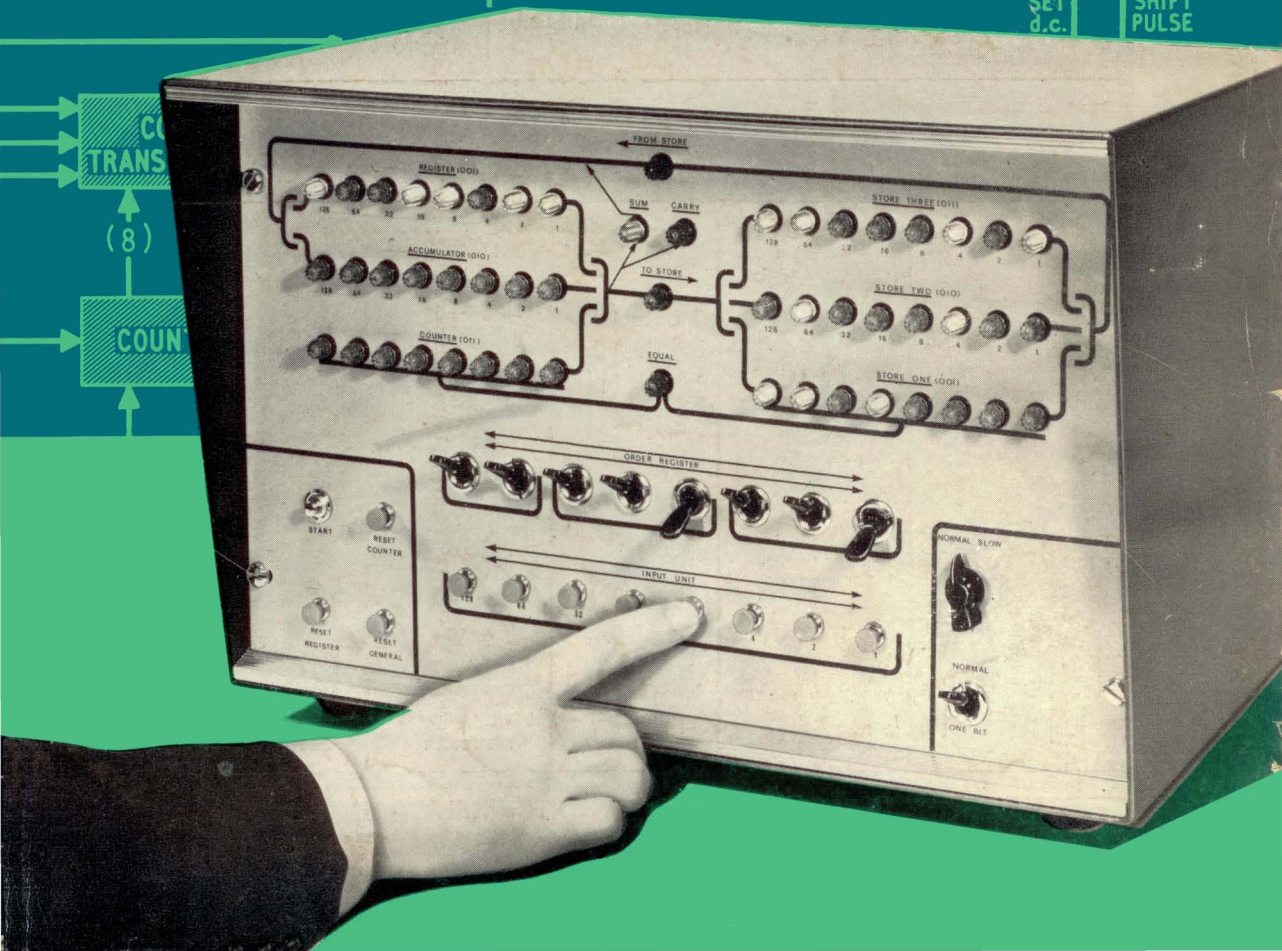
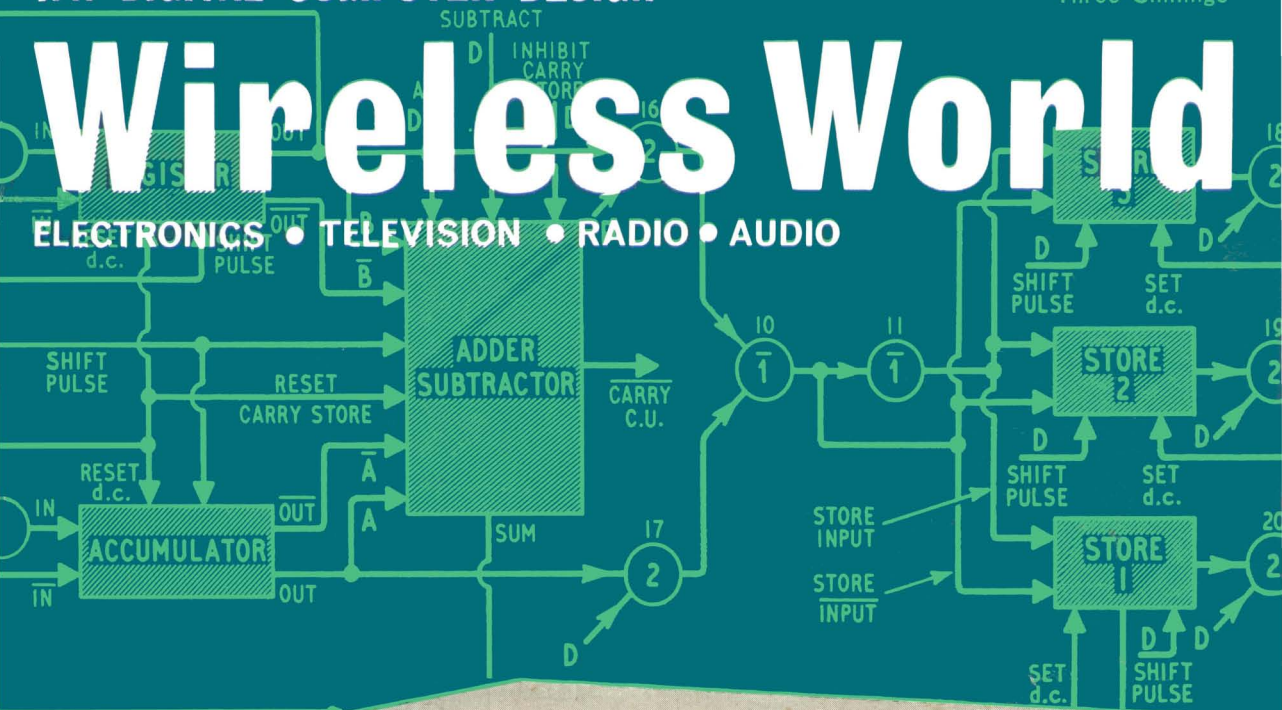
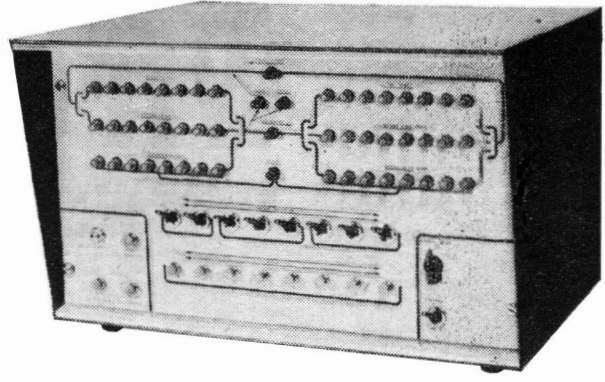


