\times\underline{8}^{-1}$
$= (207.625)_{10}$ * 8 in $(\ )_8$ shows that it is an octal number

Decimal numbers	0	1	2	3	4	5	6	7	8	9	10	...
Octal numbers	0	1	2	3	4	5	6	7	10	11	12	...

Figure 1-7 Decimal numbers and corresponding octal numbers

(3) Hexadecimal numbers

Hexadecimal numbers are numerical values that are represented by using sixteen numbers from 0 through 9, and A(10), B(11), C(12), D(13), E(14), and F(15) and that are carried over when 16 is reached.

$(1A6.E)_{16} = 1\times\underline{16}^{2} + 10\times\underline{16}^{1} + 6\times\underline{16}^{0} + 14\times\underline{16}^{-1}$
$= (422.875)_{10}$ * 16 in $(\ )_{16}$ shows that it is a hexadecimal number.

Decimal numbers	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	...
Hexadecimal numbers	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	...

Figure 1-8 Decimal numbers and corresponding hexadecimal numbers

(4) Relation between binary numbers, octal numbers, and hexadecimal numbers

Binary numbers have an extremely large number of digits compared with decimal numbers. Although computers can handle binary numbers without problems, it is very difficult for a human being to understand when the internal status of computers is concerned. Therefore, octal numbers and hexadecimal numbers are used so that people can understand more easily.

Octal numbers use the power of 8 (= 2^3) as the weight of each digit, and therefore the information of 3 digits of binary numbers corresponds to 1 digit of octal numbers.

Binary number: $(1011100.11101)_2$

↓

$(\underline{001}\ \underline{011}\ \underline{100}\ .\ \underline{111}\ \underline{010})_2$ ... Separate after every 3 digits with radix point as reference.

↓ (0 is supplemented in the parts that fall short of 3 digits.)

Octal numbers: $(1\ \ 3\ \ 4\ .\ 7\ \ 2)_8$ ... Convert every 3 digits

Hexadecimal numbers use the power of 16(= 2^4) as the weight of each digit, and therefore the information of 4 digits of binary numbers corresponds to 1 digit of hexadecimal numbers.

Binary number: $(1011100.11101)_2$

↓

$(\underline{0101}\ \underline{1100}\ .\ \underline{11100}\ \underline{1000})_2$ ... Separate after every 4 digits with radix point as reference.

↓ (0 is supplemented in the parts that fall short of 4 digits.)

Hexadecimal number: $(5\ \ C\ .\ E\ \ 8)_{16}$ ... Convert every 4 digits

(5) Radix conversion

Radix conversion means changing the radix of a numerical value. It is also radix conversion to convert a binary number (radix 2) into an octal number (radix 8) or a hexadecimal number (radix 16).
When radix has a constant (power) relation as in the case of binary numbers, octal numbers, and hexadecimal numbers, radix conversion can be done by consolidating multiple digits. However, when radix does not have a constant (power) relation as in the case of binary numbers and decimal numbers, it is necessary to convert through computation.

1) Radix conversion from a binary number into a decimal number

Radix conversion from a binary number to a decimal number can be performed by adding the weight of each digit that is 1 in the binary number. (This is the same as adding the results that are obtained by multiplying the weight of each digit with the number (0 or 1) of each digit.)

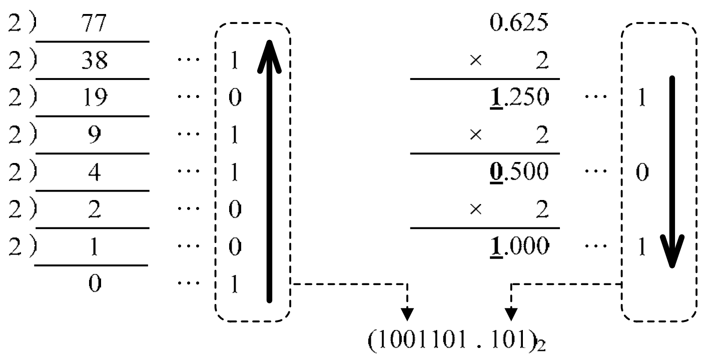
Example: Perform radix conversion of the binary number $(1001101.101)_2$ into a decimal number.

$$\begin{array}{cccccccccccc}
(1 & 0 & 0 & 1 & 1 & 0 & 1 & . & 1 & 0 & 1)_2 & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \\
2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & & 2^{-1} & 2^{-2} & 2^{-3} & \\
\downarrow & & & \downarrow & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
64 & & & +\,8 & +\,4 & & +\,1 & & +\,0.5 & & +\,0.125 & = (77.625)_{10}
\end{array}$$

2) Radix conversion from a decimal number into a binary number

Radix conversion from a decimal number into a binary number is performed separately for the integer part and the fractional part.
The integer part can be converted by repeatedly dividing by 2 until the quotient becomes 0, and then arranging the remainder of calculation results in sequence from the back.
A fractional part can be converted by continuously multiplying by 2 until the fractional part in calculation results becomes 0, and then arranging the integer part of each calculation result in sequence from the front.

Example: Perform radix conversion of the decimal number $(77.625)_{10}$ into a binary number.



A summary of radix conversion between decimal numbers and *n*-adic numbers is provided below.

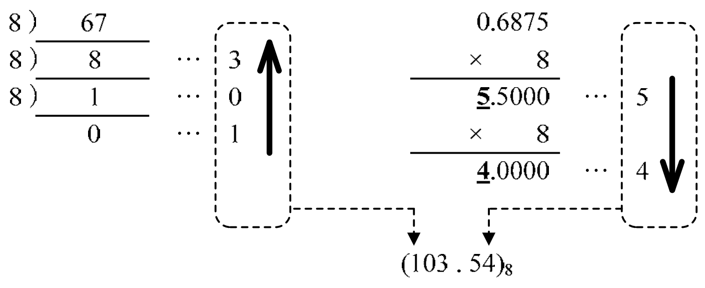
[Procedure of radix conversion from an *n*-adic number into a decimal number]

Add all results obtained by multiplying the weight (power of *n*) of each digit by the number {from 0 through (*n*−1)} of each digit.

[Procedure of radix conversion from a decimal number into an *n*-adic number]

Integer part:	Repeatedly divide by *n* until the quotient becomes 0, and then arrange the remainder of calculation results in sequence from the back.
Radix part:	Continuously multiply the radix part by *n* until the radix part in calculation results becomes 0, and then arrange the integer part of each calculation result in sequence from the front.

For example, below is an example of radix conversion of the decimal number $(67.6875)_{10}$ into an octal number.



However, when the radix conversion from decimal to octal or hexadecimal is performed, it may be easier to convert from decimal to binary, and then convert the result into octal or hexadecimal.

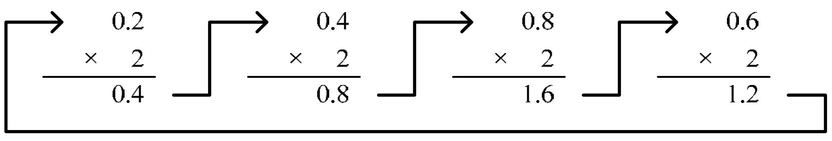
Decimal number: $(67.6875)_{10}$ = Binary number: $(1000011.1011)_2$

= $(\underline{001}\ \underline{000}\ \underline{011}\ .\ \underline{101}\ \underline{100})_2$

= Octal number: $(1\quad 0\quad 3\ .\ 5\quad 4)_8$

[Fractional number that cannot be represented by the finite number of digits]

When the fractional part of a decimal number is converted into an *n*-adic number, it gets into a loop as the fractional part of calculation results does not become 0, and it may not be converted into a finite fraction. For example, converting the decimal number $(0.2)_{10}$ into a binary number gives the following.



$(0.2)_{10} = (0.0011001100110011\cdots)_2$

In this manner, when the results of radix conversion become an infinite fraction (or recurring fraction), numerical values are handled as approximate values inside the computer.

2 - 3 Representation Form of Data

All data is represented with 0 or 1 inside computers. This representation form can be classified as follows:

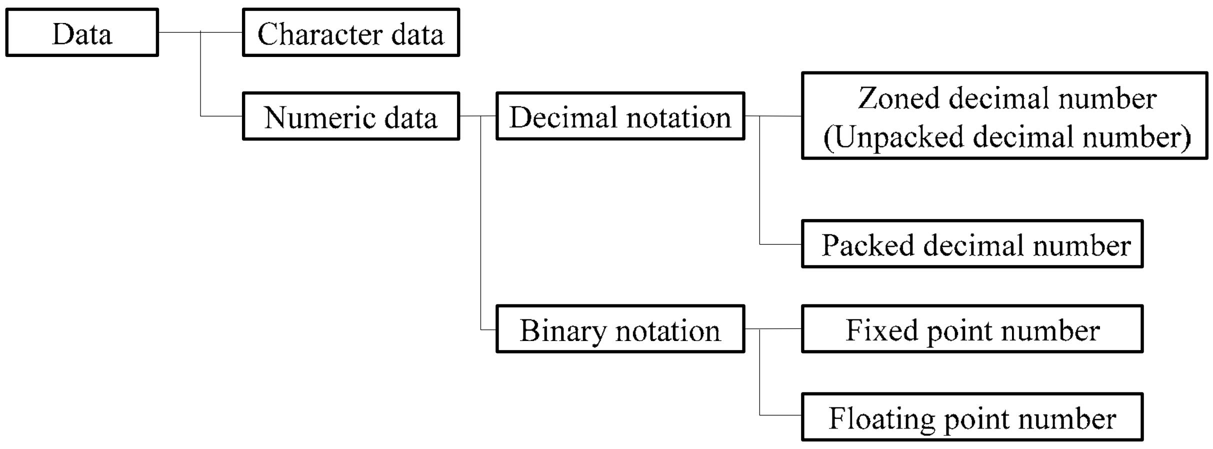


Figure 1-9 Representation Form of data

2-3-1 Character Data

Inside computers, character data is represented with combinations of 0 and 1. In computers during the early stage of their development, 1 character was linked to a bit pattern of 8 bits (1 byte). Moreover, bit patterns linked to characters were referred to as character codes, with the six main character codes as described below. When data between computers is exchanged by using different character codes, character corruption (garbled characters) may occur where characters that are different from the original data are displayed.

(1) ASCII code

ASCII code is the character code defined by ANSI (American National Standards Institute) in 1962. It is composed of 8 bits, which include code bits (7 bits) that represent an alphabetic character or a number, and a parity bit (1 bit) that detects an error. Although it is used in PCs, it does not have any definitions concerning Japanese characters (e.g., kanji, kana).

(2) ISO code

ISO code is a 7-bit character code defined by the standardization agency ISO (International Organization for Standardization) on the basis of ASCII code in 1967. This character code forms the basis of character codes used in different countries all over the world.

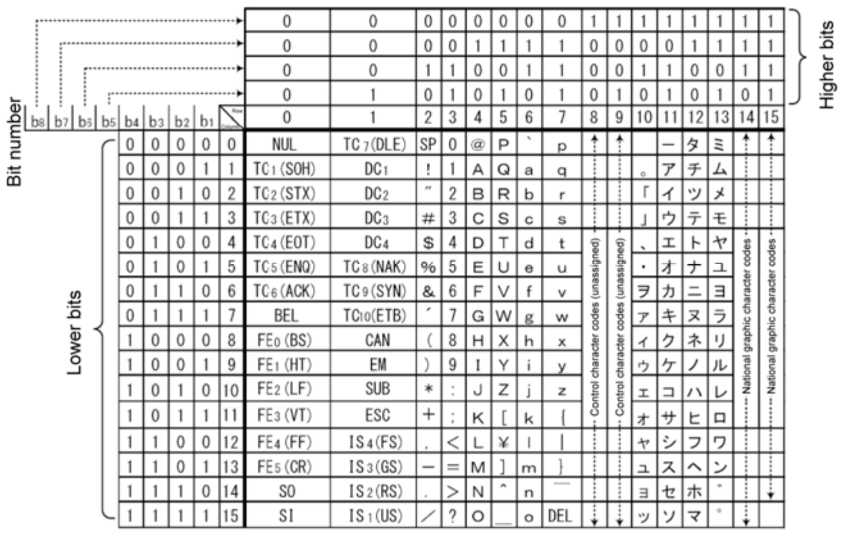
(3) JIS code

JIS code is a character code that is deliberated by JISC (Japanese Industrial Standards Committee) on the basis of the Industrial Standardization Act for representing Japanese-specific characters based on the ISO code and is defined as JIS (Japanese Industrial Standards).

1) JIS 7-bit codes / JIS 8-bit codes (JIS X 0201)
   This character code can represent half-width katakana characters. Figure 1-10 shows the table of JIS 8-bit codes for reference. Here is how to use the table. Firstly, search in the table for the character to be represented with the code. Secondly, arrange columns (higher 4 bits) and rows (lower 4 bits) from top to bottom and left to right respectively. The arranged numbers form the character code of that character.

Example: Convert “JIS” into character code.

Character		Position in table		Character code	(Hexadecimal notation)
J	...	4th column 10th row	:	0100 1010	(4A)
I	...	4th column 9th row	:	0100 1001	(49)
S	...	5th column 3rd row	:	0101 0011	(53)


Bit number
Lower bits
Higher bits
Row
Column

				b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	NUL	TC$_7$(DLE)	SP	0	@	P	`	p				ー	タ	ミ		
0	0	0	1	1	TC$_1$(SOH)	DC$_1$	!	1	A	Q	a	q			。	ア	チ	ム		
0	0	1	0	2	TC$_2$(STX)	DC$_2$	”	2	B	R	b	r			「	イ	ツ	メ		
0	0	1	1	3	TC$_3$(ETX)	DC$_3$	#	3	C	S	c	s			」	ウ	テ	モ		
0	1	0	0	4	TC$_4$(EOT)	DC$_4$	$	4	D	T	d	t	Control character codes (unassigned)	Control character codes (unassigned)	、	エ	ト	ヤ	National graphic character codes	National graphic character codes
0	1	0	1	5	TC$_5$(ENQ)	TC$_8$(NAK)	%	5	E	U	e	u			・	オ	ナ	ユ		
0	1	1	0	6	TC$_6$(ACK)	TC$_9$(SYN)	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
0	1	1	1	7	BEL	TC$_{10}$(ETB)	´	7	G	W	g	w			ァ	キ	ヌ	ラ		
1	0	0	0	8	FE$_0$(BS)	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
1	0	0	1	9	FE$_1$(HT)	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
1	0	1	0	10	FE$_2$(LF)	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
1	0	1	1	11	FE$_3$(VT)	ESC	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
1	1	0	0	12	FE$_4$(FF)	IS$_4$(FS)	,	<	L	¥	l	|			ャ	シ	フ	ワ		
1	1	0	1	13	FE$_5$(CR)	IS$_3$(GS)	−	=	M	]	m	}			ュ	ス	ヘ	ン		
1	1	1	0	14	SO	IS$_2$(RS)	.	>	N	^	n	‾			ョ	セ	ホ	゛		
1	1	1	1	15	SI	IS$_1$(US)	／	?	O	_	o	DEL			ッ	ソ	マ	゜		

Figure 1-10　JIS 8-bit codes (JIS X 0201)

2) JIS kanji code (JIS X 0208)

This is a character code for representing 1 hiragana or kanji character with 2 bytes (16 bits).

3) Shift JIS code

This code system is formed by expanding JIS kanji code, and it is a character code that enables mixing of 1-byte characters and 2-byte characters without using a special switching code.

(4) EBCDIC (Extended Binary Coded Decimal Interchange Code)

EBCDIC is an 8-bit character code developed by the IBM Corporation of the United States. This code was originally developed by IBM for its own computers. However, 3rd generation (1960s) computers were mostly IBM computers, and therefore this code became industry standard (de facto standard) in the area of large computers.

(5) Unicode

Unicode is the character code that was developed and proposed by U.S.-based companies like Apple, IBM, and Microsoft as a 2-byte universal uniform code for smooth exchange of computer data. It supports characters of several countries such as alphabets, kanji and hiragana/katakana, Hangul characters, and Arabic letters. This was standardized by ISO as an international standard, and at present, UCS-2 (2 bytes) and UCS-4 (4 bytes) are defined.

(6) EUC (Extended Unix Code)

EUC (Extended Unix Code) is the character code that was defined by the AT&T Corporation for internationalization support of UNIX (an OS (Operating System), which is a software program for controlling computers). An alphanumeric character is represented with 1 byte, and a kanji or kana character is represented with 2 bytes. In this case, kanji characters are codes where the hexadecimal number “80” is added to JIS kanji code. Therefore, in the kanji part, the value of the most significant bit is “1”, and it is possible to differentiate single byte alphanumeric characters and kanji characters.

Moreover, although it is not a character code, zoned decimal numbers are also classified as a code that represents character data (Details of zoned decimal numbers are explained in the next sub-subsection entitled “Numeric data”).

2-3-2 Numeric Data

In broad terms, there are two methods for numeric representation inside computers. The first method is to represent numerical values with binary numbers, which is suitable for performing calculations inside computers. However, binary numbers are not easy for humans to use, and therefore there exits another method where decimal numbers are adopted to facilitate understanding for humans.

(1) Decimal notation

This is a numeric representation method of introducing the way of thinking about decimal numbers that are easy-to-understand for humans. In specific terms, BCD code (Binary-Coded Decimal code) where each digit of a decimal number is converted into a 4-bit binary number is used.

Example: Represent $(2741)_{10}$ with binary-coded decimal code.

Decimal numbers	:	2	7	4	1
		↓	↓	↓	↓
Binary-coded decimal code	:	0010	0111	0100	0001

1) Zoned decimal number

This is a format that represents 1 digit of a decimal number with 1 byte. A binary-coded decimal code is stored in the lower 4 bits of 1 byte corresponding to each digit, while zoned bits are stored in the higher 4 bits. However, a sign bit indicating a sign is stored in the higher 4 bits of the lowest digit.

Zoned bits and sign bits differ depending on the character code used in computers. When EBCDIC is used, "1111 (F)" is stored in the zoned bit, while "1100 (C)" is stored as a sign bit if it is positive (+) and "1101 (D)" is stored as a sign bit if it is negative (−). On the other hand, when ASCII code or JIS code is used, "0011 (3)" is stored in the zoned bit. In most of the cases, the sign bit uses the same value as in the case of EBCDIC (it differs by manufacturer because there are no uniform definitions).

Example: Represent $(+672)_{10}$ and $(-1905)_{10}$ with zoned decimal numbers (EBCDIC).

1111	0110	1111	0111	1100	0010
	6		7	+	2

1111	0001	1111	1001	1111	0000	1101	0101
	1		9		0	−	5

2) Packed decimal number

This is a format that represents 2 digits of a decimal number with 1 byte. Each digit of a decimal number is represented with a binary-coded decimal code and this is the same as the zoned decimal number, however, it differs because the sign is shown with the lowest 4 bits, and when it falls short of a byte unit, 0 is inserted to make it a byte unit.

Example: Represent $(+672)_{10}$ and $(-1905)_{10}$ with packed decimal numbers.

0110	0111	0010	1100
6	7	2	+

0000	0001	1001	0000	0101	1101
0	1	9	0	5	−

If we compare zoned decimal numbers and packed decimal numbers, by packing after the omission of the zoned bits (higher 4 bits) of zoned decimal numbers, the packed decimal numbers can represent more information (digits) with fewer bytes. Moreover, it is called unpacking to represent packed decimal numbers by using zoned decimal numbers, and therefore the zoned decimal numbers are also referred to as unpacked decimal numbers.

While zoned decimal numbers cannot be used in calculations, packed decimal numbers can be used in calculations. However, when decimal numbers from an input unit are entered, or when decimal numbers are sent to an output unit, it is preferable to use zoned decimal numbers because the unit (1-digit numerical value) of input/output and the unit (1 byte) of information correspond to each other. Therefore, conversion between zoned decimal numbers and packed decimal numbers is performed inside computers.

