COMPUTER ARITHMETIC

Lawrence Borg

This article is mainly intended as an introduction to binary numbers as another system of representation of numbers and their relation to digital computers. However, a preliminary discussion of the basic types of computers has been included as a matter of interest

There are two basic types of computer currently in use — the analogue and the digital computer. The hybrid computer is a combination of both. In its calculations, the analogue computer represents numbers of quantities by physical magnitudes of some measurable thing (the analogue) which the computer can handle. The analogue representing the number or quantity may be the length of a rule (as in the slide rule), the voltage across a resistance (as in the electronic analogue compupter), angular position of a pointer (as in the speedometer) or any physical quantity that can be easily measured. The accuracy of such representations and subsequent calculations is of course limited by the accuracy with which we can measure the physical quantities involved. Probably the best known analogue calculator is the slide rule. It is based on the use of logarithms. By adding the logarithms of two numbers, the logarithm of the product of the two numbers is obtained. Distance along the slide rule is directly proportional to the logarithm of a number, so that multiplication is easily achieved by adding together lengths of the scale. Division is achieved by subtracting lengths.

The digital computer, on the other hand, represents its variables in a quantised or digital form. This means that digital calculations are based on the use of actual numbers or digits which are operated on directly. Thus, numbers must be represented by using a discrete state or condition to represent each symbol (0-9 for decimal numbers). One of the earliest types of digital calculators is the abacus by which a number can be exactly represented by a combination of of beads on several rods. Now consider an automatic decimal number counter consisting of mechanical gear wheels of the type found in most car mileage indicators. The ten symbols are represented by ten cogs on the gearwheels and each decade has its own individual gearwheel. Each complete revolution of a gearwheel (count of 10, 10^2, 10^3, etc.) causes the next gearwheel, representing the next highest power of ten, to enmesh so producing the effect of a carry. However, such a mechanical system is limited by friction, wear and tear, etc. For high speed counting the same task must be performed electronically. However, a difficulty arises because an electronic device with ten distinct states is very rare and when specially made tends to be very expensive. The obvious solution for the realization of the electronic digital computer is to use a number system with fewer symbols. Thus, if we could use a method of counting which involved only two symbols we could utilize the many two-state devices

which are readily available in the form of switches or relays (on/off), transistors (conducting/cut off), capacitors (charged/discharged), etc., to devise an economical system of number representation. Such a number system is called the binary system.

NUMBER SYSTEMS

The most common numbering system today is the decimal system which utilizes ten digits: 0,1,2,3,4,5,6,7,8,9. The number of digits used in the system is known as its base or radix. The radix of the decimal system is ten.

Each term in a number system is associated with the radix raised to a power according to its position in the number. Thus for example, the decimal number 2375 can be represented as follows

1000's	100's	10's	1's
2	3	7	5

In other words, it may be represented in terms of the radix as

$$2375 = (2 \times 10^3) + (3 \times 10^2) + (7 \times 10^1) + (5 \times 10^0)$$

Looking at it this way it can be seen that there is nothing unique about the decimal system with its ten symbols and place value system of representation, i.e. units, tens, hundreds columns. In fact, the only reason for this choice of base seems to be due to the fact that we have ten fingers and ancient man used to count with his fingers. In fact, it was once suggested that the duodecimal system, i.e. the system with a base twelve, should replace the decimal system since at that time twelve was used in such units as feet and inches, shillings and pence, and the dozen. Almost all cash registers used to work using this system which means that they used to effect a carry once every twelve counts. The duodecimal system uses twelve symbols: 0,1,2,3,4,5,6,7,8,9,T,E. The last two symbols are the equivalent of the decimal numbers 10 and 11. Thus,

$$245_{12} = (2 \times 12^2)+(4 \times 12^1)+(5 \times 12^0) = 341_{10}$$

$$1E3T_{12} = (1 \times 12^3)+(E \times 12^2)+(3 \times 12^1)+(T \times 12^0)$$
$$= (1728 + 1584 + 36 + 10)_{10} = 3358_{10}$$

N.B. when we want to indicate the base or radix of a number we write the number followed by a suffix.