Wireless World

ELECTRONICS • TELEVISION • RADIO • AUDIO



WIRELESS WORLD DIGITAL COMPUTER



1—Outline description : basic theory : circuit elements

Low-cost desk-top binary machine for small-scale calculations and for use in schools as a teaching aid, designed by B. Crank of "Wireless World" staff. Numbers are fed in manually and results of calculations are read from indicator lamps. Instructions, entered in binary coded form by a set of switches, are interpreted and carried out automatically by the machine. www.keith-snook.info

THE *Wireless World* Digital Computer has been designed as a low cost system capable of demonstrating basic computer methods and various operations in the binary number system. It will add, subtract, multiply and divide eight-bit binary numbers, which are entered manually by means of press switches. It will also complement a binary number, and this feature makes possible arithmetic operations with mixed positive and negative numbers and subtraction using the 1s and 2s complement methods. The machine can be programmed to convert natural binary numbers into natural binary coded decimal form, making the job of interpreting results easier. The largest number that can be accommodated by the computer is the maximum obtainable with eight binary digits, which is 255.

Results of calculations and the states of all major circuits are indicated on the front panel by small neon lamps. This means that each computer operation can be analysed in detail and fault diagnosis is made easier. Instructions to the computer to perform required operations are entered, in numerical code form, by means of a set of eight switches, and the machine interprets the code and carries out the instructions automatically.

A choice of three operating speeds is provided. These are: "one bit," in which the start button is used to initiate separately each operation at each successive binary position; "slow," in which pressing the start button causes a complete arithmetical operation (e.g. adding two 8-digit numbers) to be performed at the rate of about 2 binary positions per second; and "normal" which is similar in principle to "slow" but at the higher speed of 2,500 positions per second. In its present form the computer will carry out one instruction—a complete arithmetical operation—at a time. With the addition of a few extra parts a whole sequence of instructions could

be carried out automatically, enabling basic programming to be taught.

A simplified block schematic diagram of the computer is shown in Fig. 1. Numerical data are fed straight into the arithmetic unit by the data input unit and are operated on by the computer in a manner determined by the order register, at one of three speeds (mentioned above) selected by the demonstration switching. The order register is the means by which binary coded instructions to the computer to perform a particular operation are fed in and held. The order decoder translates the instruction presented to it by the order register into a form that the computer can "understand," and causes it to be carried out by routing pulses, generated in the control unit, to the correct sections of the computer. The exact number of pulses generated by the control unit will depend on what the decoder "tells" it to do and on the internal condition of the arithmetic unit. Data can be transferred from the arithmetic unit to the store for later use or from the store to the arithmetic unit. The condition of all circuits in the arithmetic unit and store are continuously monitored on the front panel by the readout section so that if desired any particular operation can be analysed in detail.