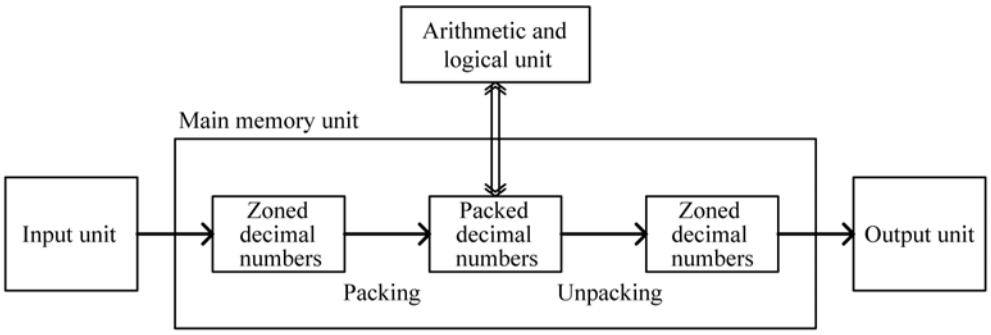


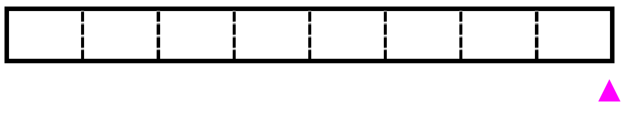
Figure1-11 Conversion between zoned decimal numbers and packed decimal numbers inside computers

(2) Binary notation

Forms for representing numerical values as binary numbers that are easy to handle inside computers include fixed point numbers where the number of digits of integer part and fractional part are decided in advance, and floating point numbers where the position of radix point is changed according to the numerical value to be represented.

1) Fixed point numbers

In this form of representation, numerical values are handled by fixing the radix point at a specific position. Generally, it is mostly used for handling integer numbers by fixing the position of the radix point in the least significant bit.



In this form of representation, after numerical values are converted into binary numbers, they are used without modification according to the position of the radix point.

Example: Represent $(89)_{10}$ as an 8 bit fixed point number.

$(89)_{10}=(1011001)_2$

$\downarrow$

$\boxed{0 \mid 1 \mid 0 \mid 1 \mid 1 \mid 0 \mid 0 \mid 1}$

In fixed point numbers, methods of representing negative numbers (representation forms of negative numbers) are important. The two main methods are described below.

A: Signed absolute value notation

In this method, the most significant 1 bit is used as a sign bit, and 0 is stored in it if the numerical value is positive, while 1 is stored in it if the numerical value is negative. In the remaining bits, the binary bit string is stored as it is.

Example: Represent $(+89)_{10}$ and $(-89)_{10}$ with an 8-bit fixed point number (signed absolute value).

$(89)_{10} = (1011001)_2$

$(+89)_{10}$: $\boxed{0}\boxed{1 \mid 0 \mid 1 \mid 1 \mid 0 \mid 0 \mid 1}$

$(-89)_{10}$: $\boxed{1}\boxed{1 \mid 0 \mid 1 \mid 1 \mid 0 \mid 0 \mid 1}$

B: Complement notation

In this method, when a negative number is represented, the complement of the positive number is used.

[Complement]

By using a certain number as the reference value, the shortfall is shown with respect to this reference value.

In *n*-adic number, there is a complement of *n* and a complement of *n*−1.

- Complement of *n*−1 ... Reference value is the maximum value that has the same number of digits.
- Complement of *n* ... Reference value is the minimum value that has one more additional digit.

The complement can be determined by subtracting from the reference value the number for which the complement is to be determined.

Example 1: Determine the complement of decimal number $(614)_{10}$

- 9's complement

$$\begin{array}{r} 999 \\ -\ 614 \\ \hline 385 \end{array}$$

- 10's complement

$$\begin{array}{r} 1000 \\ -\ 614 \\ \hline 386 \end{array}$$

Example 2: Determine the complement of binary number $(01101101)_2$

- 1's complement

$$\begin{array}{r} 11111111 \\ -\ 01101101 \\ \hline 10010010 \end{array}$$

- 2's complement

$$\begin{array}{r} 100000000 \\ -\ 01101101 \\ \hline 10010011 \end{array}$$

Example: Represent $(+89)_{10}$ and $(-89)_{10}$ with an 8-bit fixed point number (2's complement).

$(89)_{10} = (1011001)_2 \rightarrow$

$$\begin{array}{r} 100000000 \\ -\ \ 01011001 \\ \hline 10100111 \end{array}$$

$(+89)_{10}$: $\boxed{0}\boxed{1\,|\,0\,|\,1\,|\,1\,|\,0\,|\,0\,|\,1}$

$(-89)_{10}$: $\boxed{1}\boxed{0\,|\,1\,|\,0\,|\,0\,|\,1\,|\,1\,|\,1}$

Generally, 2's complement notation is often used in fixed point numbers for the following reasons.

1. **Subtraction can be substituted for addition.**

When signed absolute values are used, addition or subtraction must be appropriately selected according to the first bit of each numerical value to be calculated. However, when 2's complement is used, calculation can be performed without any selection. By using this, subtraction can be represented with addition.

Example: Calculate "$(15)_{10} - (10)_{10}$" as "$(15)_{10} + (-10)_{10}$".

1) Represent $(-10)_{10}$ with 2's complement.

$$(10)_{10} = (1010)_2 \quad \rightarrow \quad \begin{array}{r} 100000000 \\ -\quad 00001010 \\ \hline 11110110 \end{array}$$

$(-10)_{10}$:

1	1	1	1	0	1	1	0

2) Add $(15)_{10}$ to $(-10)_{10}$ represented as 2's complement.

$(15)_{10} = (1111)_2$

$(+15)_{10}$

0	0	0	0	1	1	1	1

$$\begin{array}{rl} 00001111 & \ldots (+15)_{10} \\ +\ 11110110 & \ldots (-10)_{10} \\ \hline 1\ 00000101 & \ldots (00000101)_2 = (+5)_{10} \end{array}$$

$\uparrow$

* Discard the carry-over bit.

By using this concept, a feature for subtraction is not required if computer has the feature for addition and the feature for calculating complement. Moreover, if multiplication is done with iterations of addition, and division is done with iterations of subtractions, four basic arithmetic operations (addition, subtraction, multiplication, division) can be done with addition only.

2. **A wide range of numerical values can be represented.**

When signed absolute values are used, two types of 0s, namely (+0) and (−0), occur. However, there is only one type of 0 when 2's complement is used. Since the information amount of identical numbers of bits is the same, 2's complement enables the representation of a wider range of numerical values.

Signed absolute value	
01111111	+127
01111110	+126
:	:
00000000	+0
10000000	−0
:	
11111110	−126
11111111	−127

2's complement	
01111111	+127
01111110	+126
:	:
00000000	0
11111111	−1
:	
10000001	−127
10000000	−128

[Range of numerical values that can be represented with *n* bits (2's complement)]

From -2^{n-1} through $+2^{n-1}-1$

2) Floating point numbers

If the number of bits that can be used for representing numerical value is decided, the range of numerical values that can be represented with fixed point numbers is limited. Therefore, when a very large numerical value is handled, the number of bits should also be increased accordingly. However, in reality, the number of bits used for recording data cannot be increased infinitely. Therefore, floating point numbers are used to represent extremely large numbers or extremely small numbers below the radix point.

Before the explanation of the representation of numbers inside computers, the mechanism of floating point numbers is explained by using decimal numbers. The case that there is the numerical value below is considered here.

+456,000,000,000

For the ease of explanation, when 1 unit is used for recording a sign or one number, 13 units are required for recording this numerical value as it is.

+	4	5	6	0	0	0	0	0	0	0	0	0

Here, if we convert the numerical value and then record only required information in the following manner, we can represent this numerical value with a total of 6 units.

$$+456,000,000,000$$
$$= +0.456\times10^{12}$$

$= (-1)^0 \times 0.456 \times 10^{12}$ * $(-1)^0 = +1$, $(-1)^1 = -1$

Sign: Sign to show whether the numerical value is positive or negative	0
Fraction (mantissa): Numerical value to be placed after the radix point	4 5 6
Exponent: Power of 10 to be multiplied by fraction (mantissa)	1 2

This is the concept of floating point numbers. Using this concept, from extremely large numerical values to extremely small numerical values can be represented with the same number of digits.
"10" in power of 10 used in representing exponent is called radix. Since a decimal number was used in this explanation, "10" was used as it is easy to understand in terms of moving the radix point. However, since binary numbers are used in computers, either "2" or "16" is used.

[Basic form of floating point numbers]

$(-1)^{\text{Sign}} \times \text{Fraction (mantissa)} \times \text{Radix}^{\text{Exponent}}$

In reality, the form of representation of floating point numbers used inside computers differs depending on the model. Here, we explain the single-precision floating point number (32-bit format) and double-precision floating point number (64-bit format) standardized as IEEE 754 format. A double-precision floating point number can handle a wider range of numerical values with higher precision.

[Single-precision floating point number (32-bit format)]

Sign (1 bit)	Exponent (8 bits)	Fraction (mantissa) (23 bits)

[Double-precision floating point number (64-bit format)]

Sign (1 bit)	Exponent (11 bits)	Fraction (mantissa) (52 bits)

There are several forms of representation of information in each part of the floating point number format. Main forms of representation of each part are as follows:

A: Sign (S)

This is used for representing the sign of a numerical value. It is 0 for positive values, and 1 for negative values.

B: Exponent (E)

This is used for representing exponent with respect to radix. The two main representation methods below are used.

- 2's complement:

  In this method, exponent is recorded as a binary number, and 2's complement is used if it is negative (−).
- Excess method:

  This method records a value (i.e., biased value) that is obtained by adding a certain value to the exponent. The value (i.e., bias value) to be added is decided according to the number of bits of exponent. Below is an example of the excess method used with single-precision floating point numbers.

Method name	Intended format (Typical example)	Format of exponent		Bias value
		Number of bits	Information amount	
Excess 127	IEEE 754 format	8 bits	256 types	127
Excess 64	IBM format	7 bits	128 types	64

C: Fraction (mantissa) (M)

This is used for representing a numerical value after the radix point. In order to represent the part after the radix point, it is common to perform normalization. In this case, the integer part is either set to 0 or 1 (storing after the omission of 1 from the integer part).

- When the integer part is set to 0: 0.101101 → Store "101101" in fraction (mantissa)
- When the integer part is set to 1: 1.011010 → Store "011010" in fraction (mantissa)

[Normalization]

This operation is for maintaining the precision of numerical values by increasing digits that can be used in fraction (mantissa). In most cases, this is done by reducing extra 0s after the radix point.

For example, in the case of the decimal number $(0.000123456789)_{10}$ represented by using a 7-digit fraction (mantissa), if the 7-digit fraction is registered in fraction (mantissa) as is, the following result is obtained.

0	0	0	1	2	3	4

In contrast, if the 7-digit fraction is registered in fraction (mantissa) after normalization, the following result is obtained.

Normalized form: $0.123456789 \times 10^{-3}$

1	2	3	4	5	6	7

In other words, more digits can be represented if normalization is performed (the number of significant digits increases). Therefore, the precision of numerical values becomes high.

Example: Express the decimal number $(1234.625)_{10}$ with the single-precision floating point number (32 bits) of IEEE 754 format shown below.

Sign (1 bit): Show a positive value with 0, and a negative value with 1.

Exponent (8 bits): Show with excess 127 where radix is 2.

Fraction (23 bits): Show with normalized expression where integer part is 1.

1) Convert the decimal number $(1234.625)_{10}$ into a binary number.

$(1234.625)_{10} = (10011010010.101)_2$

2) Normalize the binary number determined in 1).

Normalized form: $+(1.0011010010101)_2 \times 2^{10}$

3) Show exponent as a binary number (8 bits) of excess 127.

Power of 10 $\rightarrow$ 10 + 127 = 137 $\rightarrow$ $(10001001)_2$

4) Describe according to the form of representation.

0	10001001	00110100101010000000000

Sign (1 bit): 0

Exponent (8 bits): 10001001

Fraction (mantissa) (23 bits): 00110100101010000000000

2-3-3 Error

Error refers to the difference between the actual value and the value represented inside the computer. When numerical values inside the computer are handled, we must pay attention to error.

For example, when $(0.1)_{10}$ is converted into a binary number, it is represented as follows:

$$(0.1)_{10} = (0.00011001100110011...)_2$$

The conversion result becomes a recurring fraction where “0011” is repeated an infinite number of times. However, since there is a limit for the number of bits that can be used for representing numerical values inside computers, the only option for storing in fraction (mantissa) is to store after the loop is cut in the middle. If it is 8 bits, only up to $(0.00011001)_2$ can be stored. When this binary number is converted into a decimal number, $(0.09765625)_{10}$ is obtained. In other words, this is different from the original numerical value $(0.1)_{10}$ by only $(0.00234375)_{10}$. This is error.

(1) Rounding error

Rounding error occurs when the part smaller than the least digit is rounded off, rounded up, or rounded down in order to represent the real number with the effective number of digits in the computer. When a decimal number is converted into a binary number, this kind of error occurs to represent the resulting binary number with the effective number of digits. As a measure against rounding error, there are methods such as minimizing the value of error as much as possible by changing single precision (32 bits) into double precision (64 bits).

(2) Loss of trailing digits

Loss of trailing digits is the error that occurs when an extremely small value is ignored at the time of computing two values: one absolute value is extremely large and the other is extremely small.

Example: Calculate $(0.10110011)_2 \times 2^{10} + (0.11010001)_2 \times 2^{-10}$.

$$
\begin{array}{rr}
 & 1011001100 \\
+ & 0.000000000011010001 \\
\hline
 & 1011001100.0000000000\underline{11010001} \\
= & (0.10110011)_2 \times 2^{10} \qquad \downarrow
\end{array}
$$

This value will be ignored.

In order to minimize loss of trailing digits, when multiple numerical values are added with floating point radix numbers, it is necessary to take steps such as arranging all data in the ascending order of absolute value, and adding them in sequence from the top (smaller value).

(3) Cancellation of significant digit

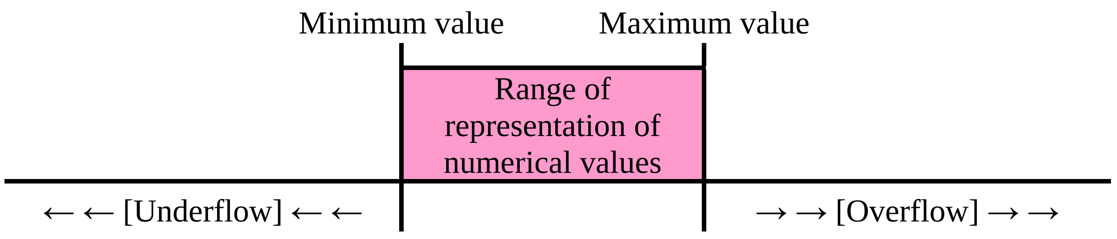
Cancellation of significant digit is the error that occurs because of decline in the number of significant digits that can be trusted as numerical values when calculation is performed between almost equal numerical values.

Example: Calculate $(0.10110011)_2 \times 2^0 - (0.10110010)_2 \times 2^0$.

$$
\begin{array}{rl}
 & 0.10110011 \\
- & 0.10110010 \\
\hline
 & 0.00000001 \\
= & (0.1\underbrace{0000000}_{\text{These 0s cannot be trusted as a numerical value.}})_2 \times 2^{-7}
\end{array}
$$

(4) Overflow, underflow

Inside a computer, the range of numerical values that can be represented is already decided, because the limited number of bits is used for representing them. Flow is the error that occurs when the calculation results exceed this range of representation. When the calculation results exceed the maximum value of range of representation, it is referred to as overflow, and when it exceeds the minimum value, it is referred to as underflow. When it is simply referred to as "flow," it mostly means overflow.



Moreover, the following two indexes are used as the concept of evaluating these errors (whether precision is high or low).

- Absolute error: This is an index that evaluates on the basis of how large the actual error is.

Absolute error = | True value − Computed value (including error) |

- Relative error: This is an index that evaluates on the basis of the proportion (ratio) of error with respect to true value.

$$\text{Relative error} = \frac{|\ \text{True value} - \text{Computed value (including error)}\ |}{|\ \text{True value}\ |}$$

2-3-4 Shift Operation

Shift operation is the operation of shifting the position of bit to left or right. Shift operation is used in computation of numerical values, and in changing the position of bits.

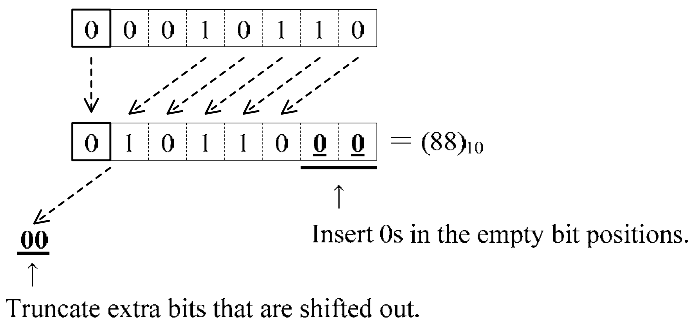
(1) Arithmetic shift

Arithmetic shift is the shift operation used when numerical values are computed. It is mainly used in fixed point numbers that represent negative values in 2's complement.

[Rules of arithmetic shift]

- Do not shift the sign bit.
- Truncate extra bits that are shifted out as a result of the shift.
- Store the following bit in the empty bit position that is created as a result of the shift.
  In the case of left shift: 0
  In the case of right shift: Same as the sign bit

Example: Shift $(22)_{10}=(00010110)_2$ arithmetically left by 2 bits.


0 0 0 1 0 1 1 0
0 1 0 1 1 0 0 0
= (88)10
00
Insert 0s in the empty bit positions.
Truncate extra bits that are shifted out.

Binary numbers have weight of power of 2 in each digit. Therefore, even for the same numeral 1, 1 as the second digit and 1 as the third digit have different meanings.

1 as second digit: $(10)_2 \rightarrow 2^1 = 2$

1 as third digit: $(100)_2 \rightarrow 2^2 = 4$

Because of this, computation rules of arithmetic shift can be summarized as follows:

[Computation rules of arithmetic shift]

- With an arithmetic shift to left by n bits, the value becomes 2^n times of the original number.
- With an arithmetic shift to right by n bits, the value becomes 2^{-n} times of the original number. (It is the value that is obtained by dividing the original number by 2^n)

(2) Logical shift

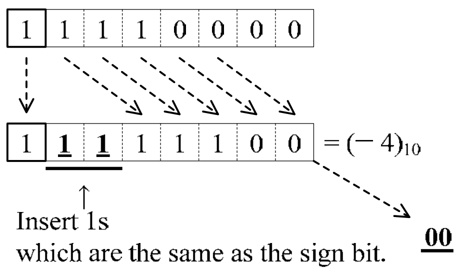
Logical shift is the shift operation used when the position of bits are changed. Its main difference from the arithmetic shift is that it does not treat the sign bit in a special manner.

[Rules of logical shift]

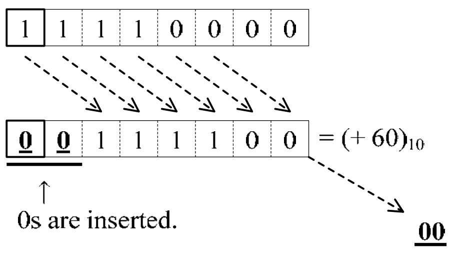
- Shift (move) the sign bit as well.
- Truncate extra bits that are shifted out as a result of the shift.
- Store 0 in the empty bit position that is created as a result of shift.

Example: Compare the results of 2-bit arithmetic shift right and 2-bit logical shift right for $(-16)_{10} = (11110000)_2$.

1) Perform 2-bit arithmetic shift right.

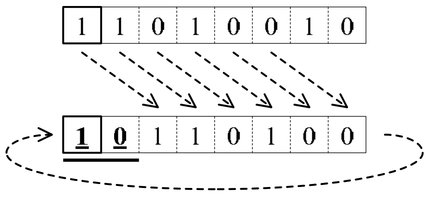


2) Perform 2-bit logical shift right.



Rotation shift (circular shift) is a type of logical shift. In rotation shift, the bits shifted out are circulated to the empty positions.

Example: Perform 2-bit rotation shift right on $(11010010)_2$.



Move the shifted-out "10" to the head.

3 Central Processing Unit and Main Memory Unit

Data processing in computers takes place in the steps of “Input → Processing → Output.” Section 1 explained the CPU (Central Processing Unit) that handles “Processing” in this series of steps is composed of a control unit and an arithmetic and logical unit. This section describes more detailed operating principles of the CPU and the main memory unit that is closely related to the CPU.