THE BINARY SYSTEM

This approach can be extended to any other radix. So it can easily be seen that the simplest system that can be devised is one which uses just two symbols, 0 and 1. This is called the binary system and the digits are often referred to as *bits* (*bi*nary digi*ts*). Hence, in the binary scale any number can be represented by just these two digits. The radix is therefore two so that consecutive digits represent consecutive powers of 2. Hence, reading from right to left, the positions in a binary number have the decimal values 1, 2, 4, 8, 16, 32, etc. Thus

$$11001_2 = (1 \times 2^4)+(1 \times 2^3)+(0 \times 2^2)+(0 \times 2^1)+(1 \times 2^0)$$
$$= (16 + 8 + 0 + 0 + 1)_{10} = 25_{10}$$

$$1100101_2 = 2^6+ 2^5+ 2^2+ 2^0$$
$$= (64 + 32 + 4 + 1)_{10} = 101_{10}$$

Decimal, duodecimal, octal and binary representations of the same numbers are shown in the following table, the octal being introduced because of its extensive use in computer programming.

System	Duodecimal	Decimal	Octal	Binary
Radix	12	10	8	2
	0	0	0	0
	1	1	1	1
	2	2	2	10
	3	3	3	11
	4	4	4	100
	5	5	5	101
	6	6	6	110
	7	7	7	111
	8	8	10	1000
	9	9	11	1001
	T	10	13	1010
	E	11	12	1011

Obviously the greater the radix the most economical is the system in terms of the number length. However, the binary notation is the most economical in symbols since only two different are used. Hence, the most important practical reason for choosing the binary system for the arithmetic of the electronic digital computer is the ease with which a number can be represented by the state

of an electronic device which can be switched either on or off. In fact, all electronic digital computers are really complicated switching networks, each switch capable only of two states, either ON or OFF. It is possible to design these switches so that the computer can perform the most complicated logical and arithmetic operations in a few thousandths of a second.

This principle of electronic representation can be demonstrated by means of a system of lamps and their corresponding switches as shown in fig. 1. A lamp switched ON represents the '1' digit and a lamp switched OFF represents the '0' digit. Thus, the state of the lamp shown in fig. 2 represents the number