The computer operates in the "serial" mode, which means that the binary information being transferred along the routes shown in Fig. 1 is represented by time sequences of electrical states. Thus when a number is being handled the digits in the successive binary positions are dealt with one after the other, starting with the least significant digit and working upwards.

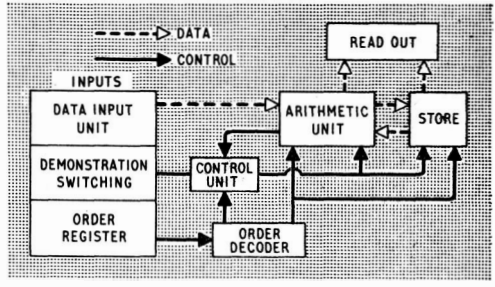


Fig. 1. Simplified schematic showing principal units of the computer.

An important factor in any project is cost and everything possible has been done to keep this within reasonable bounds. The prototype cost something in the region of £50-60, not including the cabinet. The transistors used are, for the most part, reject germanium types available for under 1s each, the remainder being 60 V silicon types that can be bought for about 2s each. An attempt has been made to use resistors of the same value wherever possible so that price reductions can be obtained by quantity buying (say 2d each). The large quantity of diodes used cost about 4½d each. The main difficulty is in obtaining cheap capacitors, as prices range from about 9d to 1s per item. However, if a systematic approach is made to this problem much more favourable prices can be obtained. The method adopted in the prototype was to ask various retailers for quotations for the quantities involved, in this manner a price of 4d per capacitor was achieved.

Before construction of the computer is contemplated it is essential that the intending builder be thoroughly conversant with the theory involved (see "reminder"

sections following). Accuracy in construction is equally important, for while trouble-shooting on the correctly built computer is not too difficult, locating faults when wiring errors are involved can be very trying.

Stored-programme facility.—Development work is being done on a sub-routine store for the computer. Early results are encouraging and should the store prove to be reliable it will be described in detail later. Basically it provides a further 64 words of storage (512 bits) that can hold either control words or data. Each word is stored by means of wired-in diodes or by diode "pegs" inserted into a matrix programming board. As the computer completes each operation the next instruction is automatically fed to the computer from the sub-routine store and is executed and this process continues until a "stop" instruction is received by the computer. In this way complete sequences of up to 64 separate instructions can be carried out automatically and basic programming can be taught and demonstrated.

Continued on page 369

REMINDER ON BINARY ARITHMETIC

Binary notation.—In the binary system, only two characters are required for counting, and we shall use the conventional 0 and 1. As in the decimal system the digits have positional as well as numerical value as shown in the table (right).

The values of digits in successive positions from right to left are increasing powers of two, $2^0, 2^1, 2^2, 2^3$ (or 1, 2, 4, 8). Each binary digit is termed a "bit" and a complete binary number a "word." The weights of the digits in the eight-bit word used in our computer are therefore $2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7$ (or 1, 2, 4, 8, 16, 32, 64, 128).

To convert a binary number to a decimal number one can add the weights for each column in which a 1 appears. Consider the word 0110 (which from the table can be seen to equal decimal 6). The decimal number is obtained as follows:— $0110 = (0 \times 8) + (1 \times 4) + (1 \times 2) + (0 \times 1) = 0 + 4 + 2 + 0 = 6$.

A decimal number can be converted into a binary word by successive division by the weights to give successive quotients of 1s and 0s as follows:

Convert decimal 163 to binary.

```

128)163   1
   128
   ---
    35    0
    32    1
    ---
     3    0
     3    0
     2    1
     ---
     1    1
     1
     ---
     0
  
```

Reading the right hand column of quotients from top to bottom, this gives $163 = \text{binary } 10100011$.

A number in the decimal system is based on powers of 10 and is said to have a radix of 10; similarly the binary system has a radix of 2. To indicate the radix being used, where necessary, the radix will be enclosed in brackets at the end of the number as is standard practice, i.e. $163_{(10)} = 10100011_{(2)}$

In addition to the pure or natural binary system discussed above, the natural binary coded decimal (n.b.c.d.) system is used in the computer. This uses four bits for each decimal

Decimal Number	Positional value				Decimal Number	Positional value			
	2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)		2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

place, these bits being the natural binary representation of each decimal digit, i.e.

$163_{(10)}$ in n.b.c.d. is 0001 0110 0011

The instructions to the computer are given in another number system with a radix of 8 known as the octal code. The method used to convert a pure binary number to octal is similar to that for converting n.b.c.d. to decimal. The number is divided into groups of three digits starting from the right, then the decimal equivalent of each group is written down, as follows:—

00/001/001
0 / 1 / 1 $00001001_{(2)} = 011_{(8)}$

Another example:—

11/101/111
3 / 5 / 7 $11101111_{(2)} = 357_{(8)}$

Because the computer uses only an eight-bit instruction word two bits appear in the left hand group, and therefore the maximum octal number that can appear in the left-hand place is 3.

Binary arithmetic.—This is best started by examining the following rules for adding two binary numbers:—

$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 0$ carry 1 to next most significant position.