3 - 1 Configuration of CPU

In computers based on stored-program, programs (instructions) recorded in the main memory unit are read in the CPU one by one, and the control unit issues directions to each device on the basis of the contents of the instruction for processing. In order to carry out this operation, the CPU is composed of various devices and components.

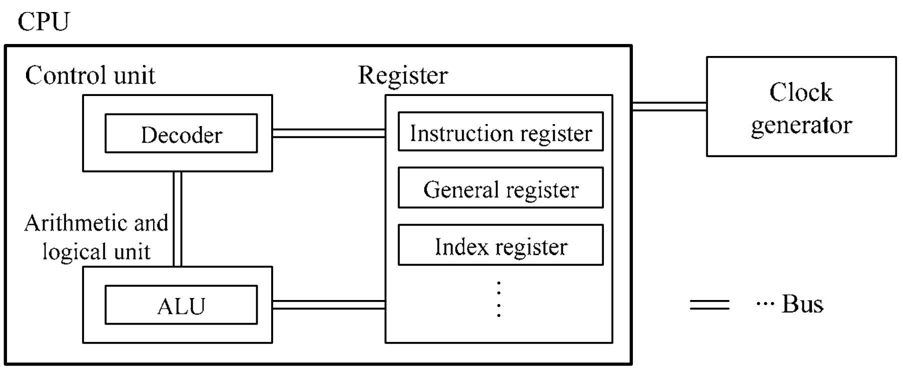


Figure 1-12 Components of CPU

(1) Control unit

Control unit is a unit that decodes the instruction to be executed and gives directions to each device. It is composed of a decoder (instruction decoder that interprets the instruction to be executed), etc.

(2) Arithmetic and logical unit

Arithmetic and logical unit is a unit that performs computations inside computers. It is composed of an ALU (Arithmetic and Logical Unit) that uses an adder for addition, a complementer for calculating complements, and such other components.

(3) Register

Register is a device used for temporarily storing various data inside the CPU. Different types of registers are available according to the type of data to be stored. The main registers (details of each register are explained later) are as follows:

- Instruction register
  It stores the instructions to be executed. It is composed of an instruction part and an address part.
- Instruction address register (program counter, program register)
  It stores the address (storing position in the main memory) of the instruction to be executed next.
- General register
  It is used for various purposes such as storing the data to be processed.
- Accumulator
  It stores the data for performing computations. It is sometimes also substituted with a general register.
- Base address register
  It stores the beginning address of a program.
- Index register
  It stores the index for address modification.
- Flag register
  It stores the information of a certain status (whether the results of computations are positive or negative, etc.). Depending on the status of this register, the next action, such as branch destination of the condition branch instruction, is decided.
- PSW (Program Status Word)
  It stores the running status of a program (value of program counter, value of flag register, etc.)

(4) Clock generator

Clock generator is a device for generating signals (clock signals) in order to synchronize and control the timing of operations between various devices inside computers. The speed of generating clock signals is shown with clock frequency, and MHz (Mega Hertz) (1 Million times in 1 second) is used as the unit. Usually, either the rising or falling edge of a signal is used as synchronization timing, and there is a technique called DDR (Double Data Rate) that uses both.

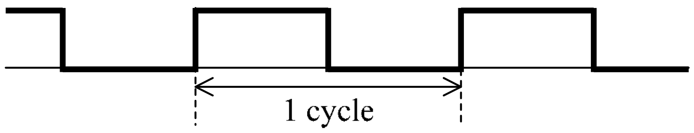


Figure 1-13 1 cycle of clock frequency

(5) Bus

Bus is the signal path for connecting various devices and registers, and transmitting data and control signals. Concerning the bus, there is a serial bus that sends data in the sequence of 1 bit each, and there is a parallel bus that simultaneously sends multiple bits. Performance of bus is decided by the access mode, such as the number of bits that can be sent in one clock signal (bus width) and clock signal, which is defined for each bus.

[Classification according to connection point]

1) Internal bus (CPU internal bus)
   This is a bus used inside the CPU.
2) External bus (CPU external bus)
   This is a bus that connects the CPU and the external devices. The clock frequency of the external bus is generally different from that of the CPU.
   - System bus: Collective term for the buses directly connected from CPU to outside
   - Memory bus: Bus that is mainly connected to the main memory unit
   - Input/output bus: Bus that is mainly connected to the input/output devices
3) Expansion bus
   This is a bus that connects the PC and the expansion cards.

[Classification according to usage]

1) Address bus: Bus for specifying the reference address to the main memory unit
2) Control bus: Bus for giving directions to each device from the control unit
3) Data bus: Bus for exchanging data

In conventional buses (Neumann architecture), the same bus was used for loading instructions and data. However, in modern-day computers, for loading instructions and data, Harvard architecture is used where independent buses are provided for loading instructions and data.
The main buses presently used in PCs are shown below. Other than in PCs, various buses are used (there are also manufacturers' specific buses).

- PCI (Peripheral Component Interconnect)
  This is an external bus specification defined by Intel Corporation, USA. PCI specification of bus width 32 bits, bus clock 33 MHz, and transfer rate of 133 megabytes/second is common.
- PCI Express (PCIe)
  This is an external bus specification defined by PCI-SIG to replace PCI. The transfer rate is 500 megabytes/second in full-duplex mode. The specification of "PCI Express x 16" where 16 transmitting lanes are contained is used as a replacement of the specification of AGP (Accelerated Graphics Port) that is used to connect graphics boards.

3 - 2 Main Memory Configuration

3-2-1 Memory Devices

Memory devices are devices that configure main memory and registers. In particular, IC-based memory devices are referred to as semiconductor memory or IC memory.

[Classification of memory devices]

1) MOS (Metal Oxide Semiconductor) type
   Although it has a high degree of integration and low power consumption, it is a semiconductor device with somewhat low operating speed. These days, CMOS (Complementary MOS) are most commonly used after improvements were made so that operating speed could be increased by transporting electrical charges by the use of free electrons and holes.
2) Bipolar type
   Although operating speed is high, this semiconductor device has a low degree of integration and power consumption is also large. A typical example of bipolar memory devices is a logic IC TTL (Transistor-Transistor Logic) composed of only bipolar transistors.

(1) RAM (Random Access Memory)

RAM is IC memory where reading and writing of data can be done freely. It is not suitable for storing data for a long time because it has a property (volatility) where data is cleared when the power is off.

- SRAM (Static RAM)

  While its operating speed is high, it is expensive and it also has high power consumption. As for its memory mechanism, it uses a flip-flop circuit that continues to retain the preceding status, and electrical charge that records information can be retained as long as power is supplied. However, its disadvantages are a low degree of integration because of complex configuration of the circuit, and storage capacity that is smaller compared with DRAM. It is mainly used in registers or other memory devices.
- DRAM (Dynamic RAM)

  While its operating speed is somewhat slow, it uses a simple circuit where electrical charge is retained by condenser or capacitor. Therefore, the degree of integration is high, and large capacity memory can be easily created at low cost. However, electrical charge that records information is lost over time, and therefore it is necessary to rewrite (refresh) the information. Examples include SDRAM (Synchronous DRAM) or DDR SDRAM (Double Data Rate SDRAM) used in main memory, and RDRAM (Rambus DRAM) that uses Rambus technology in the external bus interface.

(2) ROM (Read Only Memory)

ROM is IC memory that can be used only for reading data. It has a property (non-volatility) where data is not lost even when the power is off.

- Mask ROM

  This is a type of ROM where users cannot write data. This memory is used to store programs or data in factories, and the information is used only for the purpose of reference.
- User programmable ROM

  This is a type of ROM where users can write data. On the bases of the writing methods and restrictions on the number of rewritable times, this is classified as follows:
  - PROM (Programmable ROM):

    This is a type of ROM where user can write information only once.
  - UV-EPROM (UltraViolet-Erasable PROM):

    This is a type of ROM where data can be rewritten after information is erased by irradiating ultraviolet rays.
  - EEPROM (Electrically EPROM):

    This is a type of ROM where data can be rewritten after all or a part of information is electrically erased. It has limited life because of deterioration, and the number of rewritable times is restricted to a few tens of thousands of times to a few million times.

• Flash memory:

This is semiconductor memory where data can be rewritten after data is erased in units of blocks through electrical operations. Flash memory is a type of EEPROM. Therefore, the number of rewritable timesis limited. However, it is used for various applications as a portable and convenient storage medium.

3-2-2 Components of Main Memory

In broad terms, the main memory unit is composed of three components.

• Memory unit

This contains memory cells (storage devices) that record data.

• Read/write feature

This reads and writes data in the recording area (collection of memory cells).

• Address selection feature

This interprets the specified address and selects the recording area of data.

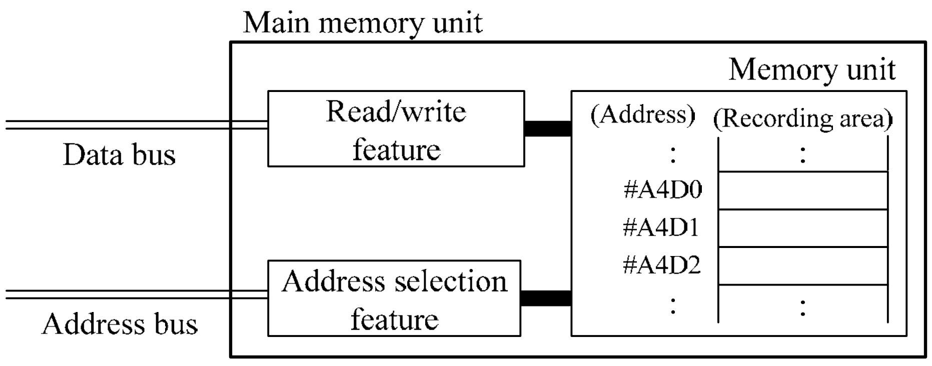


Figure 1-14 Components of main memory

Access operation to main memory is done as follows:

[Access operation to main memory]

1) Through address bus, the specified address is handed over to the address selection feature.
2) The address selection feature decodes the specified address by using the address decoder, and selects the recording area to be accessed. (address selection operation)
3) The read/write feature reads and writes data in the selected recording area. When data is read, data that is read from the recording area is passed on to the CPU via data bus. When data is written, data that is transferred from the CPU via data bus is written in the recording area.

3-2-3 Capacity Expansion of Main Memory

In the commercially available PCs, the capacity of main memory is fixed beforehand. Below are two methods of further expanding this capacity.

- Extended memory (additional memory)
  This is a type of memory added in the expansion slots provided for in desktop PCs so that devices and electronic boards (components) can be added. Examples include SIMM (Single In-line Memory Module) where DRAM memory chips are consolidated and mounted on a small board, and DIMM (Dual In-line Memory Module).
- Memory card
  This IC memory is used in capacity expansion of a notebook PC, and a typical example of this is a flash memory-based memory card. It is standardized by JEIDA (Japan Electronic Industry Development Association) and PCMCIA (Personal Computer Memory Card International Association).

3 - 3 Instruction and Addressing

3-3-1 Types and Configuration of Instructions

The CPU reads the instructions that are stored in main memory one by one, and interprets and executes the instructions. Instructions the CPU can interpret are machine language instructions, which are represented with combinations of 0 and 1, and a programming language (i.e., language used to write the programs) that is used to write such instructions is called the machine language.

Below are the main types of machine language instructions.

- Arithmetic operation instruction
  This is an instruction for performing arithmetic operations such as addition and subtraction.
- Logical operation instruction
  This is an instruction that performs logical operations such as logical product and logical sum operations.
- Transfer instruction
  This is an instruction that transfers data such as load and store.
- Comparison instruction
  This is an instruction that compares the magnitude relation between two values.
- Branch instruction
  This is an instruction that branches (jumps) the control on the basis of the value of a flag register.
- Shift instruction
  This is an instruction that performs shift operations such as arithmetic shift and logical shift.
- Input/output instruction
  This is an instruction that reads data from or write data to I/O devices.

Moreover, the configuration of machine language instructions includes instructions recorded in 1 word (recording area corresponding to 1 address) of main memory (1-word instructions) and instructions recorded consecutively in multiple words (when recorded in 2 words, it is referred to as 2-word instructions). However, even if the length of instructions is different, there is a similarity in the sense that both of them are composed of an instruction part where the instruction code specifying the process to be executed is recorded, and an address part (or operand part) for specifying the address to be processed.

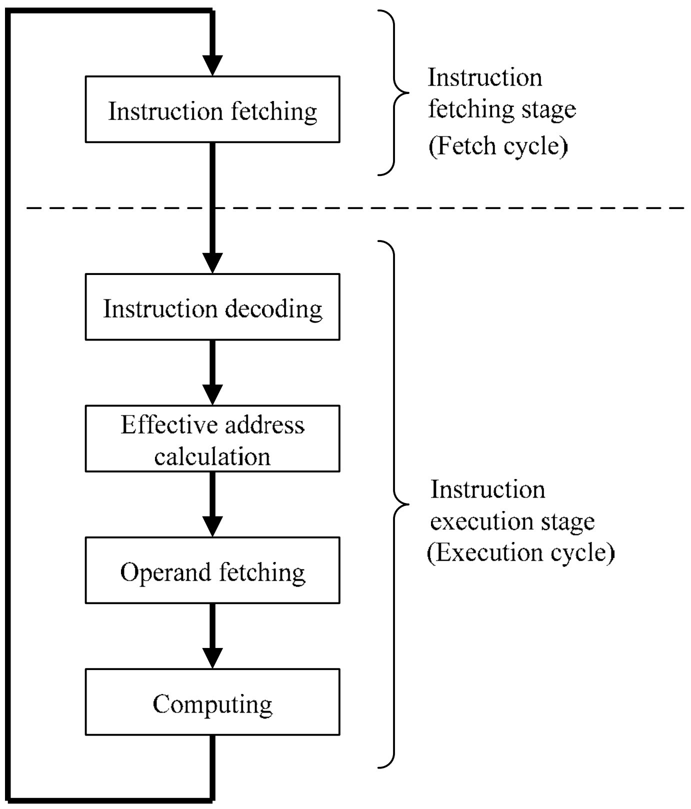
Instruction part	Address part (Operand part)

Figure 1-15 Configuration of machine language instructions

Based on the number of address parts (operand parts), they are also separately referred to as 0-address instruction, 1-address instruction, 2-address instruction, and 3-address instruction.

3-3-2 Execution Sequence of Instructions

Figure 1-16 shows the general execution sequence of instructions. Here, the process until fetching the instruction is called the instruction fetching stage (fetch cycle), and the process until decoding and executing the instruction is called the instruction execution stage (execution cycle).



Instruction fetching
Instruction fetching stage (Fetch cycle)
Instruction decoding
Effective address calculation
Operand fetching
Computing
Instruction execution stage (Execution cycle)

Figure 1-16 Execution sequence of instructions

Details of the processing sequence from "instruction fetching" until "computing" are explained in order by using the following figure.

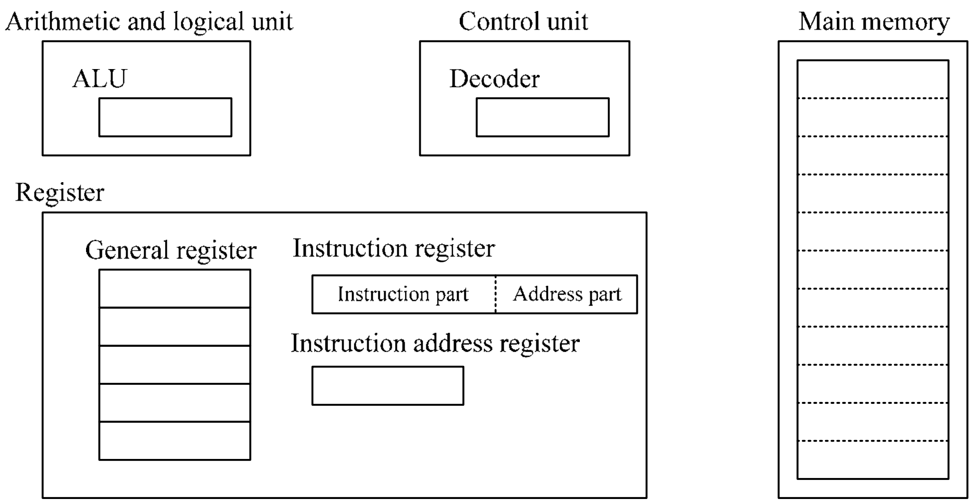


Figure 1-17 CPU and main memory

(1) Instruction fetching

In instruction fetching, on the basis of the directions given by the control unit, an instruction stored in the address shown by the instruction address register is fetched from main memory, and it is stored in the instruction register. After the instruction is fetched, instruction word length *L* (number of words where 1 instruction fetched is stored) is added to the instruction address register for fetching the next instruction.

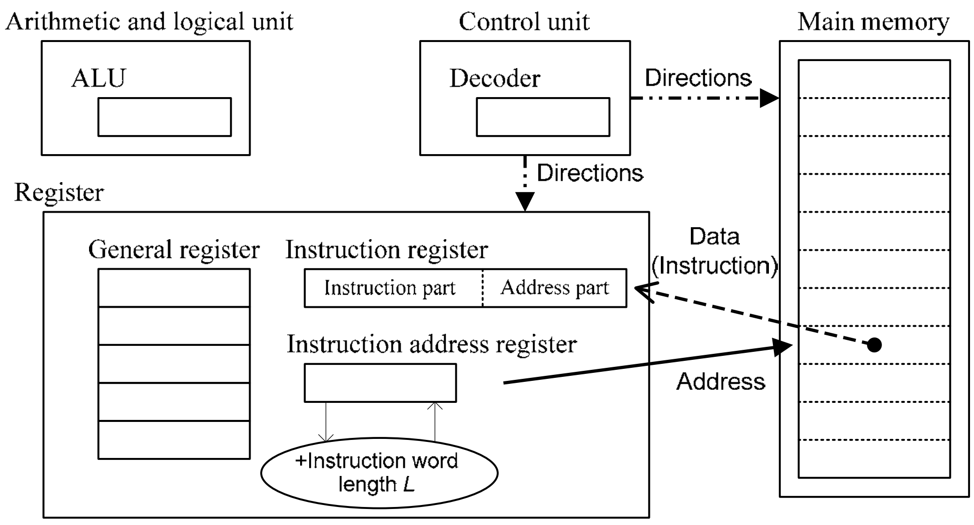


Figure 1-18 Instruction fetching

(2) Instruction decoding

In instruction decoding, the instruction part of the instruction fetched in the instruction register is decoded by the decoder (instruction decoder) of the control unit.

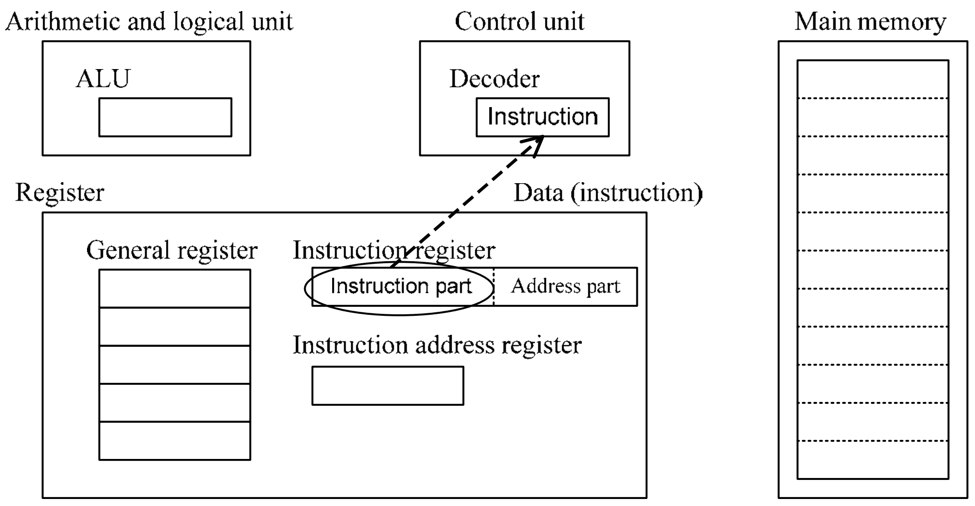


Figure 1-19 Instruction decoding

(3) Effective address calculation/operand fetching

In effective address calculation, the storage position (effective address) of data stored in main memory is determined from the address part of the instruction. (This is referred to as address modification)
In operand fetching, the effective address calculated is sent to main memory, and the value or variable (operand) to be computed such as arithmetic operation instructions is read into the general register. There may not be any operand in some cases, depending on the type of instruction, and this process may be omitted in such cases.