1001_2 or 9_{10}

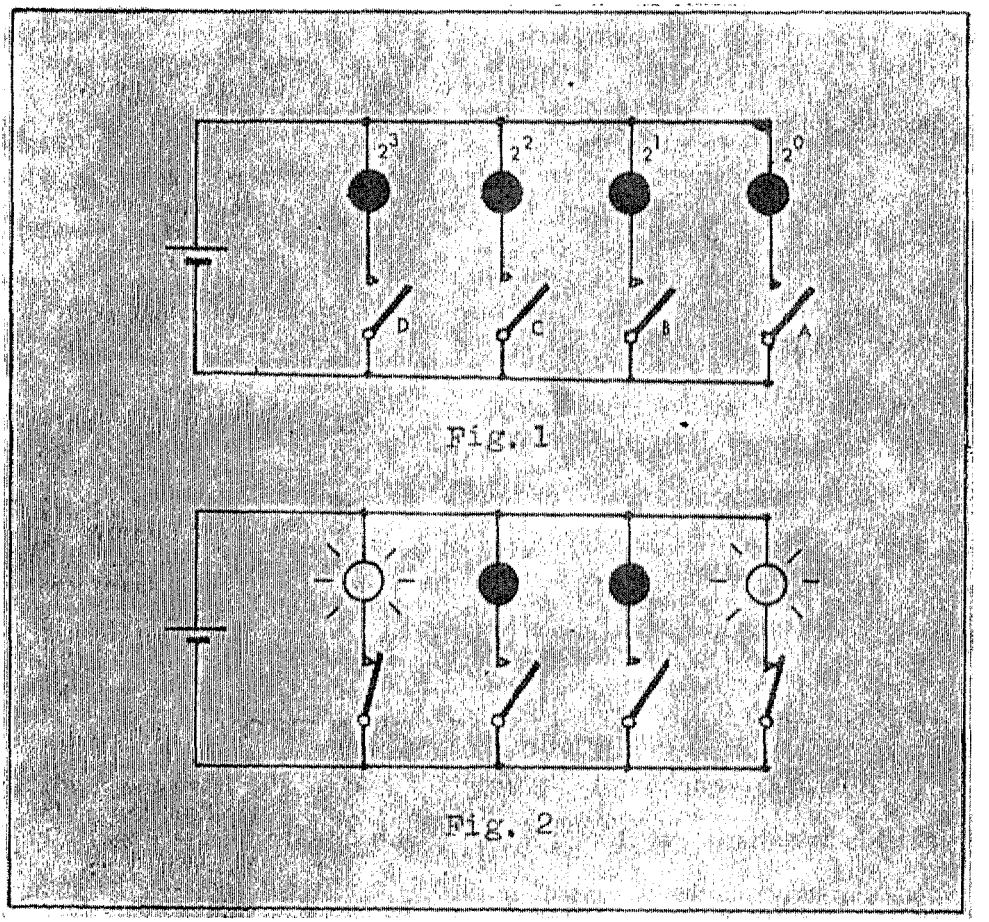

Fig. 1

Fig. 2

PARALLEL AND SERIAL MODES

The above case is called a parallel mode of operation in which an n-bit number requires n switches and all the digits appear simultanously. In another form of operation called serial mode a single switch and lamp only is employed and activated in sequence. Each sequence, which lasts for a fixed interval of time, represents either the digit '1' or '0' according whether the switch is ON or OFF. Thus, an n-bit number would appear in sequence, least significant digit first, and would require n digit times for the complete number to appear. Serial systems are much slower than parallel systems but require considerably less circuitry (hardware).

ARITHMETIC OPERATIONS

To represent decimal numbers in binary requires approximately three times as many digits. This makes the binary system unsuitable for conventional everyday arithmetic. However, arithmetic using binary numbers is a far simpler procedure than the corresponding decimal process, due to the very elementary rules of addition and multiplication. In fact, the complete binary addition and multiplication tables are:

$0 + 0 = 0$	$0 \times 0 = 0$
$0 + 1 = 1$	$0 \times 1 = 0$
$1 + 0 = 1$	$1 \times 0 = 0$
$1 + 1 = 0$ carry 1	$1 \times 1 = 1$

Two illustrative examples are now given in which the same operation has been performed in both the decimal and binary system:

Examples.	Add 19 to 25		Multiply 26 by 18
		26	11010
		18	10010
decimal	binary		
19	10011	208	110100
25	11001	260	110100000
44	101100	468	111010100

REPRESENTATION OF A NUMBER LESS THAN UNITY

In practice we also meet with quantities which are less than one or with a whole number and a fraction (mixed numbers). These quantities can be indicated in any system by placing a dot or point before the fractional part of the number. This point indicates where the power to which the radix is raised becomes negative. In the binary system this dot is called the binary point. Thus,

$$10{\cdot}11_2 = 2^1 + 2^{-1} + 2^{-2} = (2{\cdot}75)_{10}$$

RADIX CONVERSION

To convert any number in any radix to the decimal system, expand the number in powers of the radix and add the terms. This has already been shown in the various examples.

When converting a decimal number into binary, the parts to the left and and right of the decimal point must be dealt with separately. The integral part is successively divided by 2, the remainders (which can be either 0 or 1) forming the required number. The fractional part is multiplied successively by 2, the resulting integral parts giving the required number. For example, to convert 53.42 from decimal to binary the procedure is as follows:

integral part		fractioanl part
2) 53		0.42 x 2 = 0.84
2) 26	r 1	0.84 x 2 = 1.68
2) 13	r 0	0.68 x 2 = 1.36
2) 6	r 1	0.36 x 2 = 0.72
2) 3	r 0	0.72 x 2 = 1.44
2) 1	r 1	0.44 x 2 = 0.88
0	r 1	

$53_{10} = 110101_2$ $\qquad$ $0{\cdot}42_{10} = 0{\cdot}011010_2$

$53{\cdot}42_{10} = 110101{\cdot}011_2$ correct to three binary numbers.