Where the addition of two 1s takes place, as there is no symbol to represent any number greater than 1, a carry to the next

most significant position occurs. This is the same as saying $2^0 + 2^0 = 2^1$ or $1 + 1 = 10$. Throughout this article, to indicate in a table that a carry has been generated the following symbolism will be used:

$$1 + 1 = 0 \rightarrow 1$$

The next step is to add two four-bit binary numbers. The working is as follows:—

$$\begin{array}{r} 2^4 2^3 2^2 2^1 2^0 \\ 1111 + \\ 0110 \\ \hline 10101 \quad (\text{sum}) \\ 111 \quad (\text{carries}) \end{array}$$

From this it is seen that in order to add two binary numbers it is necessary to be able to add three digits to take into account any carry that may be generated during the previous addition. Another addition table is obviously called for:

A	B	C	
0	+	0	+ 0 → 0
1	+	0	+ 0 = 1 → 0
0	+	1	+ 0 = 1 → 0
0	+	0	+ 1 = 1 → 0
1	+	1	+ 0 = 0 → 1
1	+	0	+ 1 = 0 → 1
0	+	1	+ 1 = 0 → 1
1	+	1	+ 1 = 1 → 1

Binary subtraction can be explained by a similar table. However, here a "borrow" can occur instead of the carry in addition. In the following table a borrow is indicated in the same way as a carry i.e. → 1.

A	B	C	
0	-	0	with an 0 borrow = 0 → 0
1	-	0	" " = 1 → 0
0	-	1	" " = 1 → 1
1	-	1	" " = 0 → 0
0	-	0	" " = 1 → 1
0	-	1	" " = 0 → 1
1	-	1	" " = 1 → 1
1	-	0	" " = 0 → 0

As an example we will subtract 01 from 10, as follows:—

$$\begin{array}{r} 2^1 2^0 \\ 10 \quad (\text{positional values}) \\ 10 \quad (\text{minuend}) \\ 01 \quad (\text{subtrahend}) \\ \hline 01 \quad (\text{difference}) \\ 1 \quad (\text{borrow}) \end{array}$$

Starting with the right hand column $0 - 1 = 1 \rightarrow 1$. In the left-hand column we have already borrowed a 1, so this has to be returned and we get:— $1 - 0$ with a 1 borrow = $0 \rightarrow 0$.

Hereforward a borrow will be called a carry and an unnecessary term dispensed with.

Multiplication can be performed by repeated addition (e.g. $8 \times 4 \equiv 8 + 8 + 8 + 8 = 32$) and division by repeated subtraction, e.g. $32 \div 8 \equiv [((32 - 8) - 8) - 8] - 8 = 0$. The quotient being obtained by counting the number of times subtraction was necessary to reduce 32 to 0, that is 4.

Subtraction can be performed by use of the addition process, although our computer does not normally operate in this mode. The computer, however, will demonstrate the process, so an explanation is called for here. Two methods can be used, known as the 1s complement method and the 2s complement method.

First the 1s complement method. Consider $1011 - 0101$. First, it is necessary to form the 1s complement of the subtrahend 0101 . This is done by changing all the 1s to 0s and all the 0s to 1s, to give 1010 . To complete the subtraction we now add the two numbers and perform the "end around carry" operation:—

$$\begin{array}{r} 1011 \\ + 1010 \\ \hline 10101 \\ + \xrightarrow{1} 1 \quad (\text{end around carry}) \\ \hline 0110 \quad (\text{sum, and result of subtraction}) \\ 1 \end{array}$$

We have used a four bit word and a carry is generated that exceeds our word length. This carry is added to the result of the first addition—hence "end around carry". Thus $1011 - 0101 = 0110$.

In the 2s complement method, first the 2s complement of the subtrahend is formed. This is equal to the 1s complement + 1. Using the previous example again ($1011 - 0101$), the 2s complement of $0101 = 1010 + 1 = 1011$. Then:

$$\begin{array}{r} 2^4 2^3 2^2 2^1 2^0 \\ 1011 \\ + 1011 \\ \hline 10110 \\ 1 \quad 1 \quad 1 \end{array}$$

but here the carry generated in the 2^4 position is ignored.

Our computer forms the 1s complement of a number by adding to it a series of 1s and ignoring any carry that may be generated.

Thus to form the 1s complement of 0101 :—

$$\begin{array}{r} 0101 \\ + 1111 \\ \hline 1010 \\ \cancel{1} \quad \cancel{1} \quad (\text{ignore carries}) \end{array}$$

Other processes in binary arithmetic, such as operations with positive and negative numbers, will be discussed in the section dealing with programming the computer.

REMINDER ON BOOLEAN SYMBOLS AND LOGIC ELEMENTS

Boolean symbols do not represent quantities but logical states or conditions. For example, the symbols A and B could represent two switches in the "on" position, while \bar{A} and \bar{B} could represent the same two switches in the "off" position. If these switches were connected in series with a battery and lamp, when both switches were "on" this condition would be symbolized as "A and B", and it would result in the lamp being lit. If any other condition existed, i.e. A and \bar{B} (A on, B off), or \bar{A} and B (A off, B on) or \bar{A} and \bar{B} (both off), the lamp would not light. This is a demonstration of the Boolean AND function—the lamp is lit only when the condition A AND B obtains. If the condition of the lamp being lit is represented by the symbol C then it can be said that

$$A \text{ and } B = C \quad \text{or } A \cdot B = C \quad \text{or } AB = C$$

These three statements are identical; the two letters with a dot between, or close together, are shorthand for the AND

function. The condition \bar{A} is read as NOT A. Proceeding further with our two switches and lamp we can therefore say that

$$AB = C \quad \bar{A}\bar{B} = \bar{C} \quad \bar{A}B = \bar{C}$$

where \bar{C} (NOT C) indicates that the lamp is not lit.