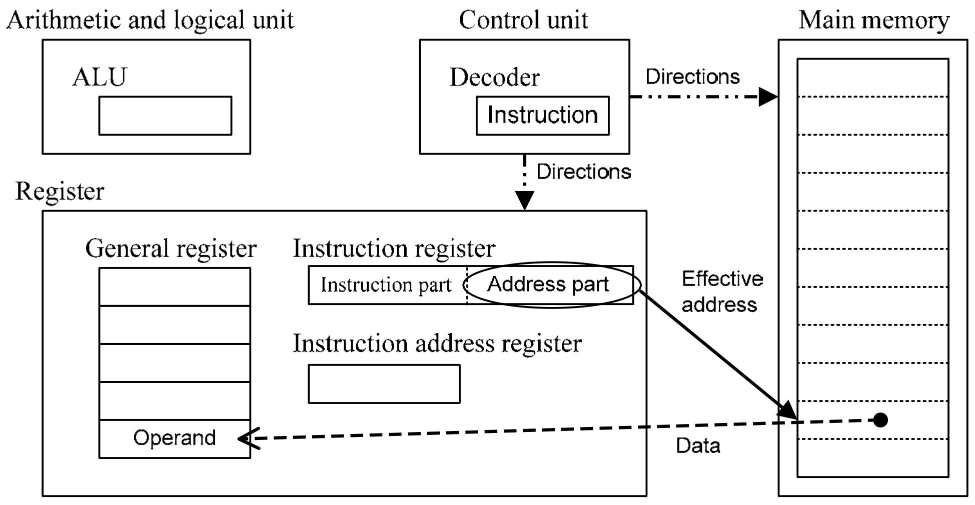


Figure 1-20 Effective address calculation and operand fetching

(4) Computation (instruction execution)

In computation (instruction execution), the arithmetic and logical unit executes the computation based on the decoded instruction. The operation result is recorded in the general register and written in main memory.

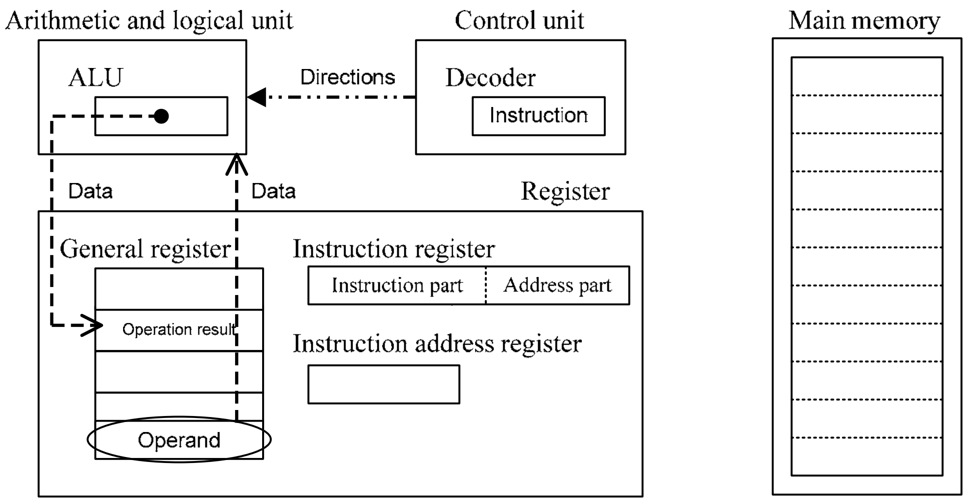


Figure 1-21 Arithmetic operation (instruction execution)

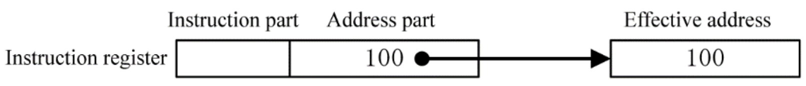
3-3-3 Addressing Mode

Addressing mode is the method of determining the effective address from the value recorded in the address part of instruction, and then fetching the operand. The addressing mode is classified into two methods: one is the method (address modification) of determining the effective address from the value of the address part of instruction and the other is the method of determining the operand from the effective address.

(1) Method of determining effective address from the value of the address part (address modification)

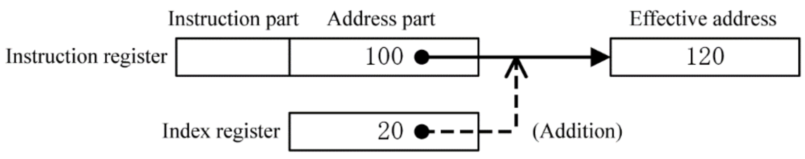
1) Absolute addressing

In this method, the value of the address part is used as the effective address as it is.



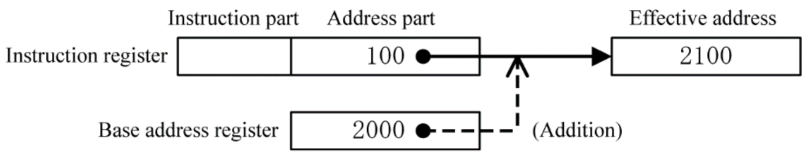
2) Index addressing

In this method, the effective address is determined by adding the value of the index register to the value of the address part. A general register is also used as the index register, and it is necessary to specify which general register to use in such cases.



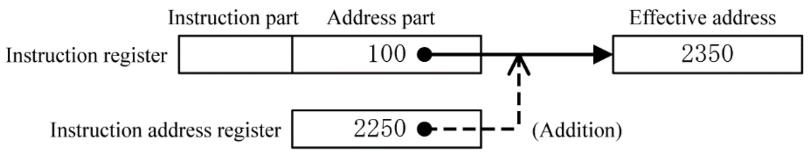
3) Base addressing

In this method, the effective address is determined by adding the value of the base address register to the value of the address part. This means the relative position from the beginning of the program.



4) Relative addressing

In this method, the effective address is determined by adding the value of the instruction address register (program counter) to the value of the address part. This means the relative position from the instruction being executed.



(2) Method of determining operand from the effective address

1) Immediate addressing

In this method, the effective address is used as it is as operand. Since it does not reference main memory, execution speed is somewhat faster than other addressing.

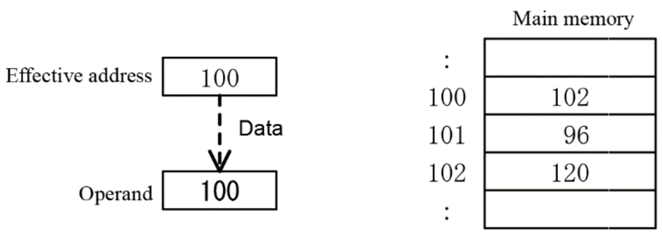


Figure 1-22 Immediate addressing

2) Direct addressing

In this method, the content (value) of main memory referenced by the effective address is used as operand. Generally, direct addressing where there is no description concerning address modification can be considered as absolute addressing, and it can be considered as direct addressing when only address modification is specified.

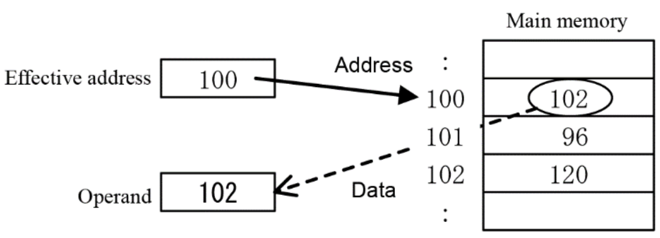


Figure 1-23 Direct addressing

3) Indirect addressing

In this method, the content (value) of main memory referenced by the effective address is used as the address of operand. Double (in the example shown in Figure 1-24, data stored in address 120 is fetched as operand), and triple indirect addressing is also possible.

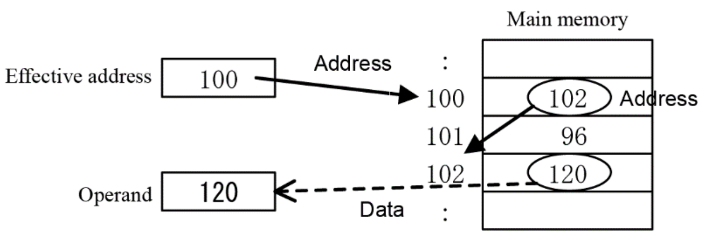


Figure 1-24 Indirect addressing

3-3-4 Interrupt

Interrupt means executing a separate process (instruction) while a series of processes (instructions) are being executed. Instead of user mode (mode where there are restrictions on the use of CPU), which is normally used, the interrupt process is executed in privilege mode (mode where there are no restrictions on the use of CPU).

[Processing sequence of processor when interrupt has occurred]

1) Switch from user mode to privilege mode.
2) Save the values of various registers (program counter, etc.).
3) Decide the starting address of the interrupt process routine (interrupt program).
4) Execute the interrupt process routine.
5) After the execution of the interrupt process routine is complete, restore the values of various registers that are saved in step 2.
6) Switch from privilege mode to user mode, and restart the interrupted process.

According to conditions that an interrupt occurs, this can be classified described below. Here, it is assumed that multiple interrupts may occur simultaneously, and therefore priority ranking is assigned to each interrupt.

1) External interrupt: This is an interrupt that occurs because of a reason not related to the process being executed.

- Timer interrupt
  This is an interrupt that occurs when the time measured in the timer, such as interval timer and watchdog timer, has exceeded the specified time (i.e., the timer has been timed-out).
- Input/output interrupt (input/output completion interrupt)
  This is an interrupt that occurs when an input/output operation has been completed.
- Machine check interrupt
  This is an interrupt that occurs because of a hardware malfunction, a power failure, or such other factor.
- Restart interrupt
  This is an interrupt that occurs when the user has pressed the external restart switch.

2) Internal interrupt: This is an interrupt that occurs because of the process being executed. This is also referred to as a trap.

- SVC (SuperVisor Call) interrupt
  This is an interrupt that occurs when supervisor (program that offers basic functions) is requested to invoke a process, such as when input/output instruction is used or storage protection exception happens.
- Program interrupt

This is an interrupt that occurs by the error of a running program (e.g., divide-by-zero or overflow)

3 - 4 Circuit Configuration of ALU

ALU (Arithmetic and Logical Unit) is a unit that performs arithmetic operations and logical operations. This unit is composed of various circuits.

3-4-1 Logic Circuit

Logic circuit is a circuit that performs logical operations for logical processing of instructions by computers. Logical operations refer to arithmetic operations that have only two truth values of True (1) and False (0).

[Main logical operations]

1) Logical product operation (AND)
   The output becomes True (1) when both input values are True (1), or else the output is False (0) (when either one is False (0)).
2) Logical sum operation (OR)
   The output becomes False (0) when both input values are False (0), or else the output is True (1) (when either one is True (1)).
3) Negation operation (NOT)
   The output becomes False (0) when the input value is True (1), and the output is True (1) when the input value is False (0).
4) Exclusive logical sum (Exclusive OR) operation (XOR or EOR)
   The output becomes False (0) when both input values are the same, and the output is True (1) when both input values are different.

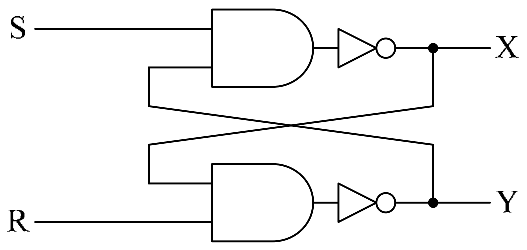
The table below summarizes the relations between input and output of the respective logical operation. This kind of table is referred to as a truth table.

Input		AND	OR	NOT	XOR
X	Y	(X AND Y)	(X OR Y)	(NOT X)	(X XOR Y)
True (1)	True (1)	True (1)	True (1)	False (0)	False (0)
True (1)	False (0)	False (0)	True (1)	False (0)	True (1)
False (0)	True (1)	False (0)	True (1)	True (1)	True (1)
False (0)	False (0)	False (0)	False (0)	True (1)	False (0)

Besides “X AND Y”, AND is also expressed as “X • Y” and “ $X \wedge Y$ ”; besides “X OR Y”, OR is also expressed as “X + Y” and “ $X \vee Y$ ”; besides “NOT X”, NOT is also expressed as “ $\overline{X}$ ” and “¬X”; and besides “X XOR Y”, XOR is also expressed as “ $X \oplus Y$ ” and “ X∀Y ” .

(1) Sequential circuit

Sequential circuit is a logic circuit where the output is decided on the basis of the input at that time and earlier status. Flip-flop circuit, which is one of the main sequential circuits, has two stable states, and it is used in storage cells of SRAM. Figure 1-25 shows the circuit diagram of a flip-flop for reference and the truth table. Circuit symbols used in Figure 1-25 are one of the methods of representing electronic circuits, and it is referred to as MIL (Military Specifications and Standards) symbols.



S	R	X	Y	Comments
1	1	X	Y	No change
1	0	0	1	Reset
0	1	1	0	Set
0	0	-	-	(Non-allowed)

Figure 1-25 Circuit diagram of flip-flop and truth table

(2) Combination circuit

Combination circuit is a logic circuit where the output is decided on the basis of the input at that time. There are various types of combination circuits, from basic circuits for implementing basic logical operations (logical product, logical sum, negation) to circuits formed by combining these basic circuits.

1) AND circuit

This circuit performs logical product operation (AND). The truth table and MIL symbol of AND circuit are as shown below.

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

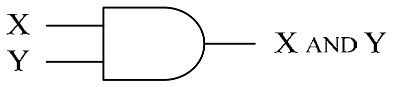


Figure 1-26 AND circuit

2) OR circuit

This circuit performs logical sum operation (OR). The truth table and MIL symbol of OR circuit are as shown below.

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

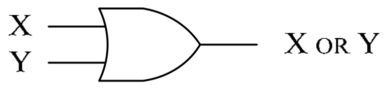


Figure 1-27 OR circuit

3) NOT circuit

This circuit performs negation operation (NOT). The truth table and MIL symbol of NOT circuit are as shown below.

X	NOT X
0	1
1	0

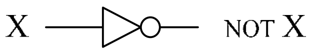


Figure 1-28 NOT circuit

4) NAND circuit

This circuit performs negative logical product (negative AND) operation (NAND). NAND operation is the arithmetic operation that negates the results of AND operation. The truth table and MIL symbol of NAND circuit are as shown below. NAND circuit is formed by combining AND circuit and NOT circuit.

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

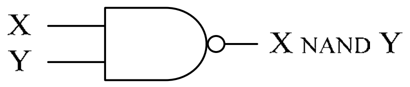


Figure 1-29 NAND circuit

5) NOR circuit

This circuit performs negative logical sum operation (NOR). NOR operation is the arithmetic operation that negates the results of OR operation. The truth table and MIL symbol of NOR circuit are as shown below. NOR circuit is formed by combining OR circuit and NOT circuit.

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

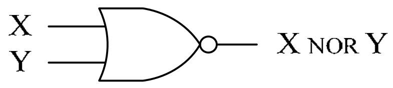


Figure 1-30 NOR circuit

6) XOR circuit

This circuit performs exclusive logical sum operation (XOR). The truth table and MIL symbol of XOR circuit are as shown below.

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

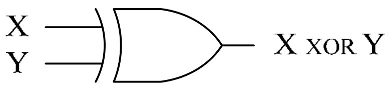


Figure 1-31 XOR circuit

XOR circuit is also formed by combining the basic circuits. However, it cannot be formed with a combination of simple circuits as in the case of NAND circuits and NOR circuits. In such cases, it is called logical design (or circuit design) to think about the combination of basic circuits for achieving the targeted results (truth values).

In logical design, logical expressions (logical functions) obtained from the truth tables are simplified using the rules of logical operations (logical laws), and the optimum combination of basic circuits is determined. ("Optimum" means that the design is performed in consideration of performance, efficiency, and cost.

[Main logical laws]

* AND operation is represented with ∧, OR operation is represented with ∨, and NOT operation is represented with ¬.

- Idempotent laws

  $A \vee A = A$

  $A \wedge A = A$
- Commutative laws

  $A \vee B = B \vee A$

  $A \wedge B = B \wedge A$
- Associative laws

  $A \vee (B \vee C) = (A \vee B) \vee C$

  $A \wedge (B \wedge C) = (A \wedge B) \wedge C$
- Distributive laws

  $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

  $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
- Absorption laws

  $A \vee (A \wedge B) = A$

  $A \wedge (A \vee B) = A$
- De Morgan's laws

  $\neg(A \vee B) = (\neg A) \wedge (\neg B)$

  $\neg(A \wedge B) = (\neg A) \vee (\neg B)$
- Others

  $A \vee 0 = A$

  $A \wedge 0 = 0$

  $A \vee 1 = 1$

  $A \wedge 1 = A$

  $A \vee (\neg A) = 1$

  $A \wedge (\neg A) = 0$

  $\neg(\neg A) = A$ (Returns to original by double negation)

[Logical design of XOR circuit]

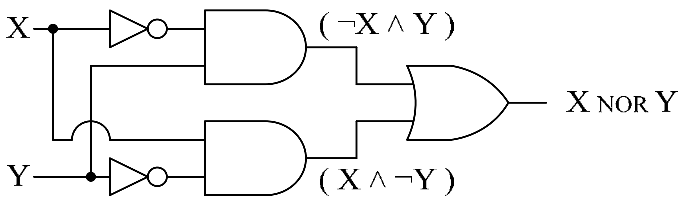
1) Focus on the parts where the output of the truth table is 1 and determine AND operation where the result is 1 on the basis of the input at that time.
   - When (X=0, Y=1): $\neg X \wedge Y$
   - When (X=1, Y=0): $X \wedge \neg Y$
2) Create a circuit where the output will be 1 when either of the AND operations you determined is 1. In order to design such a circuit, determine logical expression F

where two AND operations are combined with OR operation.

Logical expression $F = (\neg X \wedge Y) \vee (X \wedge \neg Y)$

3) Determine the circuit diagram by combining the basic circuits such that it represents logical expression F.

[Circuit diagram]



By expanding the logical expression F determined in [Logical design of XOR circuit] using logical laws, we can determine different logical circuits.

Logical expression $F = (\neg X \wedge Y) \vee (X \wedge \neg Y)$

$= ((\neg X \wedge Y) \vee X) \wedge ((\neg X \wedge Y) \vee \neg Y)$ ...Distributive law

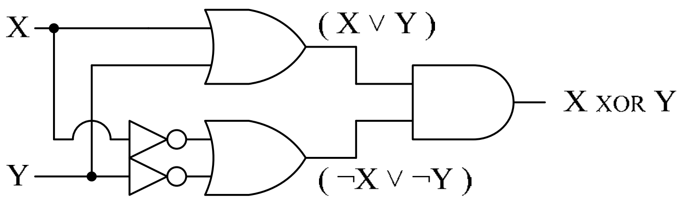
$= ((\neg X \vee X) \wedge (Y \vee X))$

$\wedge ((\neg X \vee \neg Y) \wedge (Y \vee \neg Y))$ ...Distributive law

$= (1 \wedge (Y \vee X)) \wedge ((\neg X \vee \neg Y) \wedge 1)$ ... $A \vee (\neg A) = 1$

$= (X \vee Y) \wedge (\neg X \vee \neg Y)$ ... $A \wedge 1 = A$, Commutative law

[Circuit diagram]



In the actual logical design, it is common to use NAND circuit, NOR circuit, or XOR circuit as one circuit (basic circuit) as it is.

3-4-2 Arithmetic Operation Circuit

In a computer, subtraction can be performed with addition using complement notation. Similarly, the four basic arithmetic operations can be implemented with addition only. This is because multiplication can be performed with repeated addition, and division can be performed with repeated subtraction (i.e. repeated addition). Therefore, in order to implement the four basic arithmetic operations in a computer, only an adder circuit and a complement circuit are required.

First, the adder circuit is considered here. The basis of addition in a computer is 2-bit addition that adds one bit to one bit. Here, then, a circuit to perform 2-bit addition is considered. As 2-bit addition, the following four types are possible.

```
X :       0        0        1        1
Y :   +   0    +   1    +   0    +   1
     ------   ------   ------   ------
        0 0      0 1      0 1      1 0
```

The results of the addition of two 1-bit binary numbers are shown below.