BINARY-DECIMAL CODES

Consider the numbers $\qquad$ $365_{10} = 101101101_2$

The latter number is a natural or pure binary number. Though this is ideal for electronic computing machines, the operator of such machines still likes to think and communicate in the decimal system. However, a compromise is reached by coding decimal numbers in binary form in such a way as to be acceptable to both human and machine element. The most common is the 8421 or natural binary-coded decimal (NBCD). In this code four binary bits are used

to covey the units in the binary number. Another four bits are used to represent the tens and so on. For example, the decimal number 365 expressed in NBCD is 0011 0110 0101.

ELECTRONIC DISPLAY

For those interested in electronics, the following is a circuit designed to display in sequence the decimal numbers and their equivalent in NBCD. Light emitting diodes (l.e.d.'s) are used to represent the bits (ON represents 1 and OFF represents 0) and a seven segment l.e.d. display to represent the decimals. The integrated circuits used are popular and easily obtainable types. The circuit has many other applications such as a timer and a high speed counter/display.

Fig. 3 shows a schematic diagrm of the whole system displaying three decades. Fig. 4 shows the circuitry involved in the first stage (the units section). Of course, the other stages are similar.

brief description·

IC1 is connected as a pulse generator with a frequency of about one pulse per second to represent the data.
IC2 (4 & 6) is an NBCD counter which resets to zero after each count of nine. Its output is monitored by the four l.e.d.'s.
IC3 (5 & 7) accepts the 4-bit NBCD output of IC2 and decodes this data to drive a seven segment l.e.d. display indicator.

THE GAME OF NIM

It was my intention when writing this article about computer arithmetic to avoid as much theory as possible. In fact, I want to end by describing an interesting game which is related to the theory of binary numbers. This is the game of Nim which is a game for two players and is usually played with matches. A number of matches are placed in three or more heaps, each heap containing an arbitrary number. With every move a player may take any number of matches from one, and only one, of the heaps, He must remove at least one match but he can also take all the matches in one heap. The winner is the player who removes the last match or the last heap. (In an alternative form of the game the winner is the one who forces his opponent to take the last match.)

Winning strategy

Now, any player who has a knowledge of binary numbers can force a win possibly at his very first or consecutive move because it happens that the winning strategy is closely related to the binary representation of the number of matches in the heaps. One can be sure of winning even at the very beginning because any arrangement of heaps can be categorised as either a winning or losing position. Also, if the arrangement represents a winning position the next move is certain to create a losing position. Thus if player A (who knows the

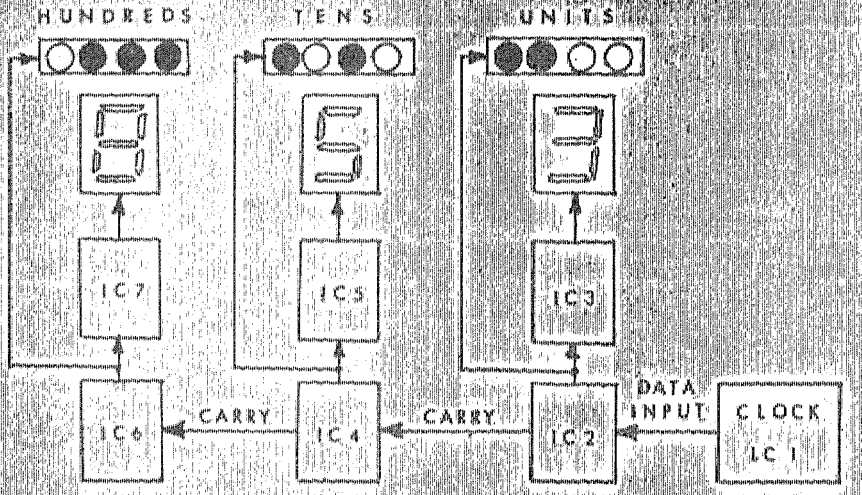
HUNDREDS
TENS
UNITS
IC 7
IC 8
IC 3
IC 6
CARRY
IC 4
CARRY
IC 2
DATA INPUT
CLOCK
IC 1