If the switches were connected in parallel then either switch being "on" would result in the lamp being lit. Here it is true to say that the condition "A or B" would result in C. The Boolean symbol for this function is +, which is read as OR. With the switches connected in parallel the following equations are true:—

$$\begin{array}{l} A + B = C \quad (\text{A OR B} = C) \\ \bar{A}\bar{B} = \bar{C} \quad (\text{NOT A and NOT B} = \text{NOT C}) \end{array}$$

In the above example \bar{A} (NOT A) was represented by the absence of something (absence of conduction path), but it is important to remember that NOT A really means the presence

of something other than A —in its opposite state \bar{A} . Thus A and \bar{A} can be represented by any pair of defined electrical states: $+6V$ and $+2V$, $0V$ and $-4V$ and so on.

Logic elements.—The simplest of the circuits performing the above Boolean functions is the NOT gate. With this, for example, state A can be negated into \bar{A} or *vice versa*, and the graphical symbol for such an element is shown in Fig. 2. If the input of this element goes to earth the output will go to some voltage above earth. The $\bar{1}$ in the centre of the circle indicates that the element will negate one input, while the arrow shows the direction of information flow.

The time has come to introduce one piece of terminology that will be used throughout the series of articles. The fact that a voltage exists at a particular point in a circuit can be represented by the terms "true", "up", 1 (binary), $-ve$, $+ve$ and so on, and the fact that a voltage does not exist can be represented by "false", "down", 0 (binary), $0V$, etc. The use of 1s and 0s to define voltage levels has been rejected, as far as this series is concerned, for fear that they may become confused with binary 1s and 0s, i.e. numerical data. The term "up" will be used to indicate that a voltage is present and "down" to indicate that that line is at earth potential.

An extension of the NOT gate is the NOR gate. Electronically they are almost identical but the NOR gate has more inputs, as shown by the symbol in Fig. 2. In the logic element used in the computer any input going "up" will result in the output going "down" and for the output to be "up" all inputs must be "down". More or fewer inputs may be provided.

Next in Fig. 2 is shown the symbol for an element performing the previously discussed OR function. In the practical

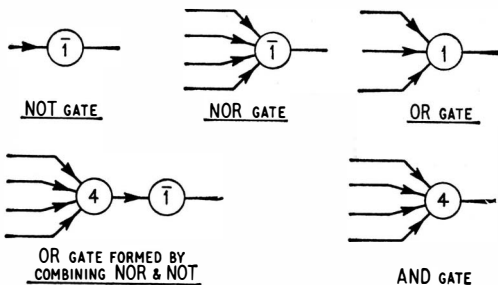


Fig. 2. Symbols for basic logic elements used in the computer.

device any one of the inputs going "up" will result in the output going "up" as indicated by the "1" in the symbol, and the output will be "down" only when all the inputs are "down".

An OR gate can be formed by the combination of a NOR and a NOT gate as shown in Fig. 2, and in fact this method is used in certain parts of the computer. Any "up" input to the NOR gate will cause its output to go "down", and the resulting "down" input applied to the NOT gate will cause its output to go "up".

The final symbol shown is for an AND gate, the "4" indicating that all four inputs have to be "up" simultaneously before the output will go "up", and (switches-in-series example in Boolean algebra section). More or fewer inputs can be provided, within limits, as required.

www.keith-snook.info

THE BASIC CIRCUITS

In the circuit diagrams Figs. 3-12 an indication is given in the captions of the cost of each circuit block, based on the prices in existence at the time of writing for quantity buying. This, however, is only intended as a rough guide. Components only, and not mounting boards, wire, solder, etc., are taken into account.

The computer employs negative logic, that is, an "up" signal is represented by the presence of a negative voltage. It is very important to note, especially when bistables and flip-flops are discussed, that a change of signal level from "down" to "up" is a negative-going voltage change, and a change of signal level from "up" to "down" is a positive-going voltage change.

The circuit of the NOT gate is shown in Fig. 3. This is an exception to one of the rules explained in the "reminder" section, in that a "down" signal can be represented by an open circuit—a feature that is used to advantage in the computer's order decoder. When the input is earthed the base of $Tr1$ is held at a positive potential by virtue of the potential divider between the $+V$ supply and earth ($0V$) formed by R_1 and R_2 . $Tr1$ will therefore be switched off and its collector held at a negative potential. When the input goes "up", $Tr1$ base goes negative, and $Tr1$ switches on and connects anything coupled to the output effectively to earth.

The NOR circuit, shown in Fig. 4, can be seen to be almost identical to the NOT circuit except that more inputs are provided. The operation is exactly the same.