[Results of the addition of two 1-bit binary numbers]

Values to be added		Operation result	
X	Y	Carry (c)	Sum (s)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

From this table, it is clear that carry is 1 only when two entered values are both 1, this is a logical product operation (AND). It is also clear that sum is 0 when two entered values are the same and 1 when they are different, and therefore, this is an exclusive logical sum operation (XOR). Therefore, a circuit diagram with the configuration below is derived. This circuit is called a half adder.

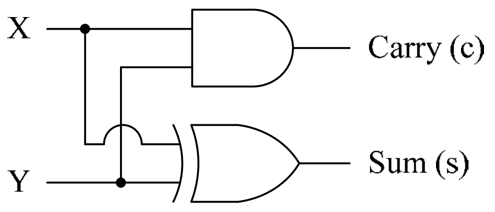


Figure 1-32 Circuit diagram of half adder

Adder circuit of a computer is configured with this half adder as the basis. However, in half adder, carry from low order is not taken into account. Therefore, arithmetic operation of numerical values represented with multiple bits cannot be performed. Because of that, another adder circuit of 3 bits including carry from low order becomes necessary.

The table below summarizes the results of arithmetic operation of 3-bit addition considering carry (c0) from low order.

[Results of arithmetic operation of 3-bit addition]

Values to be added			Operation result	
X	Y	c0	Carry (c1)	Sum (s)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

In order to derive circuit configuration from this table, we will get the following if we think about logical expression F1 for determining carry (c1) and logical expression F2 for determining sum (s).

[Logical expression F1 for determining carry]

1) With the input for which the output of truth table will become 1, determine AND operations where the result will be 1.
   - When (X=0, Y=1, c0=1): $\neg X \wedge Y \wedge c0$
   - When (X=1, Y=0, c0=1): $X \wedge \neg Y \wedge c0$
   - When (X=1, Y=1, c0=0): $X \wedge Y \wedge \neg c0$
   - When (X=1, Y=1, c0=1): $X \wedge Y \wedge c0$
2) Determine logical expression F1 by combining AND operations we determined with OR operations.

   Logical expression $F1 = (\neg X \wedge Y \wedge c0) \vee (X \wedge \neg Y \wedge c0)$
   $\vee (X \wedge Y \wedge \neg c0) \vee (X \wedge Y \wedge c0)$

[Logical expression F2 for determining sum]

1) With the input for which the output of truth table will become 1, determine AND operations where the result will be 1.
   - When (X=0, Y=0, c0=1): $\neg X \wedge \neg Y \wedge c0$
   - When (X=0, Y=1, c0=0): $\neg X \wedge Y \wedge \neg c0$
   - When (X=1, Y=0, c0=0): $X \wedge \neg Y \wedge \neg c0$
   - When (X=1, Y=1, c0=1): $X \wedge Y \wedge c0$
2) Determine logical expression F2 by combining the AND operations we determined with OR operations.

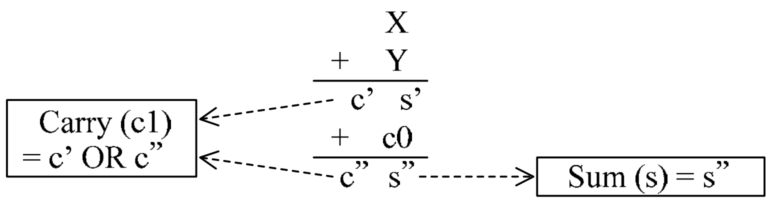
   Logical expression $F2 = (\neg X \wedge \neg Y \wedge c0) \vee (\neg X \wedge Y \wedge \neg c0)$

$$\vee (X \wedge \neg Y \wedge \neg c0) \vee (X \wedge Y \wedge c0)$$

By expanding (simplifying) these logical expressions F1 and F2 by using logical laws, we can derive the configuration of a 3-bit adder circuit.
However, even expansion of logical expression with this method is difficult, and it can be anticipated that configuration of the circuit will become complicated. Therefore, we will use half adders after 3-bit addition is divided into 2-bit addition.

[Dividing 3-bit addition into 2-bit addition]

1) Add X and Y, and determine carry (c') and sum (s').
2) Add s' and c0, and determine carry (c'') and sum (s''). s'' determined here will be sum (s) in overall arithmetic operation.
3) If there is any carry in this addition, then carry will occur for overall arithmetic operation. Therefore, carry (c1) in overall arithmetic operation is determined with OR operation of c' and c''. (Carry will never be 1 in both the operations)



Full adder shown in Figure 1-33 is the adder circuit configured on the basis of this concept. In a computer, the number of full adders equal to the number of bits of arithmetic operation is arranged to form the arithmetic operation circuit.

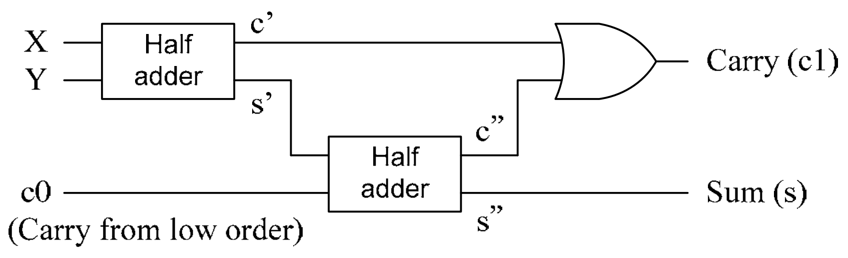


Figure 1-33 Circuit diagram of full adder

Next, let us think about the complement circuit. In order to perform arithmetic operations (four basic arithmetic operations) with only adders (half adder/full adder), we will need a circuit for determining complement (complementer) in order to perform subtraction by using addition. Inside a computer, binary arithmetic operations are performed. Therefore, it is common to use two types of complementers, namely, 1's complementer that determines 1's complement, and 2's complementer that determines 2's complement. 1's complement in binary numbers is determined by inverting the original bit sequence (0 is inverted to 1, and 1 is inverted to 0). Therefore, 1's complementer can be formed only with NOT circuit. On the other hand, there are various methods of forming 2's complementer. However, the concept of using an adder to add 1 to 1's complement determined using 1's complementer is a relatively easy method.

3 - 5 High Speed Technologies

High speed technologies are technologies that increase the operating speed (processing capacity) of computers by creating new mechanisms using each device.

3-5-1 High Speed Memory Access

Memory is the term that means a device or a medium that records data, and when simply referred to as "memory," it mostly means the main memory unit (main memory).
High speed memory access is a technology that resolves bottlenecks caused by difference in access speeds (access gap) by even slightly accelerating access to main memory (DRAM), which has slower access speed compared with the register (SRAM) of the processor.

(1) Buffer memory

Buffer memory is the medium-speed storage installed between high-speed storage and low-speed storage. Cache memory, which is a buffer memory, is the medium-speed storage composed of SRAM and installed between high-speed register and low-speed main memory.

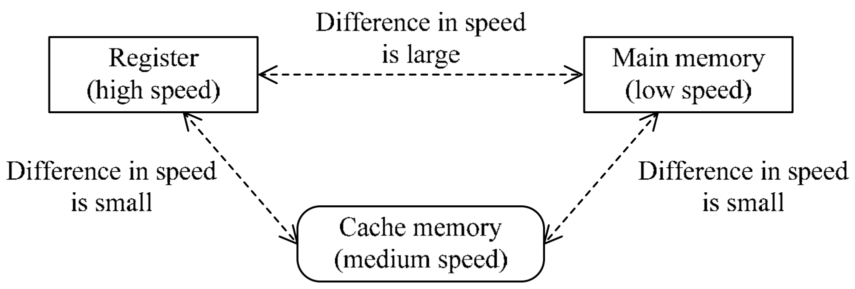

Difference in speed is large
Register (high speed)
Main memory (low speed)
Difference in speed is small
Difference in speed is small
Cache memory (medium speed)

Figure 1-34 Working of cache memory

In a computer system where cache memory is used, if the data required by the processor is available (when it is hit) in cache memory, then cache memory is accessed. On the other hand, when the data is not there (when it is mishit) in cache memory, main memory is accessed (in precise terms, on the basis of the directions of hardware or OS (software for controlling the computer), the data is loaded from main memory into the corresponding block of cache memory). In other words, average access time (effective access time) as seen from the processor is decided on the basis of the extent to which the required data is found in cache memory. Therefore, it is determined by weighted average of access time of cache memory and access time of main memory. Here, the access time is determined by considering both the probability (hit ratio) that the required data is available in cache memory and the probability (NFP (Not Found Probability)) that the required data is not available in cache memory. "Hit ratio + NFP = 1" always holds true.

Example: Calculate the average access time of the following processor. NFP (probability that the required data is not available in cache memory) is 0.1.

Access destination	Access time (nanosecond)
Cache memory	50
Main memory	500

Average access time of processor

= Access time of cache memory × hit ratio

+ Access time of main memory × NFP

= 50 nanoseconds × (1 − 0.1) + 500 nanoseconds × 0.1

= 95 nanoseconds

From this example, it is clear that the higher the hit ratio is, the shorter the average access time is because of increased access to cache memory. When Harvard architecture, which independently loads instructions and data, is used, cache memory is also separated into instruction cache and data cache. In this case, creating a program consolidating process (instruction) sections that are frequently executed will increase the hit ratio of instruction cache, which may shorten the average access time (access speed will improve).

Moreover, when cache memory is used, data inside cache memory is updated (changed). However, original data is stored in main memory, so it is necessary to reflect the data updated inside cache memory (write it in main memory). Below are the two writing methods according to the timing of reflecting data update.

[Writing methods from cache memory]

1) Write-through method

This method reflects the update in main memory whenever data is updated. While certainty is high, main memory needs to be accessed for every process, so operating speed is somewhat slow.

2) Write-back method

In this method, data is not updated in main memory while data is updated in cache memory. Data is updated in main memory only when data from cache memory is removed. While its operating speed is fast, inconsistency in the data of main memory may occur at some point in time.

In modern-day computers, there is a large difference in the speed of processor and the speed of main memory. Therefore, it is common to use multi-level cache configuration where cache memory is divided into 2 levels, namely, primary cache and secondary cache. In this case, the processor will access in the sequence of primary cache → secondary cache → main memory.

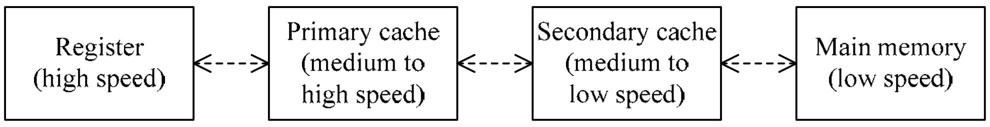


Figure 1-35 Multi-level cache configuration

Meanwhile, there is disk cache, which is another buffer memory. This buffer memory is placed between main memory and auxiliary storage (hard disk), and it may use a part of main memory, or semiconductor memory may be installed separately.

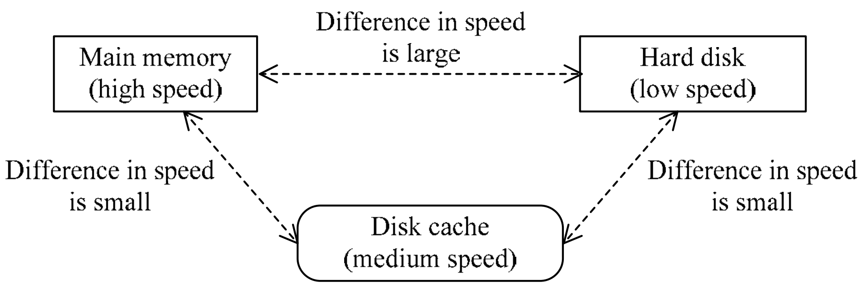


Figure 1-36 Working of disk cache

Basically, the higher the access speed of storage, the smaller the storage capacity. Therefore, by combining high-speed low-capacity storage and low-speed high-capacity storage as shown in Figure 1-37, we can configure high-speed high-capacity storage on an overall basis. This concept (concept where each storage is tiered on the basis of access speed and capacity) is referred to as memory hierarchy.

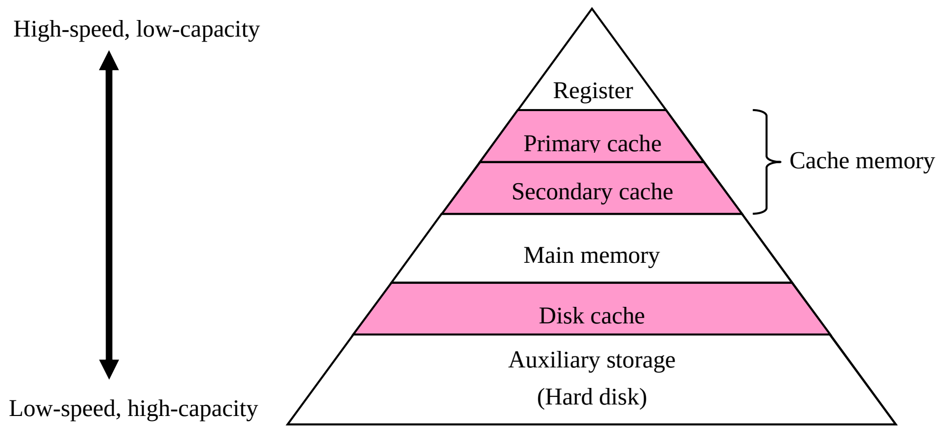


Figure 1-37 Memory hierarchy

(2) Memory interleave

Memory interleave is a method where main memory is divided into several sections (bank) that are simultaneously and independently accessed in order to improve the average access time of main memory.

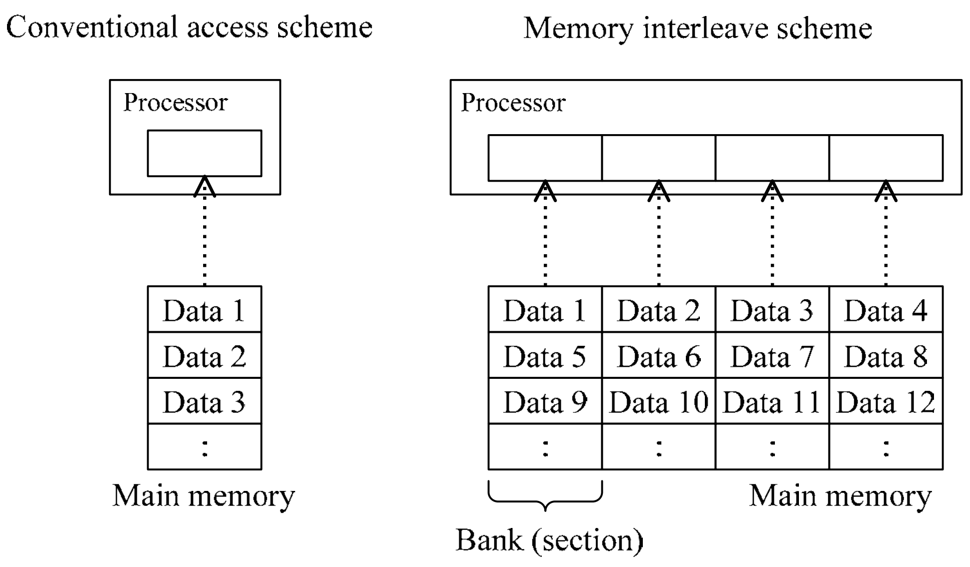


Figure 1-38 Memory interleave method

Memory interleave is a method of reading the data in advance, and it is useful for the processing of data that is located in the contiguous area of main memory. However, data may not be referenced in the sequence of storage area at all times, and the average access time may not be always proportionate to the number of banks (even if there are 4 banks, the average access time may not be 1/4).

3-5-2 Speeding-up of Processor

The speeding-up of processors is a technology for increasing the operating speed of processors. It includes various high-speed techniques starting such as a technique for enhancing the performance of the processor itself and a technique for enhancing the apparent operating speed by managing the execution procedure of instructions.

(1) Multi-core processor

In a multi-core processor, two or more processors cores are integrated on one chip (LSI). A processor with only one processor core on one chip (LSI) is referred to as a single-core processor. The multi-core processor enhances processing performance by allocating processes to multiple processor cores (if *n*-core processors are integrated, the processing performance will become approximately *n* times greater). Moreover, in comparison with mounting multiple single-core processors, the processing performance of the processor itself can be increased while power consumption is minimized. However, when multiple processor cores operate simultaneously, contention for shared resources may occur.

(2) Pipeline control

Pipeline control (or the pipeline processing) divides the execution cycle of each instruction into multiple stages, and by shifting each stage little by little and executing multiple stages separately, multiple instructions are executed simultaneously and in parallel.

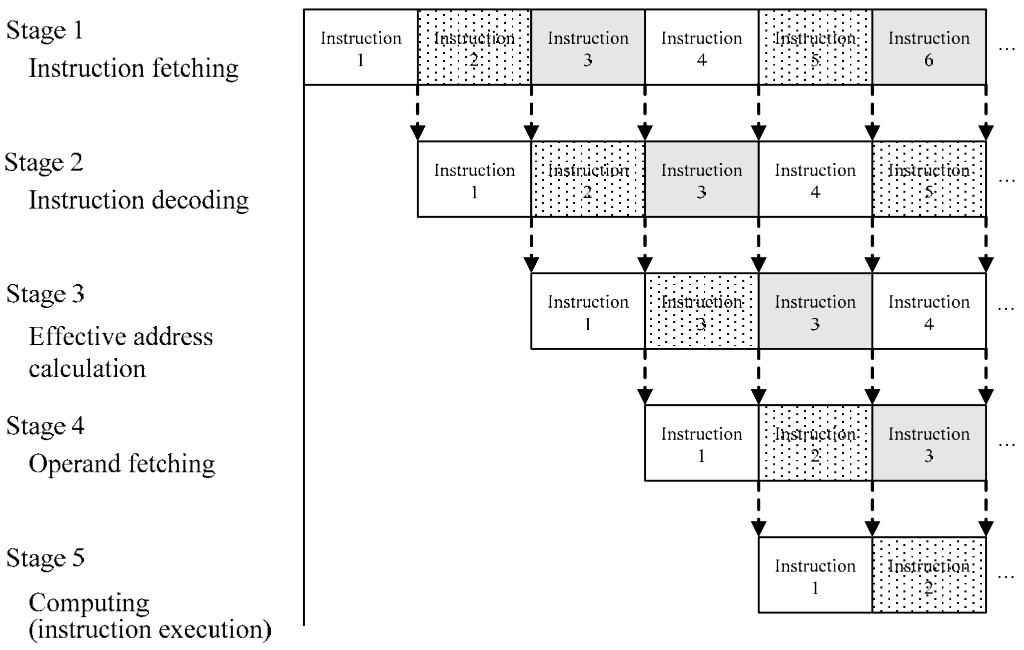


Figure 1-39 Pipeline control

Pipeline control is a method of advanced control of instructions. Therefore, efficiency will decline if there are instructions (jump instructions) that change the execution sequence of instruction. Moreover, if the execution time of each instruction is different, the waiting time may be required before starting the execution of a stage, and efficiency will decline.

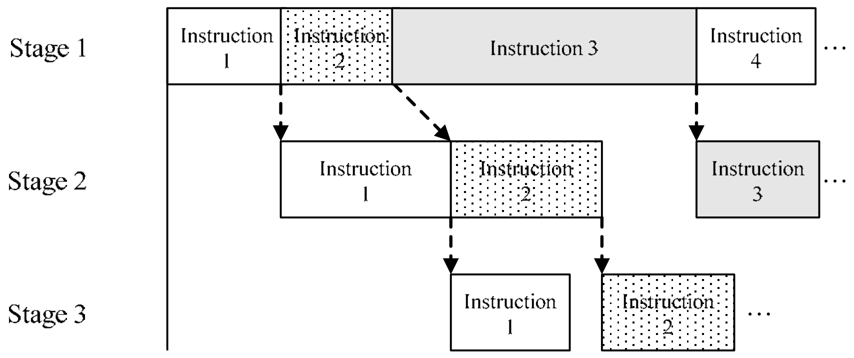


Figure 1-40 Pipeline control with poor efficiency

For efficient pipeline control, it is necessary that the execution time of all instructions be almost the same. Among the different architectures of processors (computers), RISC architecture is suitable for this concept, while CISC architecture is not very suitable.