Fig. 3

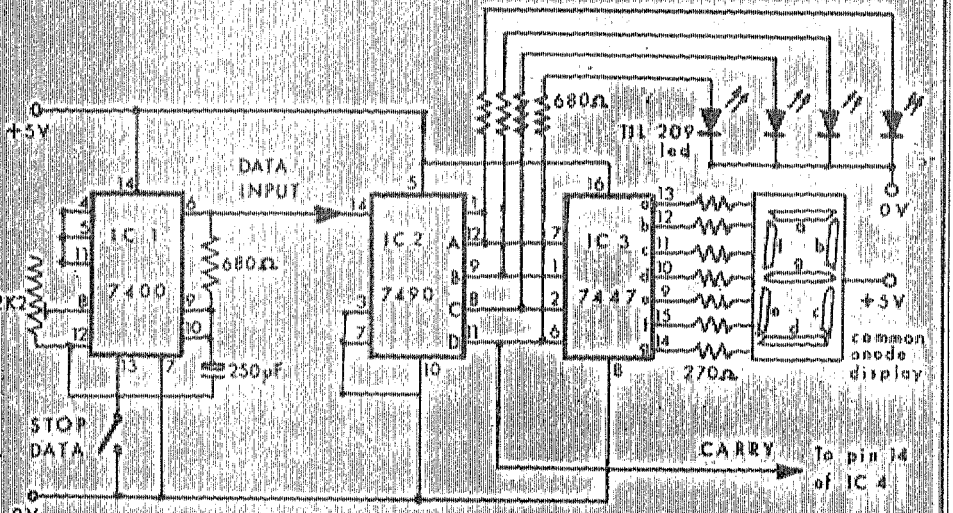
+5V
0V
DATA INPUT
IC 1
7400
2K2
680Ω
250μF
STOP DATA
IC 2
7490
A
B
C
D
680Ω
TIL 209 led
IC 3
7447
270Ω
common anode display
+5V
0V
CARRY
To pin 14 of IC 4

Fig. 4

game) starts or reaches a winning position, his opponent B is bound to leave a losing one after his move. In his next move A converts the position to a winning one by making the correct move. This cycle repeats itself until A finally wins.

To discover whether a given combination represents a winning position or not, you first express in the binary scale the number of matches in each heap. For example, suppose there are four heaps containing 7, 5, 4, and 3 matches. Expressing these numbers in binary form using three columns we get

7	1 1 1
5	1 0 1
4	1 0 0
3	1 1
	3, 2, 3

The bits in each column are added so that in this case we get the numbers 3, 2, 3 as shown. A winning combination is one in which the sum of the bits in each column is *even* (0 is considered even). It must be pointed out that the sums of individual columns are considered. There is no carry over from one column to the next as there would be in ordinary addition of binary numbers. Note that in our example the combination is a losing one. If it is your turn to play you proceed by either (i) removing 5 from the 7 matches, or (ii) removing all the matches of the second heap since 2, 5, 4, 3 and 7, 4, 3 are winning-combinations.

2	1 0
5	1 0 1
4	1 0 0
3	1 1
	2, 2, 2

7	1 1 1
4	1 0 0
3	1 1
	2, 2, 2

If you choose the alternative form of the game in which the player who removes the last match is the loser, the above procedure still holds with the only exception that in your final moves you must never leave an even number of heaps containing only one match.

Lawrence J. Borg. B.Sc. (Eng.) teaches Mathematics at the Upper Secondary School, Valetta.