The circuit of an AND gate is shown in Fig. 5. When none of the inputs is "up" all the inputs are connected to earth ($0V$ line), hence all the input diodes ($D1-D4$) are forward-biased by virtue of R_1 . As a result the left-hand side of R_2 is connected to earth, and the right-hand side of R_2 and the base of $Tr1$ are held at a positive poten-

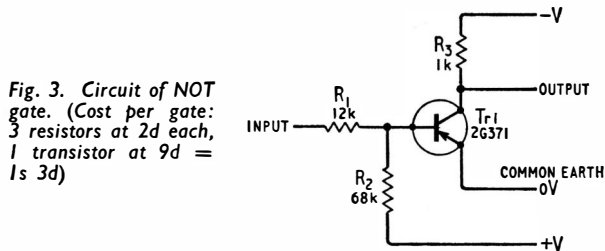


Fig. 3. Circuit of NOT gate. (Cost per gate: 3 resistors at 2d each, 1 transistor at 9d = 1s 3d)

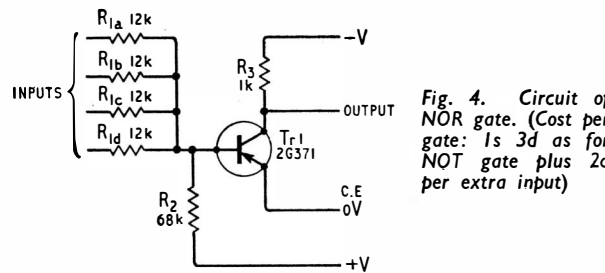


Fig. 4. Circuit of NOR gate. (Cost per gate: 1s 3d as for NOT gate plus 2d per extra input)

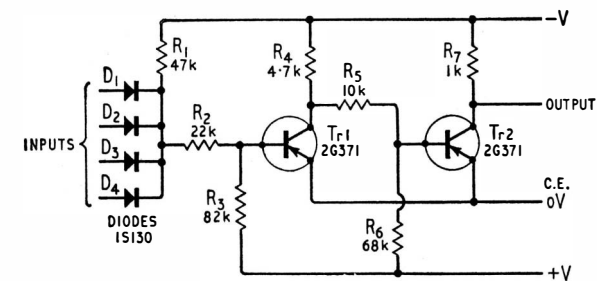


Fig. 5. Circuit of AND gate. (Cost per gate: 7 resistors at 2d each, 4 diodes at 4½d each, 2 transistors at 9d each = 4s 2d)

tial by means of R_3 . Tr1 is cut off and its collector is negative. This negative potential is felt at the base of Tr2 which is held in the conducting state and the output is therefore "down." An input going "up" results in its associated diode becoming reverse-biased; this, however, has no effect on the circuit as the other three input diodes remain forward-biased and Tr1 base is still held positive. When all the inputs are present, however, Tr1 base goes negative by virtue of the potential divider formed by R_1 ,

R_2 and R_3 between the negative and positive supply lines. Tr1 switches on and its collector falls to almost 0V. Tr2 base goes positive, because R_3 is connected to the positive rail, and the output goes "up" as Tr2 switches off.

In certain parts of the computer, as has already been stated, a "down" signal can be an open circuit. In these cases the OR circuit shown in Fig. 6 is used. In all other cases the OR function is performed by a combination of the NOR and NOT gates as described in the "reminder" section on logic elements.

To enable the computer to multiply, as will be seen, it is necessary to compare the states of two circuits, a and b . We will call the outputs of these circuits (indicating their states) A, \bar{A} and B, \bar{B} . A comparator is required for this purpose, and its output must be "up" when one of two conditions exists: A and B present or \bar{A} and \bar{B} present. The output must be down when the conditions $\bar{A}B$ or $A\bar{B}$ exist. There is a large number of possible ways of performing this operation, the logical layout used in the computer being shown in Fig. 7. The numbers written outside the gates are their circuit reference numbers, and this method of identification will be used throughout the computer description. The comparator is made up from two AND gates, one NOR gate and one NOT gate, the NOR and NOT gates forming an OR gate. When AB are present AND gate 4 will open and the output of NOT 12 will be "up." When $\bar{A}\bar{B}$ are present AND gate 5 will open and the output of NOT 12 will again be "up." For any other combination of A and B the output of NOT gate 12 will remain "down."

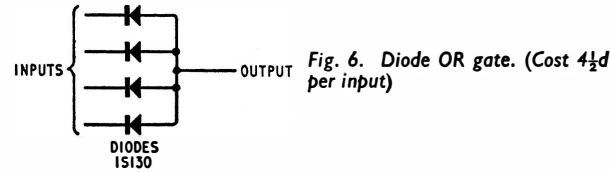


Fig. 7. Logical diagram of comparator.

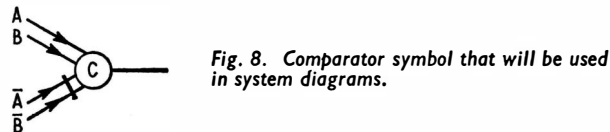
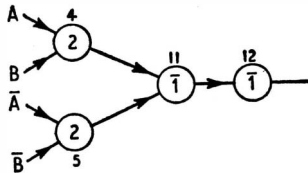


Fig. 8. Comparator symbol that will be used in system diagrams.