[Processor architecture]

1) RISC (Reduced Instruction Set Computer)

Reduced instruction set computer - this is a computer where, by reducing the instruction set (collection of instructions) to the basic instructions with high frequency of use in order to fix the instruction word length, the execution time for a single instruction is shortened and unified as much as possible. Instructions are executed by hardware (wired logic).

- Wired logic control:

  Instruction is executed by passing signals through a logic circuit having a certain function. Since logic is formed by connecting signal lines for transmitting signals, it is also called hard-wired logic control.

2) CISC (Complex Instruction Set Computer)

Complex instruction set computer - this computer implements complex instructions by using software (microprograms), and therefore it can handle multifunction instructions. However, since the length and execution time of each instruction are largely different, it is not suitable for pipeline control.

- Microprogram control:

  Micro instruction refers to giving control directions to hardware such as various logic circuits and registers. This method executes instructions with a microprogram where these micro instructions are combined. It is also referred to as stored logic control.

In pipeline control, the execution cycle of instructions is divided into stages that mean “instruction fetching,” “instruction decoding,” “effective address calculation,” “operand fetching,” and “computing.” The method where this division is further broken down is referred to as super-pipeline. Super-pipeline increases the number of stages (depth of pipeline) and thereby reduces the processing time (pipeline pitch) of one stage in order to aim for performance enhancement.

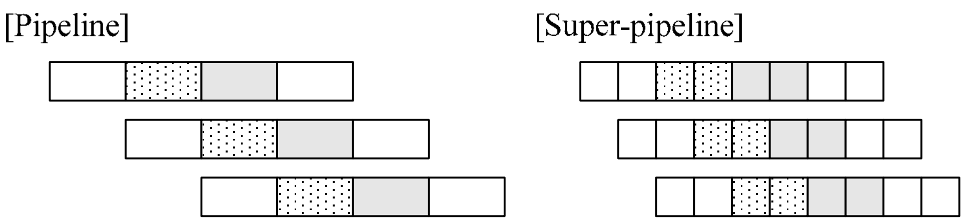


Figure 1-41　Comparison image of pipeline and super-pipeline

(3) Superscalar/VLIW (Very Long Instruction Word)

Superscalar and VLIW aim to enhance performance of the processor by simultaneously executing multiple instructions.

At the stage of executing a program, superscalar extracts multiple instructions that can be executed simultaneously, and executes them in parallel while the arithmetic unit to be used is dynamically decided. Therefore, the sequence of instructions written in a program may differ from the sequence of instructions actually executed (this kind of execution method is referred to as out-of-order execution).

On the other hand, VLIW consolidates multiple instructions that can be executed simultaneously into one instruction at the stage of generating the object program (program translated into machine language) by compiling (translating into machine language) the program, and statically allocates the arithmetic and logical units to be used. (VLIW means that multiple instructions (functions) are consolidated into one instruction, and thus the length of instruction) increases. In VLIW, instructions that can be executed simultaneously are not dynamically selected at the time of execution, so the sequence of instructions written in the object program is always same as the sequence of instructions actually executed. (This kind of execution is called in-order execution).

(4) Parallel processing

Pipeline/super-pipeline, superscalar, and VLIW aim to enhance processor performance by parallel execution of instructions in the processor. On the other hand, there is another approach of enhancing the performance of a computer by the parallelization of processors. The architecture of these parallel computers is classified into 4 types from the relation between flow of instructions and flow of data.

- SISD (Single Instruction stream Single Data stream)
  This computer processes one unit of data with one instruction. General computers that do not have parallelized processors correspond to this architecture.
- SIMD (Single Instruction stream Multiple Data stream)
  This computer processes multiple units of data with one instruction. Vector computers equipped with a vector processor (array processor) that simultaneously computes multiple data in array with one instruction correspond to this architecture.
- MISD (Multiple Instruction stream Single Data stream)
  This computer processes a single unit of data with multiple instructions. Computers that use this architecture are not used in practice (although they are closer to pipeline control, strictly speaking, they are different).
- MIMD (Multiple Instruction stream Multiple Data stream)
  This computer processes multiple units of data with multiple instructions. Multiprocessors where multiple processors are multiplexed correspond to this architecture.

4 Auxiliary Storage

Data to be processed inside the computer must be stored in main memory beforehand. However, the capacity of main memory is limited, and therefore it cannot store all data. Accordingly, data is stored in a separate storage device, and the required data is moved into main memory for processing. In such cases, auxiliary storage devices play the role of storing the data that cannot be stored in main memory. Therefore, auxiliary storage devices basically have larger capacity than main memory, and have a characteristic of non-volatility. In this section, we will learn about the types and mechanism of typical auxiliary storage devices.

4 - 1 Magnetic Disk

Magnetic disk is the most extensively used storage medium and is a typical example of an auxiliary storage medium. A magnetic disk is a disc-shaped storage medium with magnetic material coated on the surface, and it stores data by magnetic variation. Data is stored in concentric circles, with one concentric circle referred to as a track, and the unit of dividing the track in a constant length is called a sector. In modern magnetic disks, it is common to use the constant density recording method (zone sector method), where the number of sectors is increased in the tracks on the outer side.

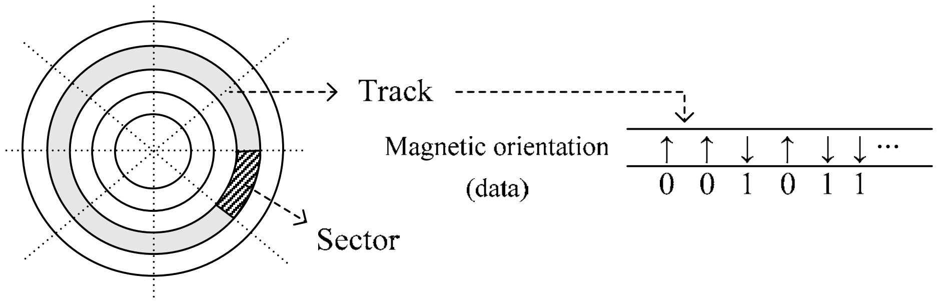


Figure 1-42 Magnetic disk

Magnetic head reads and changes the magnetic orientation of the magnetic disk. An electromagnet is embedded in the magnetic head, and it is fixed to the tip of an access arm to read and write data from and to the target track.

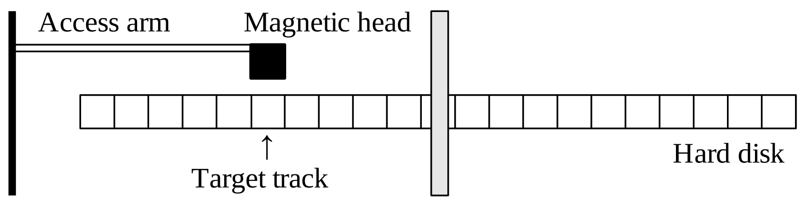


Figure 1-43 Magnetic disk and magnetic head

There exist the following types of magnetic disk-based auxiliary storage devices (auxiliary storage medium).

1) HDD (Hard Disk Drive)

This auxiliary storage device is mounted as a standard device in almost all computers. In addition to the built-in HDD provided inside the computer, an external HDD is also available that can be used by connecting it to the computer. By stacking multiple magnetic discs inside a firmly sealed case, it is possible to record a large amount of data. In order to read data from multiple magnetic discs that are stacked, multiple access arms having magnetic heads are mounted.

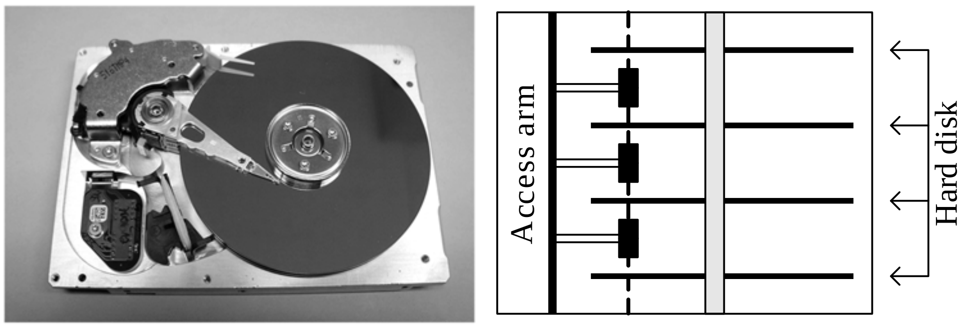


Figure 1-44 Structure of hard disk drive

Reading and writing of data from and to a hard disk drive is performed by moving the access arm horizontally and positioning the magnetic head on the target track of the hard disk. At that time, as evident from Figure 1-44, multiple tracks (tracks on a dashed line) can be read and written at the same position of the access arm (this set of tracks is called a cylinder). Reading and writing of data is performed for each cylinder separately.

In the hard disk drive, since the hard disk is continuously rotating at high speed, the reading and writing of information is performed without letting the magnetic head touch the hard disk (non-contact method). Therefore, it suffers from the disadvantage that it is sensitive to vibrations.

2) FD (Floppy Disk)

This is an auxiliary storage medium that can be carried by covering one magnetic disc (storage media that can be carried by releasing it from the device is called removable media). Although it is very inexpensive, it suffers from the disadvantages that storage capacity is small and it is easily affected by magnetism. Therefore, it is hardly used these days. In the FDD (Floppy Disk Drive) used to read and write data on the floppy disk, a magnetic head comes in contact with the disk in order to read and write data (contact method). Therefore, the disk has a tendency to deteriorate the stored contents because of wear and abrasion.

3) ZIP

This auxiliary storage medium was developed by Iomega of the United States. Its storage capacity is larger than that of a floppy disk, and it is portable. However, it is hardly used these days.

4-1-1 Storage Capacity of Magnetic Disk Drive

Storage capacity is the amount of data that can be recorded in a storage device. The configuration of general magnetic disk drives is "magnetic disk drive > cylinder > track > sector."

Therefore, the storage capacity of a magnetic disk drive can be determined in sequence from smaller to larger configuration units.

[Method of calculating storage capacity of magnetic disk drive]

1) From the amount of data that can be recorded in one sector and the number of sectors in one track, calculate the storage capacity of one track.
2) From the amount of data that can be recorded in one track and the number of tracks in one cylinder, calculate the storage capacity of one cylinder.
3) From the amount of data that can be recorded in one cylinder and the number of cylinders in the magnetic disk drive, calculate the storage capacity of the magnetic disk drive.

When the number of magnetic discs is small (e.g., FD), storage capacity is also calculated using the number of magnetic discs instead of cylinders.

Example: Calculate the storage capacity of a magnetic disk drive having the following specifications.

Number of cylinders/drive	300 cylinders
Number of tracks/cylinder	20 tracks
Number of sectors/track	30 sectors
Number of bytes/sector	500 bytes

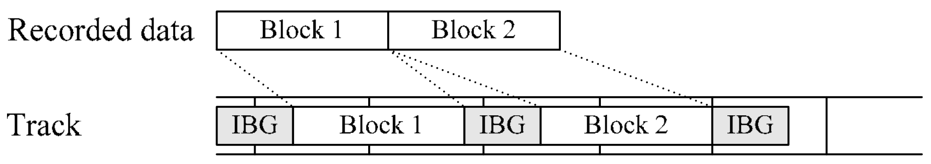
1) Storage capacity of one track
   = Storage capacity of one sector × Number of sectors in one track
   = 500 bytes/sector × 30 sectors/track
   = 15,000 bytes/track
2) Storage capacity of one cylinder
   = Storage capacity of one track × Number of tracks in one cylinder
   = 15,000 bytes/track × 20 tracks/cylinder
   = 300,000 bytes/cylinder
3) Storage capacity of magnetic disk drive
   = Storage capacity of one cylinder × Number of cylinders in magnetic disk drive
   = 300,000 bytes/cylinder × 300 cylinders/drive
   = 90,000,000 bytes/drive ... 90Mbytes/drive

4-1-2 Recording Area in Magnetic Disk Drive

Recording area refers to the area required for recording data in the storage device. It is important how to handle the unit of data (recording method of data) for calculating the size of recording area required for recording data in the magnetic disk drive. In general magnetic disk drives, multiple records, which are units having logical meanings, are consolidated into a block, and a block is treated as the unit of input and output (normal records are also referred to as logical records, and blocks are also referred to as physical records). Consolidating multiple records into a block is referred to as blocking, and the number of records consolidated into a block is referred to as the blocking factor.

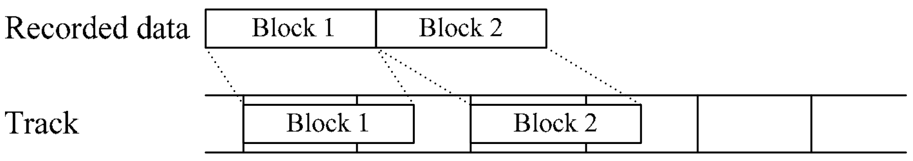
[Methods of recording data]

1) Variable method
   This method reads and writes data in units of blocks. When data is recorded on a track, an IBG (InterBlock Gap) is inserted between two blocks so as to separate them.



2) Sector method

This method reads and writes data in units of sectors. Blocks are separated by not recording multiple blocks in the same sector. A concept called cluster is also used, where multiple sectors are consolidated and treated as the unit of input/output.



In the sector method, which is commonly used in modern PCs (hard disk drive), the size of the recording area required for recording data is calculated as follows:

[Method of calculating size of recording area of magnetic disk drive]

1) From the size of one block, calculate the number of sectors required for recording the data.
2) Calculate the number of blocks of all data to be recorded.
3) Calculate the number of sectors required for recording all data, and calculate the required number of tracks and the required number of cylinders in that sequence.

Example: For a magnetic disk drive that has the following specifications, calculate the number of cylinders required for recording 10,000 records with the blocking factor of 10 where one record is of 300 bytes. Data is recorded in the magnetic disk by using the sector method.

Number of cylinders/drive	300 cylinders
Number of tracks/cylinder	45 tracks
Number of sectors/track	30 sectors
Number of bytes/sector	512 bytes

1) Size of one block
   = Size of one record × blocking factor
   = 300 bytes/record × 10 records/block
   = 3,000 bytes/block
2) Number of sectors required for recording one block
   = Size of one block ÷ Storage capacity of one sector
   = 3,000 bytes/block ÷ 512 bytes/sector
   =5.85 ··· sectors/block
   * Since 5 sectors do not suffice, it is rounded up to 6 sectors/block
3) Number of blocks to be recorded

= Total number of records ÷ blocking factor

= 10,000 records ÷ 10 records/block

= 1,000 blocks

4) Number of sectors required for recording all records

= Number of sectors required for recording one block

× Number of blocks to be recorded

= 6 sectors/block × 1,000 blocks

= 6,000 sectors

5) Number of tracks required for recording all records

= Number of sectors required for recording all records

÷ Number of sectors in one track

= 6,000 sectors ÷ 30 sectors/track

= 200 tracks

6) Number of cylinders required for recording all records

= Number of tracks required for recording all records

÷ Number of tracks in one cylinder

= 200 tracks ÷ 45 tracks/cylinder

= 4.44 ··· cylinders

* Since 4 cylinders do not suffice, it is rounded up to 5 cylinders.

* In the calculations that are performed in steps 2) and 6), it is necessary to pay attention to the meaning of rounding-up the calculation results.

4-1-3 Average Access Time of Magnetic Disk Drive

Access refers to reading and writing data, and the time from placing an access request until the result is returned is called access time (in contrast, after an access request is received, the time taken until the drive is ready to accept the next access request is called cycle time).
Reading (or writing) of data in the magnetic disk drive is carried out in the sequence described below.

1) Seek

Move the magnetic head (access arm) up to the target track (cylinder). This time is called seek time.

2) Search

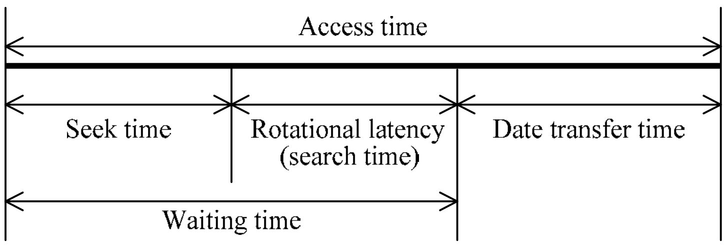
Wait until the head of the data to be read (or the position where data writing will start) comes right under the magnetic head. The time required for this process is called rotational latency (search time).

3) Data transfer

The magnetic head will read (or write) the data. The time required for this process is called data transfer time.

Therefore, access time is a summation of these times.

[Access time of magnetic disk drive]



Access time = seek time + rotational latency + data transfer time

On the other hand, average access time is the average value of access time for reading (or writing) data of one unit of input/output. In this case, since the amount of data of one unit of input/output is the same, data transfer time is the same on each occasion. However, seek time and rotational latency are different on each occasion. Therefore, their respective average values (average seek time, average rotational latency) are used.

[Average access time of magnetic disk drive]

Average access time

= average seek time + average rotational latency + data transfer time

In order to calculate average access time, the values to be used as average seek time, average rotational latency, and data transfer time are shown below. Here, when average latency is used, it can be calculated as “average seek time + average rotational latency.”

1) Average seek time

This is the time required for the magnetic head to move up to the target track (cylinder). Therefore, seek time differs on the basis of both the position of the magnetic head when the access is started and the position of the target track. Usually, therefore, the average seek time stated in the specifications of magnetic disk is used.

2) Average rotational latency (average search time)

Since the magnetic disk rotates at a constant speed, in some cases there may not be any latency when the head of data is positioned directly underneath from the beginning (minimum rotational latency). On the other hand, it may be necessary to wait until the disk completes one rotation when the head of data has just passed (maximum rotational latency). Therefore, the average of minimum rotational latency and maximum rotation latency, which is 1/2 of the time required for the magnetic disk to complete one rotation, is used.

$$\text{Average rotational latency} = \frac{\text{Time required for magnetic disk to complete one rotation}}{2}$$

3) Data transfer time

Data is read (or written) when the target data passes underneath the magnetic head. In other words, when the magnetic disk completes one rotation, the track will complete one rotation underneath the magnetic head, and therefore data recorded in one track will be read (or written). Therefore, the data transfer rate of the magnetic disk can be calculated as follows:

$$\text{Data transfer rate} = \frac{\text{Storage capacity of one track}}{\text{Time required for magnetic disk to complete one rotation}}$$

Also, calculate data transfer time by dividing the amount of data (amount of transferred data) of one unit of input/output (data to be read or written) by the data transfer rate.

Average rotational latency and data transfer time are closely related to the rotational speed of the disk. Therefore, increasing the rotational speed of the disk shortens the time. On the other hand, the proportion of seek time, which is the physical time required for access arm movement, tends to increase in access time. Therefore, shortening the seek time by recording the data in consecutive area and thereby reducing the amount of access arm movement is useful for shortening the average access time. Moreover, when data is recorded in a piecemeal manner (fragmentation), the average access time will become longer. Therefore, it is necessary to perform defragmentation in order to eliminate fragmentation.

Example: Calculate the average access time for reading a data block of 2,000 bytes recorded in a magnetic disk drive having the following specifications.

Rotational speed	3,000 rotations/minute
Bytes/track	10,000 bytes
Average seek time	30 milliseconds

1) Time required for magnetic disk to complete one rotation

3,000 rotations : 1 minute = 1 rotation : x milliseconds

3,000 rotations : $1 \times 60 \times 10^3$milliseconds=1 rotation : x milliseconds

3,000 rotations $\times$ x milliseconds = $1 \times 60 \times 10^3$ milliseconds $\times$ 1 rotation

x milliseconds = $1 \times 60 \times 10^3$ milliseconds $\times$ 1 rotation ÷ 3,000 rotations

= 20 milliseconds

2) Average rotational latency

= Time required for magnetic disk to complete one rotation ÷ 2

= 20 milliseconds ÷ 2

= 10 milliseconds

3) Data transfer rate

= Storage capacity of one track

÷ Time required for magnetic disk to complete one rotation

= 10,000 bytes/track ÷ 20 milliseconds/rotation (track)

= 500 bytes/millisecond

4) Data transfer time

= Amount of transferred data ÷ Data transfer rate

= 2,000 bytes ÷ 500 bytes/millisecond

= 4 milliseconds

5) Average access time

= Average seek time + Average rotational latency + Data transfer time

= 30 milliseconds + 10 milliseconds + 4 milliseconds

= 44 milliseconds

Finally, note that the average access time calculated here is simply a value for a magnetic disk drive as an independent device. For example, when disk cache is used, the average access time of the magnetic disk drive (auxiliary storage device) as seen from the processor (main memory unit) will not be the average access time calculated here. In such cases, with consideration of the hit ratio (or NFP), it is necessary to determine average access time (effective access time) as the weighted average of access time of disk cache and the access time of the magnetic disk drive.

4 - 2 Optical Disc

Optical disc is a commonly used auxiliary storage medium (removable media) these days. While optical disc drives use different mechanisms depending on the device, all of them are high-capacity auxiliary storage devices where the reading and writing of data is performed by means of a laser beam. Optical disc drives can be of the slot-in type where a disc is inserted into the slot as it is with an audio device, or of a tray type where a tray for mounting the disc comes out of the drive.

Although the structure of optical discs is basically the same as that of magnetic disks, the main difference is that tracks where data is recorded are spiral-shaped. Data is read by an optical head that detects the difference of reflected light after projecting a laser beam. However, seek time is longer because the optical head is heavier than a magnetic head. As a result, average access time of optical disc drives tends to be longer than that of magnetic disk drives.
Optical discs include CD (Compact Disc), with standard storage capacity of about 700Mbytes, and DVD (Digital Versatile Disc), which has large storage capacity (standard storage capacity: a single-sided, single-layer disc holding 4.7Gbytes, and a single-sided, double-layer disc holding 8.5Gbytes) by multiple layers and short wavelength of laser beams.

(1) Read-only

Read-only optical discs record information by creating a small hole called a pit on the surface of discs in the factory and by changing the reflection of the laser beam. They are inexpensive because bulk production is possible by pressing a master disc. However, the user cannot write new data on these discs. Typical examples of read-only optical discs are CD-ROM (CD-Read Only Memory) and DVD-ROM.
CD-ROM was originally developed for music. Therefore, music data and digital data can be mixed on the same disc. Moreover, when CD-ROM or DVD-ROM is produced in bulk, aluminum vapor-deposited layers (reflection layers) that are sandwiched between acrylic are pressed after adhesive agent is mounted with. When this adhesive agent peels off, resin (aluminum deposition) covering the surface will deteriorate (oxidize), and data can no longer be read. Therefore, the life span of CD-ROM and DVD-ROM is said to be about 20-30 years.

(2) Write-once

In write-once optical discs, while data written at the factory cannot be deleted or modified, data can be added (recorded) by creating a section where data can be recorded. Data can be written by making a pit, by burning an organic coloring matter that is coated on the disc with a strong laser beam. However, the burned organic coloring matter cannot be restored to its original state, and therefore, it is not possible to rewrite even the added data (WO (Write Once)). Typical examples are CD-R (CD-Recordable), DVD-R, and DVD+R.

(3) Rewritable

In rewritable optical discs, data can be rewritten. As for the recording mechanism, heat and light are used to control the temperature of the recording surface of the disc, and data is recorded with metal phase-change technique that creates two states, namely, a crystalline state and a non-crystalline state. Since data can be added/deleted/modified, this is an auxiliary storage medium that can be handled in a similar way to a magnetic disk. Typical examples are CD-RW (CD-ReWritable), DVD-RW, DVD-RAM (DVD-Random Access Memory), and DVD+RW. Write-once and rewritable optical discs can use the same playback and recording drives (CD-R/RW drive, DVD-R/RW drive). However, because the laser beam reflection rate differs depending on different data recording methods, data may sometimes not be read correctly in read-only playback drives (CD-ROM drive, DVD-ROM drive).

[Methods of writing data in optical discs]

- DAO (Disk-At-Once)
  In this method, all data is written in one go on the entire disc.
- TAO (Track At Once)
  In this method, data is written separately for each track.
- SAO (Session At Once)
  In this method, data is written separately for each session (a session is a recording unit composed of lead-in that indicates the start, data, and lead-out that indicates the end).
- Multi-session
  This method allows multiple sessions to be recorded on one disc. This recording method can be performed with a device that supports TAO or SAO.
- Packet writing
  Just like a magnetic disk, this method writes data in small units (packets). This method is expanding the range of applications of optical discs.

(4) Blu-ray disc

Blu-ray disc has drawn a lot of attention in recent years. It uses a blue-violet laser beam which has a much shorter wavelength than the laser beams that is used in conventional optical discs, which achieves high capacity. Most Blu-ray drives that is used for playback and recording of Blu-ray discs also support CD, DVD, etc. At present, this format is mainly used for entertainment purposes. However, people have started using Blu-ray discs as a form of auxiliary storage media for PCs.

4 - 3 Semiconductor Memory

Semiconductor memory (IC memory) is a storage device that uses IC such as RAM, ROM, and flash memory. In addition to being used as storage devices that constitute main memory and registers, it is also used as the recording medium of auxiliary storage devices.

(1) SSD (Solid State Drive)

SSD (Solid State Drive) is a flash memory-based auxiliary storage device that is expected to replace the HDD (Hard Disk Drive). In SSD, information is electrically read and written using flash memory. Therefore, unlike a hard disk drive, it does not require any drives (e.g., disk, access arm). For that reason, it has shorter access time because of no physical location seeking or rotational latency. Moreover, it has faster direct access (random access) that reads and writes data irrespective of the recording sequence of data. Additionally, it can be compact and lightweight, consumes less power, and is also resilient to vibrations or impact. However, there are limitations to the number of times data can be written (durability), and it suffers from the disadvantage that it is not suitable for applications where data needs to be rewritten frequently.

(2) RAM disk

RAM disk is an auxiliary storage device that uses a part of main memory (DRAM) as external storage by means of virtualization. In the early days, external RAM disks were used for expansion by combining a large amount of inexpensive RAM. However, it is rarely used these days. While high speed access is possible, data is cleared (volatility) once the power is off. Therefore, it is not suitable for storing data. (In order to save data, it is necessary to write it out to a hard disk.) In addition, data (file) that is recorded on a RAM disk is often called a RAM file, and the mechanism of data management is sometimes called a RAM file system. (A file which has the ".ram" extension and is used in a certain software product is also known as a RAM file. However, it needs to be noted that this is different from the RAM file mentioned above).

(3) USB memory/SD card (SD memory card)

USB memory is an auxiliary storage device that can easily be connected to and removed from the computer by equipping flash memory with a USB connector. On the other hand, SD card (SD memory card) is an auxiliary storage medium where flash memory is put on a chip, and it is used by insertion into the SD slot of computers or other devices such as digital cameras, smartphones, and cell phones. Both USB memory and SD card are flash memory based removable media, and they are suitable for carrying data. There are unified standards of the USB plugs and receptacles for USB memory. However, note that SD cards and SD slots may be using different standards depending on the manufacturer.

4 - 4 Other Auxiliary Storage Media and Drives

(1) MO (Magneto-Optical disc)

MO (Magneto-Optical disc) is an auxiliary storage medium that magnetizes the magnetic material that is coated on the recording surface of disc and records data by giving the direction of the magnetic force. By increasing the temperature of the magnetic material with laser beams and eliminating magnetic force for clearing data, it is possible to record data by magnetizing it once again. In comparison with magnetic storage media, it has the excellent properties of conservation and environmental resistance, and it is suitable for long-term storage of data, and even high density recording is possible. (There are standards such as a storage capacity of 230Mbytes or 640Mbytes). However, a laser beam is used for reading data, so data can be read with one rotation of disc. But, both a laser beam and magnetic head are used for writing data, so the disk needs to rotate one time each for clearing and writing data (total of two rotations). Therefore, access speed becomes slow. Moreover, the standard of 640Mbytes is a higher-level standard than that of 230Mbytes. Therefore, while the drive for 640Mbytes can read and write a 230-Mbyte disc, the logic format (format of recording data) is not unified, so it may not be possible to exchange data between different models. For these reasons, it is hardly used these days.

(2) Magnetic tape unit

Magnetic tape unit is an auxiliary storage device that records data by magnetizing the tape (magnetic tape) where magnetic material is coated. The access speed of a magnetic tape unit is extremely slow, and only sequential access is possible where data is consecutively read and written in sequence. (A device that consecutively records like a magnetic tape unit is called a streamer.) However, cost per unit information is extremely inexpensive in the case of magnetic tapes, and this auxiliary storage medium can record a large amount of data. Therefore, it is used for backing-up the hard disk. In the past, people mostly used open reel-type magnetic tapes, which were not suitable for hand-carrying because of their large size and weight. These days, however, people mostly use cartridge-type portable DDS (Digital Data Storage) on the basis of DAT (Digital Audio Tape) technology that was developed for music.

(3) PC card

PC card is a card-type auxiliary storage medium used in notebook PCs and other devices, and its standard was defined by PCMCIA (Personal Computer Memory Card International Association), an industrial body of the United States. PC cards for several applications have been available on the market, such as memory cards used for capacity expansion (increasing memory) of main memory, FAX modems, and adapters for LAN connection. But currently USB memory and SD cards prevail, and PC cards are rarely used.

5 Input/Output Unit

In order to use a computer, it is necessary to enter data in the computer, and send out the result of processing this data inside the computer. Here, a device that is used for entering data is called an input unit, and a device that is used for output is called an output unit.

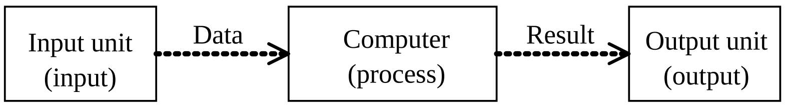


Figure 1-45 Flow of data processing

Various input units and output units are available. Each unit has appropriate functions that are suitable for input data or output format. This section describes the types, characteristics, and mechanism of typical input/output units.

5 - 1 Input Unit

Input unit is a device used for entering data in a computer. Input units include devices that enter characters or numbers by using signals, devices that read data written in a paper, and devices for entering position information (pointing devices).

5-1-1 Devices for Entering Characters and Numbers by Using Signals

(1) Keyboard

Keyboard is the most general purpose input device, and it is always supplied as a standard input device of PCs. When a key on the keyboard is pressed, the corresponding character or number is entered as a signal (code).

Keyboard

When an old keyboard is used, one depression of a key may work as if the same key were pressed several times. This is sometimes referred to as chattering. However, chattering is originally a phenomenon in which multiple ON/OFF signals occur within a short period of time (a few milliseconds after pressing a push button), in response to pressing the button with a mechanical connection once. Therefore, strictly speaking, the above-mentioned behavior is not the same as chattering.

5-1-2 Device for Reading Data Written on Paper

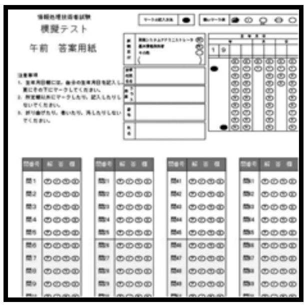
(1) OCR (Optical Character Reader)

OCR (Optical Character Reader) optically reads the handwritten characters or printed characters on the basis of the strength of the reflected light, and enters them as data. OCRs developed in the early days could only read characters that were written in special OCR fonts. However, modern OCRs can sufficiently read even handwritten characters. Therefore, it is also used for reading a zip code (or postal code) that is written on a letter or a postcard. However, there is a risk of the reading mistake (misreading) of characters, and therefore, it is necessary to thoroughly check the entered data.

OCR

(2) OMR (Optical Mark Reader)

OMR (Optical Mark Reader) is a device that optically reads the portions that are marked with black color on a mark sheet by using the reflected light, and enters them as data. Instead of reading characters or numbers, it reads the marked position information. This input device converts this position information into the corresponding data for processing. Therefore, it cannot handle stains or such spots well, but there are fewer reading mistakes. It is used for reading (grading) answer sheets of exams where there are many examination candidates.


模擬テスト
午前 答案用紙
Example of mark sheet

(3) MICR (Magnetic Ink Character Reader)

MICR (Magnetic Ink Character Reader) is a device for entering characters just like OCR; however, it can only enter characters of a special font written with a special ink mixed with magnetic material. It is not affected by stains because it reads characters that are printed with magnetized ink. Since characters cannot be created without a special device, it is used as a reading device for notes and checks.

小 切 手
小川銀行
¥1,560,000※
0123456789
Application of MICR

(4) Barcode reader

Barcode reader is a device that optically reads a bar code that represents data with bars (lines) of different thickness and different spacing. Its typical use is as the input device of a POS system, which is used in the cash registers of retail stores.

Barcode reader

[Main bar codes]

- One-dimensional bar code
  This is a horizontal bar code where vertical lines of different thickness are placed in a band. When someone simply says "bar code," this typically refers to the one-dimensional bar code. It is used extensively because it can be directly printed on packing material or containers, and printing size can be freely enlarged or reduced. In addition, it comparatively costs less, which is an added advantage.
  Typical examples include JAN code (Japanese Article Number code), ITF code (Interleaved Two of Five code), and Code128 (bar code that handles ASCII characters).
- Two-dimensional bar code (QR code)
  This is a matrix-based bar code where information is represented with small squares within a larger square. It has three symbols for detection called position detection patterns, and can recognize rotation angle and reading direction. Therefore, information can be read from any direction. Moreover, it can be read with smartphones or cell phones without using any special reading device.

149 01234 56789 1

One-dimensional bar code

Two-dimensional bar code

(5) Image scanner

Image scanner is an input device that optically reads and enters images such as drawings and photographs that are written or printed on paper by using the same principle as facsimile. The image scanner enters drawings or photographs that are written or printed on paper as a dot image, which is a set of points. An image scanner where paper is fixed and reading device is moved for reading the image data is also called an image sensor.

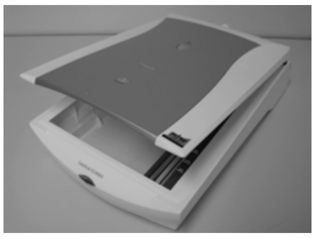


Image scanner

5-1-3 Pointing Device

(1) Mouse

Mouse is a device that feeds the amount of rotation and direction by rotating the ball that is located at the bottom. It got this name because its shape resembles a mouse. Pointing devices that have similar functions include the trackball or the acupointer where the ball is rotated by turning the bottom side up.

Mouse

(2) Joystick

Joystick is a device where a stick is moved back and forth and around, and position information is fed on the basis of the direction and its angle. It is used in operating game software.

(3) Touch screen (touch panel)

Touch screen (touch panel) is a device that feeds position information by touching the display screen with a finger.

[Input method of touch screen]

- Infrared shielding method (Infrared method)
  It detects the position from the changes in infrared ray reflection that occurs when an infrared ray beam is shielded.
- Capacitance method
  It detects the position from the changes in electrical charge on the surface which is caused by the static electricity in the portion that is touched.
- Analog resistive film method (Resistive film method)
  It detects the position from the changes in the resistance values of the resistive film in the portion that is touched.
- Matrix switching method
  It detects the position where the electrode switches that are arranged on the matrix are pressed.

(4) Light pen

Light pen is an input device that feeds position information by directly pointing and tracing on a display screen. The tip of the light pen has an optical sensor, and the position on the display can be detected from the movement of this sensor.

5-1-4 Other Input Devices

(1) Digitizer/tablet

Digitizer is an input device that feeds two-dimensional and three-dimensional drawing information as coordinate information by tracing a drawing on the panel. A compact size digitizer that can be used on a desk is specifically referred to as a tablet.
It is used for entering drawings in a design support system called CAD and creating data of an NC (Numeric Control) machine tool.

Digitizer

(2) Card reader

Card reader is used for reading information that is recorded in a card. Typical card readers include a magnetic card reader that is used for reading magnetically recorded information in a bank cash card or a credit card, and an IC card reader that is used for reading information in an IC card (a card with an embedded IC chip). The IC card is used as an entrance key card or a card key.

(3) Biometric authentication device

Biometric authentication device is a device that feeds information for biometric authentication using physical and behavioral features of human beings. There are devices that feed (authenticate) information of the intended staff members who are targeted for authentication by an authentication method, such as a face authentication device, a palm authentication device, a fingerprint authentication device, and an iris authentication device.

(4) Sound input device

Sound input device is a device that identifies sounds (analog signals) of human beings and feeds data after these analog signals are converted into digital signals by using electronic circuits (A/D converter). It is also used as a biometric authentication device for authenticating voice pattern or voice print.

(5) Digital camera

Digital camera is a device that feeds the images taken by it to a computer. Instead of the film that is used in normal cameras, it uses CCD (Charge Coupled Device) for the conversion of graphic images into digital data. In most cases, smart media, which is a type of flash memory, is used as the recording medium of images. Moreover, a digital video camera is used for recording video.

5 - 2 Output Unit

Output unit is a device that is used for exporting the processing results outside computers. Output units include displays that show the results and printers that print the results.

5-2-1 Display

Display is a device that displays images in color by combining three primary colors of light, namely RGB (Red, Green, Blue). Screen is an aggregation of points (dots), and the higher the density of dots, the higher the display quality of images. This is also called the resolution of the screen, and dpi (dots per inch) is used as a unit of resolution.

[Screen resolution standards of display]

Standard name	Screen size
VGA (Video Graphics Array)	640 × 480
SVGA (Super VGA)	800 × 600
XGA (eXtended Graphics Array)	1,024 × 768
SXGA (Super XGA)	1,280 × 1,024

Displays of present-day PCs mostly use a multiscan method that can support various resolutions. However, for supporting high resolution, graphic memory (VRAM (Video RAM)) that supports high resolution is required. Storage capacity required for VRAM is decided by the screen resolution and the number of colors per one bit (the number of bits that is required for displaying a color).

Example: On a display that has the resolution of 1,280 × 1,024, calculate the storage capacity of VRAM that is required for displaying about 16,000,000 colors by assigning 256 grades of intensity to each color of RGB.

1) Number of bits required for displaying one color
   $256 = 2^8$, therefore 8 bits/color
2) Number of bits required for displaying color information of 1 dot
   = 8 bits/color × 3 colors/dot
   = 24 bits/dot
3) Number of bits required for one screen
   = Number of bits required for one dot × Number of dots on one screen
   = 24 bits/dot × (1,280 × 1,024) dots/screen
   = 31,457,280 bits/screen ··· About 4 Mbytes/screen

In addition, for increasing the display speed and the precision of images, a graphic accelerator (a type of video chip) is required. In the raster scan display, where images are scanned line by line, there is an output method called interlaced mode. This method extracts every other scanning line and displays one image over two rounds, and thereby, increases the display speed. However, flickering can easily occur on the screen with this method, so in most of the cases, non-interlaced mode is used where scanning lines are displayed in sequence from the top.

(1) CRT display

CRT display is an output device that uses a cathode-ray tube just like television. Light is emitted by applying electron beams onto a phosphor screen, and any character or image can be drawn depending on the beam strength. While it has advantages in that the screen is easy to see and display speed is also fast, it suffers from disadvantages in that the device is large (because of depth) and power consumption is also high. Furthermore, there is another problem of burning the image on the display when the same image is displayed for a long time. In order to prevent this burning, software called a screen saver is used.

(2) LCD (Liquid Crystal Display)

LCD (Liquid Crystal Display) is an output device that uses liquid crystals whose permeation rate of light varies depending on voltage. As compared with CRT displays, it is thin and lightweight, and consumes less power. Therefore, it is mostly used in a notebook PC; however, these days it is also used in a desktop PC. Since liquid crystals themselves do not emit any light, either a reflection board is embedded or a back-light method is used where light is thrown from the rear. As compared with an STN liquid crystal display, which controls multiple pixels with one semiconductor, the response speed of a TFT liquid crystal display, which controls one pixel with one transistor, is faster and its viewing angle is also broad. Therefore, TFT LCDs are more widely used these days.

(3) PDP (Plasma Display Panel)

PDP (Plasma Display) is an output device where gas is sealed between two panels of glass and light is emitted by applying voltage. While it has higher luminance and wider viewing angle in comparison with other methods, power consumption is also large because high voltage is required.

(4) OLED (Organic Light Emitting Diode) display

OLED (Organic Light Emitting Diode) display is an output device that uses organic compounds that emit light when voltage is applied. Unlike a liquid crystal display, it emits light on its own on a glass substrate. Its response speed is fast, image quality is of high luminance because of high contrast, and the viewing angle is also wide. In addition, power consumption is small, so it has potential to become the next-generation display. However, its life is short.