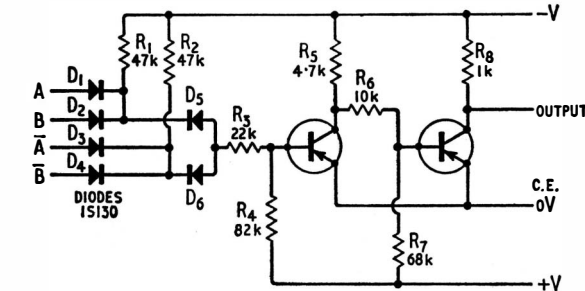


Fig. 9. Comparator circuit. (Cost per comparator: 8 resistors at 2d each, 6 diodes at 4 1/2 d each, 2 transistors at 9d each = 5s 1d)

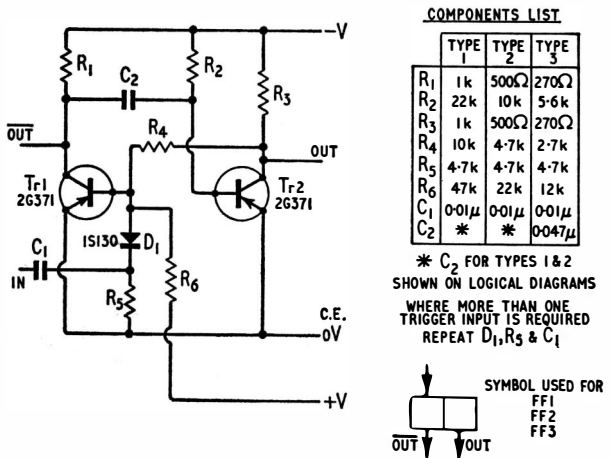


Fig. 10. Flip-flop (monostable) circuit and symbol (Cost per flip-flop: 6 resistors at 2d each, 2 capacitors at 4d each, 2 transistors at 9d each, 1 diode at 4 1/2 d = 3s 6d)

To economise on components the comparator is built as a single circuit and is depicted by the symbol in Fig. 8. The $\bar{A}\bar{B}$ inputs having a line drawn through them. The circuit is shown in Fig. 9. Here, assume that inputs A and B are "up" (\bar{A} and B being therefore "down") D_1 and D_4 will be reverse-biased by these input signals and D_2 and D_3 will be forward-biased by R_1 and R_2 . As a result the lower ends of R_1 and R_2 will be connected to earth, together with the left-hand side of R_3 via D_5 and D_6 . The base of Tr1 will be positive and the remainder of the conditions that exist will be as for the previously described AND gate. If the input condition changes to, say, A and B up, D_1 and D_2 become reverse biased, Tr1 base goes negative via R_1 and, as for the AND gate, the output goes "up." While this condition exists D_6 becomes reverse-biased, preventing the input circuits from interfering with one another. The action for the input $\bar{A}B$ is similar, R_2 providing the drive for Tr1 and D_5 becoming reverse-biased.

Three types of flip-flop are used, the actual choice being determined by circuit requirements. The circuit, together with the symbol used in each case, is shown in Fig. 10.

The flip-flop provides a convenient means of introducing a time delay and obtaining reasonably shaped pulses of known width. In its stable state Tr2 will be held switched on by R_2 , Tr2 collector will be at almost 0V and the base of Tr1 will be positive because of R_6 being connected to the positive line. The flip-flop will remain in this condition until a negative-going edge is applied to C_1 (a signal change of "down" to "up"). This will switch Tr1 on, resulting in a positive-going change at Tr1 collector, and this is felt at Tr2 base via C_2 . Tr2 switches off and Tr1 is held on by R_4 , Tr2 collector now being negative. The flip-flop will remain in this state for a length of time determined by the time constant of $C_2 R_2$, and when C_2 has discharged the flip-flop will regeneratively switch back to its original condition.

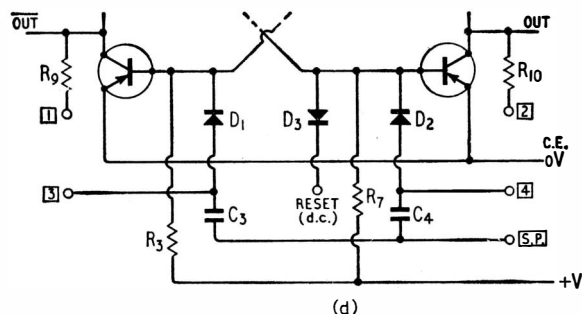
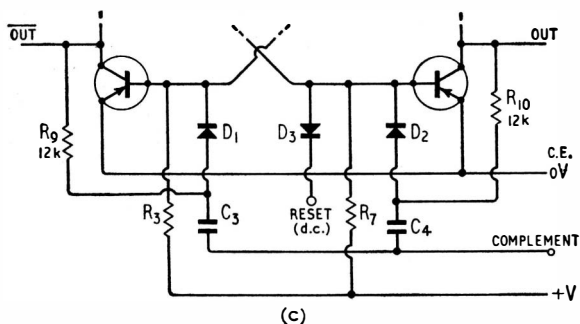
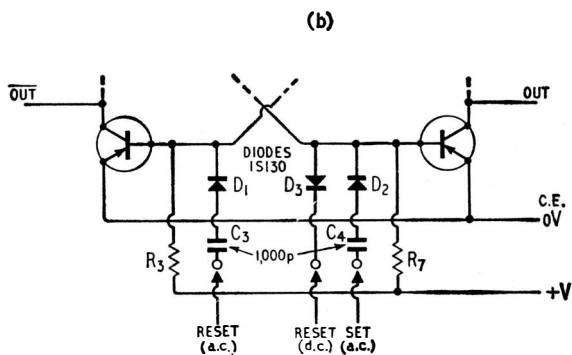
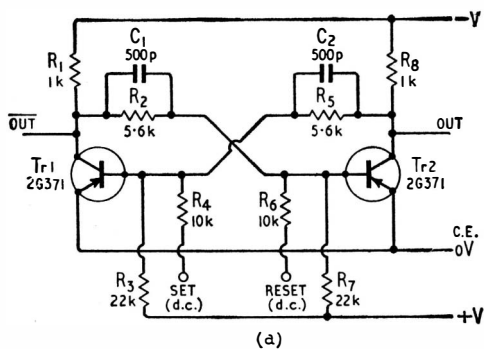