5-2-2 Printer

Printer is a device for color printing by using three primary colors known as CMY (Cyan [bright light blue color], Magenta [bright red-purple color], and Yellow). However, proper black color cannot be formed even if these three colors are mixed. Therefore, in most cases, four colors of CMYK are used where black is added as the key.
Most printers print characters and drawings as an aggregation of points (dots). Therefore, the higher the number of dots per unit area, the higher the printing quality. This is referred to as the resolution of a printer, and it is expressed in the same unit as for displays, namely, dpi (dots per inch).

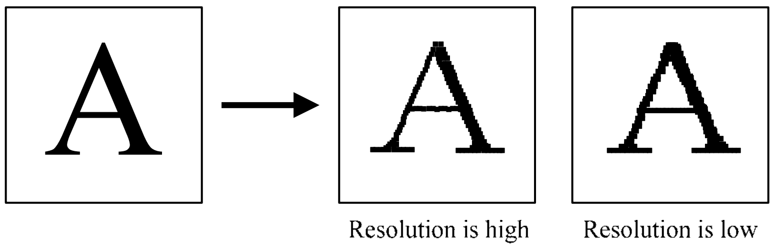


Figure 1-46 Characters are an aggregation of dots

(1) Impact printer

Impact printer is a device that prints by mechanically striking ink ribbon on paper with a print head. Although it generates noise, it is possible to simultaneously print multiple copies by placing carbon paper between multiple sheets of paper.

1) Dot impact printer

This is a serial printer that prints each character separately. From the back of the ink ribbon, it strikes the print head that has the pins each of which corresponds to a dot and these pins are arranged in the shape of characters. Kanji characters and graphic symbols can also be printed if they can be represented as an aggregation of dots.

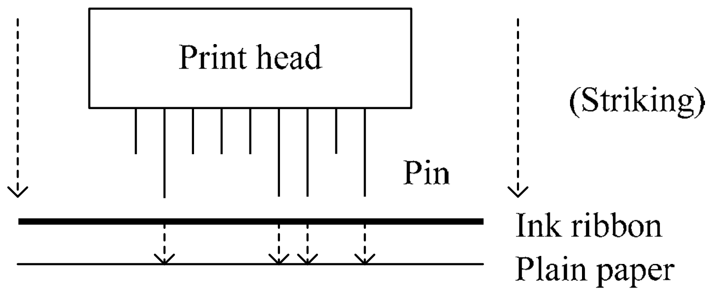


Figure 1-47 Dot impact printer

2) Line printer

This printer prints all characters in one line at a time. It uses either a print drum method, where all characters in one line are printed by striking ink ribbon on the paper while the print drum rotates, or a dot method where the number of print heads is equal to the number of characters.

(2) Non-impact printer

Non-impact printer is a device that prints by transferring ink or toner onto paper by using heat, etc. While it does not produce much noise and it can print various graphic symbols, it cannot handle simultaneous printing of multiple copies by using carbon paper.

1) Thermal printer

This is a general term for printers that use heat for printing.

- Thermal printer
  This printer uses heat and special heat-sensitive paper.
- Thermal wax transfer printer (thermal transfer printer)
  This printer transfers waxed color ink that is applied on the ink ribbon of four colors (CMYK) by melting it with heat.
- Dye sublimation thermal transfer printer
  This printer applies heat to four-color (CMYK) sublimation dyes mounted on a film in order to convert the dyes directly to the gaseous state, which is absorbed by special paper.

2) Inkjet printer

This is a serial printer that prints each character separately, in units of dots by blowing ink in the shape of characters from the print head. These days, the photo printer is commonly used where photo quality (image quality like photographs) is produced by using four or more color inks.

Inkjet printer

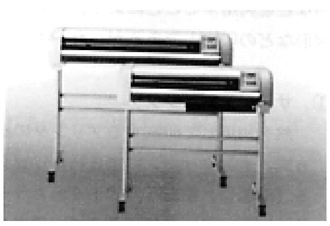
3) Laser printer

This is a page printer that prints in units of pages by affixing toner (powder ink) onto a photoconductive drum by using a laser, and then transferring the toner onto the paper. Since it uses the same principle as a copier, it is also referred to as an electrophotographic printer. A printer or a printer driver creates a print image in the bitmap format. Therefore, the size of characters and spacing between lines can be freely adjusted and even images can be printed. (The printer control code for creating page images is called page description language, and PostScript or ESC/Page is available). As the performance indicator of a laser printer, in addition to resolution dpi (number of dots per one inch), the number of pages that can be printed in one minute (ppm (pages per minute)) is used.

5-2-3 Other Output Devices

(1) Plotter

Plotter is a device that prints graphics and is used for printing drawings that are created with CAD, etc. There is the flatbed plotter, which moves the pen in *X* axis and *Y* axis by fixing the paper (this is referred to as an XY plotter), and there is the drum plotter, which moves the pen only in the *Y* axis direction while the paper wrapped on the drum is moved in the *X* axis direction.


Plotter

(2) Projector

Projector is an output device that projects the data that is inside the computer. It is generally used for enlarged projection of display images on a large screen. The brightness of the light source of the projector is expressed in units of lumen (lm).

Projector

(3) Electronic paper

Electronic paper is a very thin (about 0.1mm) display device that is made of soft material and has the characteristics of paper. Power consumption is very small, and data can be repeatedly reproduced and erased. This is one of the applied technologies of MEMS (Micro-Electro-Mechanical Systems), which is a nanometer level micro-electro-mechanical device and manufacturing technology.

(4) Voice synthesizer

Voice synthesizer converts data that is inside the computer (digital data) into analog signals by using an electronic circuit (D/A converter), and artificially creates and generate a human voice or other sounds. It is used in screen readers (screen reading software) that read out the information on the display, and is also used for making announcements in public institutions, transport facility, and event venues. Moreover, vocaloid, which sings a song by using the voice of a sampled person and by feeding melody and lyrics, is also one of the methods of using a voice synthesizer in a broad sense.

5 - 3 Other Input/Output Units

If an input unit is defined as a device that is used for entering data into a computer, and an output unit is defined as a device that is used for exporting the results of internal processing, in addition to general input/output units, the following devices are also classified as input/output units.

(1) Communication control unit

Communication control unit is a device that is used for various controls when the computer is connected to a network. It performs serial-parallel conversion of data (serialization and deserialization of characters), error control on data, etc. Since it receives data from the network and transmits data to the network, it is classified as a type of input/output device.

[LAN interface card]

This is a card-type interface device used for connecting a computer (mainly a PC) to a LAN (Local Area Network) which is a private communication network that is laid in a limited area). Wired LAN interface card, which is connected by using cable, and wireless LAN interface card, which is connected with wireless means such as electromagnetic waves, are available.

(2) Drive unit

Drive unit is a device that is used for operating other machinery or machinery. It converts the directions that are given by a computer into a mechanical operation (e.g., an operation of the robot arm that is used in a factory) and transmits it to the outside. Therefore, it can be called a type of output device. Sensors (devices that convert the measured values into electric signals) and switches that are used in machines with built-in drive units are also input devices, in a broad sense.

• Chattering

Chattering is a phenomenon where in response to pressing the push button switch of a mechanical contact once, a series of “on” and “off” signals is generated during a few milliseconds of pressing the switch. As a measure for preventing false operation, there is a method of waiting for the “on” and “off” signals to become stable.

(3) Imaging device

Imaging device is a device that feeds images. Digital cameras and digital video cameras are also classified as imaging devices. There are also electron microscopes and radio telescopes that import and analyze input images as digital data inside the computer.

5 - 4 Input/Output Control Methods

Input/output control methods are methods that control exchange of data between the input/output units and the main memory unit. When a program under execution is issued an input/output instruction, an SVC interrupt occurs to execute the interrupt program of the input/output control (the input/output control program). After that, when the input/output operation is completed, an input/output interrupt occurs to indicate the completion of input/output.

(1) Program control (direct control)

Program control (direct control) interprets the input/output control program with the arithmetic and logical unit of the CPU and gives directions to input/output units, and in this manner, data is exchanged between the input/output units and the main memory unit via the CPU. While this is the most basic control method, there is a large difference in the operating speed between CPU and input/output units, and therefore, it reduces the efficiency of utilization of the CPU.

(2) DMA (Direct Memory Access) method

In the DMA method (Direct Memory Access method), data is directly exchanged between the input/output units and the main memory unit without going through the CPU, because input/output control is performed by a special device called a DMAC (DMA Controller). When a DMA request (input/output request) is given to DMAC by the CPU, DMAC independently controls input/output, and in the meantime the CPU can execute another process, which improves its efficiency of utilization. It is widely used, mainly as the input/output control of PCs.

(3) Channel control

Channel control uses a dedicated processing unit (input/output channel or channel) for input/output. When the channel program is sent to an input/output channel from the CPU, the input/output channel independently fetches the required information and decodes the instructions. In this manner, the input/output channel controls the input/output independently of the CPU. It is mainly used as the input/output control of general-purpose computers.

[Classification of input/output channels]

- Selector channel: ··· This exclusively controls one input/output unit at a time.
- Multiplexer channel: ··· This simultaneously controls multiple input/output units.
    - Byte multiplexer channel:
      This controls low-speed data transfer in units of bytes.
    - Block multiplexer channel:
      This controls high-speed data transfer in units of blocks.

5 - 5 Input/Output Interfaces

Input/output interfaces are rules (standards) for connecting peripheral devices, such as input devices and output devices, to computers. Also, software that controls the peripheral devices connected to input/output interfaces is called a device driver.

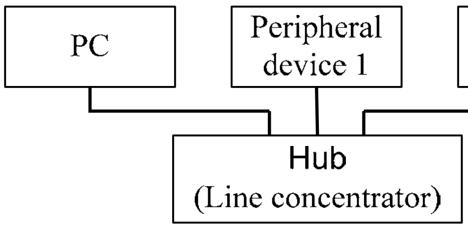
5-5-1 Classification of Input/Output Interfaces

Input/output interfaces can be classified into serial interface (serial data transfer method in units of bits) and parallel interface (parallel data transfer method in units of bytes or words). Also, since digital signals are used inside the computer, in order to connect the peripheral devices that support analog signals, it is necessary to perform conversion between analog signals and digital signals by using analog input/output interfaces (analog input/output board).

[Connection mode (topology) for connecting peripheral devices]

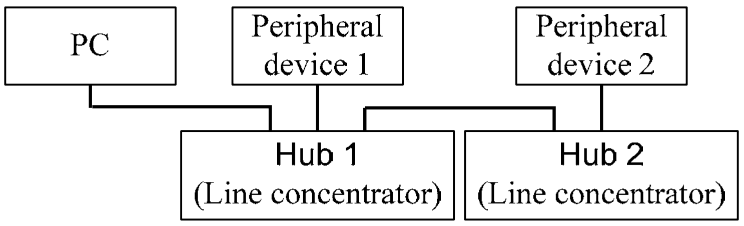
• Star connection

In this method, multiple devices are connected via a hub (line concentrator).



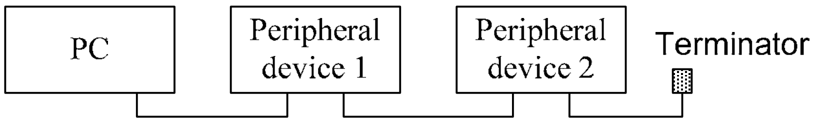
• Cascade connection

In this method, multiple devices are connected by having a multilevel connection of hubs (line concentrator). Tree-shaped connection modes like star connection and cascade connection are sometimes collectively referred to as a tree connection.



• Daisy chain connection

In this method, multiple devices are connected like a daisy chain (bunching) method from a device connected to the PC. Generally, at the other end of a cable, a terminating device (termination resistance) called a terminator is connected.



5-5-2 Types of Input/Output Interfaces and their Characteristics

(1) RS-232C (Recommendation Standard-232C)

RS-232C is a serial interface used for connecting a computer to communication devices (data circuit-terminating equipment) like modems. It is standardized by EIA (Electronic Industries Association), and it allows two-way transmission of data. These days it is also used for connecting to input/output devices such as a mouse.

• Interface board

This is a board that allows the number of ports (slots) to be increased by mounting on the expansion slot of a PC. (The analog input/output board is also an interface board.) There are some products that support serial interface board, parallel interface board, and data transfer method (input/output interface). For example, in order to increase RS-232C interface ports, you can mount a serial interface board or an RS-232C interface board.

(2) SCSI (Small Computer Systems Interface)

SCSI is a parallel interface used for connecting computers like PCs to peripheral devices. It is standardized by ANSI (American National Standards Institute), and it allows two-way transmission of data. It is mainly used for high-speed data transfer in hard disk drives and CD-ROM drives. With daisy chain connection from a device that is connected to the PC, SCSI can connect up to seven peripheral devices. A SCSI ID (identification number: 0 to 7) is assigned in order to identify the connected devices. In this case, the connection sequence does not matter as long as the ID is unique; that is, it need not be in the sequence of 0 to 7.

[Advanced SCSI standards]

• SCSI-2

• Fast SCSI:	Transfer speed of 10 Mbps with 8-bit bus
• Wide SCSI:	Transfer speed of 20 Mbps with 16-bit bus

• SCSI-3

• Ultra SCSI:	Transfer speed of 20 Mbps with 8-bit bus
• Ultra Wide SCSI:	Transfer speed of 40 Mbps with 16-bit bus

In SCSI-3, serial interface technique based on the serial SCSI is standardized, which allows high-speed data transfer at 50 to 200 Mbps.

(3) USB (Universal Serial Bus)

USB is a serial interface standardized in 1996, and it can connect various peripheral devices such as a keyboard, a mouse, and a modem. By using a USB hub, up to 127 devices can be connected in the tree configuration. Multiple standards for the shape of connectors are defined according to the device to be used.

[Data transfer modes of USB]

- USB1.1

  It is available in low-speed mode (1.5 Mbps) that is used for connecting low-speed peripheral devices such as a keyboard and a mouse, and full-speed mode (12 Mbps) used for connecting somewhat high-speed peripheral devices such as a printer and a scanner.
- USB2.0

  There are three types of data transfer modes. A mode is added to two data transfer modes of USB1.1. This third mode is called high-speed mode (480 Mbps) and is used for connecting high-speed peripheral devices such as external hard disks.
- USB3.0

  In addition to the three types of data transfer modes of USB 2.0, SuperSpeed mode (5,120Mbps) is added. Also power management is improved in the standard.

(4) IEEE 1394

IEEE 1394 is a serial interface that allows high-speed transfer of data and is used for connecting multimedia-related devices such as digital cameras. It is available in three types of data transfer modes (100 Mbps, 200 Mbps, and 400 Mbps), and its main characteristic is that devices can be connected even without going through a PC. This standard is owned by IEEE (Institute of Electrical and Electronics Engineers), and Apple's FireWire and Sony's i.Link are trade names of IEEE 1394.

[Common functions of USB and IEEE 1394]

- Bus power method (Bus powered)

  In this method, power is supplied from a PC via connection cable. The amount of power that is supplied by this method is limited. Therefore, operation of devices that have large power consumption cannot be ensured.
- Hot plug

  In this method, a cable (device) can be connected and disconnected while the power supply of the computer is on.

(5) Centronics

Centronics is a parallel interface used for connecting with printers, and it is standardized as IEEE1284 by IEEE (Institute of Electrical and Electronics Engineers) including up to the extended specifications. There are standards such as D-Sub 25 pins (24 pins) and 36 pins.

(6) GPIB (General Purpose Interface Bus)

GPIB is a parallel interface that is used for connecting with measuring instruments, and it is standardized as IEEE 488.1. Up to 31 peripheral devices can be connected.

(7) Serial ATA (Serial Advanced Technology Attachment)

Serial ATA is a high speed interface that is developed by modifying the parallel interface ATA (ATA/ATAPI-4) to the serial interface. Parallel interface ATA (ATA/ATAPI-4) was the official standard of IDE (Integrated Device Electronics) that was standardized by ANSI (American National Standards Institute). It is generally used for connecting high-speed data transfer devices such as built-in hard disk and optical disc devices like CD-ROM drive. At present, serial ATAIII that has the communication speed of 6 Gbps is most widely used. While serial ATA basically connects devices and ports (controllers) on a one-to-one basis, it is possible to increase the number of ports by using a port multiplier. In the conventional ATA, a master-slave connection was used where two devices that have a master-slave relationship were connected to one cable. However, this method is not used in serial ATA.

(8) IrDA (Infrared Data Association)

IrDA is an association for creating standards for data communication using infrared rays, and it is also a serial interface that is created by this association. The maximum communication distance is 1 meter, and it is necessary to place the communication ports within ±15 degrees without any shields such as partitions. Infrared communication that does not require connection cables is used for data exchange between personal digital assistants like smartphones or tablet terminals, and for data exchange between personal digital assistants and notebook PCs.

(9) Bluetooth

Bluetooth is a wireless technology standard that connects devices such as cell phones and PCs, and is an interface for exchanging data and voice by using the frequency band of 2.4GHz (2.402 to 2.480 GHz), which can be used all over the world. Unlike infrared rays, communication is possible even if there is a block or an obstacle between devices. Therefore, it allows the transfer of data between devices that are separated with low partitions.

[Bluetooth standards]

- Number of devices which can be connected

  In a general standard, logically up to 7 devices can be connected simultaneously. But there is a standard that allows almost unlimited number of devices (about 2^{32} devices) to be connected by expanding the address bits from 3 to 32.
- Communication speed

  Communication speed is determined by the version. In the case of Bluetooth 3.0, the maximum communication speed is 24 Mbps. In the more advanced standard of Bluetooth 4.0, the maximum communication speed can be limited to 1Mbps in order to support the low power consumption mode (BLE (Bluetooth Low Energy)).
- Communication distance

  Communication distance is determined by the class. The following table gives approximate values of communication distance of each class.

Class	Communication distance
Class 1	About 100m
Class 2	About 10m
Class 3	About 1m

(10) HDMI (High-Definition Multimedia Interface)

HDMI is a standard of communication interface that transmits voice and video as digital signals. It was jointly created by Hitachi Ltd., Panasonic Corporation, Phillips, Thomson Multimedia, Sony Corporation, and Toshiba Corporation, with Silicon Image of the United States taking the lead. With a single cable where voice, video, and control signals are integrated, wiring of AV devices can be simplified to just one wire. While it was mainly developed as an input/output interface for home appliances and AV devices, at present it is widely used in PCs as well. The base of HDMI is the serial interface DVI (Digital Visual Interface), which produces high quality images by transmitting digital signals instead of analog RGB, which sends video signals by using analog signals of three primary colors (RGB) of light.

5-5-3 Device Driver

(1) Device driver

Device driver is software that controls peripheral devices. Device driver is specific software

that is required for each peripheral device, and peripheral devices cannot be used without device drivers. For example, in order to use a printer, a device driver (printer driver) for the printer is required. Moreover, software that manages the device drivers is called a device manager. When multiple devices are connected, the device driver has another role to play in synchronizing with each device (aligning the timing of operation).
While each device driver for basic peripheral devices is provided in advance, we need to obtain a device driver for the other peripheral device (e.g., a newly purchased peripheral device) from the Internet or CD-ROM that is supplied with the device and then install it separately (i.e., install software in the computer).

(2) BIOS (Basic Input/Output System)

BIOS is software (program) that provides the standard means of input and output in computers. When the computer is started, the OS (software for controlling the computer) embeds the device drivers of the peripheral devices that are detected by the BIOS so that the use of peripheral devices become possible.

(3) PnP (Plug and Play)

PnP (Plug and Play) is a function (mechanism) that allows the immediate use of a peripheral device by just connecting it to a computer. As for the basic mechanism, as soon as a peripheral device is connected, the OS recognizes the connected device and automatically sets the device driver so that the device can be used. Since the user does not need to perform any manual settings, even a beginner who does not have any knowledge of the device driver can easily use the computer and the peripheral device.
In order to implement a hot plug (a function that allows the connection and disconnection of a device while the power supply is kept on) that is offered by USB or IEEE 1394, it is necessary that the OS of the applicable computer support Plug and Play.

This file is a part of the book:

Original Japanese edition published by Infotech Serve Inc.

ITワールド

(ISBN978-4-906859-06-1)

Copyright © 2013 by Infotech Serve Inc.

Translation rights arranged with Infotech Serve Inc.

Translation copyright © 2015 by Information-technology Promotion Agency, Japan

Information-technology Promotion Agency, Japan

Center Office 16F, Bunkyo Green Court, 2-28-8, Hon-Komagome, Bunkyo-ku, Tokyo, 113-6591 JAPAN