Fig. 11. (a) Set/reset d.c. bistable circuit (Cost per d.c. bistable: 8 resistors at $2d$ each, 2 transistors at $9d$ each = $3s\ 6d$). At (b), modifications to (a) to produce set/reset a.c. bistable. (Cost per a.c. bistable = $4s\ 11\frac{1}{2}d$). At (c), modifications to (a) to produce counter bistable. (Cost per counter bistable: $5s\ 3\frac{1}{2}d$). At (d), modifications to (a) to produce shift register bistable. (Cost $5s\ 3\frac{1}{2}d$)

A further note regarding terminology is in order here, the flip-flop has two outputs which are labelled OUT and OUT respectively. In the text they will be referred to respectively as the NOT output and the output. In the stable state of the circuit the output is "down" and the NOT output is "up." When triggered the reverse is true.

The bistable is used throughout the computer in large numbers. The basic circuit used is the same in all cases, though quite a number of variations occur. The actual type of bistable being used can be deduced from the number and nature of the inputs. However, four main basic types emerge, these being the set/reset d.c. bistable, the set/reset a.c. bistable, the counter bistable and the shift-register bistable. The circuits of all four types are shown in Fig. 11 (a) (b) (c) and (d).

The set/reset d.c. bistable (a) will be described first. When power is applied to the circuit it will assume a state determined by the various component tolerances. Let us assume that Tr1 switches on. Tr1 collector will be at nearly 0V so Tr2 base will be positive (R_2). The collector of Tr2 will be negative as will the base of Tr1 (R_5), holding Tr1 in the on condition. The circuit will remain in this state until something is done to disturb it. If a short negative pulse is applied to the reset terminal, this switches Tr2 on, and, by virtue of the cross coupling resistors, R_2 and R_5 , Tr1 switches off. The switching action is a regenerative one in that the voltage changes at each collector are felt at the opposite base where they are in the right direction to assist the switching. Capacitors C_1 and C_2 are commutating or speed-up capacitors that reduce the switching time. When a pulse is applied to the reset terminal the output does "down" and the NOT

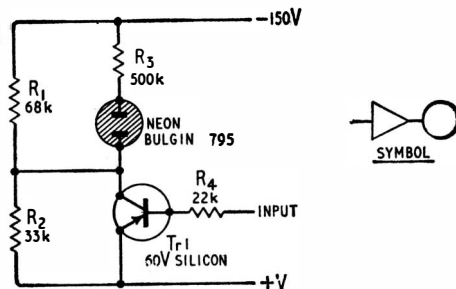


Fig. 12. Neon indicator stage and symbol. (Cost of stage: 4 resistors at $2d$ each, 1 transistor at $2s\ 0d$ and 1 neon at $2s\ 9d$ = $5s\ 5d$)

output goes "up." When a pulse is applied to the set terminal the output goes "up" and the NOT output goes down.

The set/reset a.c. bistable (b) is very similar except that the trigger pulses are a.c. coupled as shown. Here a positive pulse is used, or a positive-going voltage change ("up" to "down"). This is applied to the base of the transistor that is switched on, the positive edge turning it off and reversing the state of the circuit. Because of this the set/reset connections are transposed. An additional reset input is provided which is d.c. coupled, and this enables the starting condition of the bistable to be established. In some cases it is necessary to provide more than one set or reset input; this is achieved by duplicating D1 and C_3 , or D2 and C_1 , as required.

The counter bistable (c) is an a.c. coupled bistable with a gating facility. The starting condition is established

by applying a negative pulse to the reset terminal. Tr1 will now be switched off, its collector will be negative and its base positive. These potentials are applied to D1 in such a way as to reverse-bias it. D2 is, however, forward biased. If now a positive pulse is applied to the "complement" terminal, D2 can conduct, but the reverse-biased D1 cannot. In conducting D2 applies a positive pulse to the switched-on Tr2, turning it off and reversing the condition of the bistable. Now the state of affairs has changed and D1 is forward-biased and D2 reverse-biased, so a subsequent pulse applied to the "complement" terminal will be steered by the diodes to Tr1 base to turn it off. From this it can be seen that each positive pulse applied to the "complement" input will reverse the state of the bistable.

The shift register bistable has no additional components, R₉ and R₁₀ are not connected to D1 and D2

but are brought out as outputs. The complement terminal has been relabelled S.P. This bistable will be dealt with fully in the appropriate section of the computer description.

Indication of the states in various parts of the computer is provided by neon lamps driven by 60V silicon transistors (Fig. 12). A fixed potential is applied across the neon and R₃ that is below the striking voltage, and another voltage, within the transistor rating, is applied across the transistor. Both voltages are supplied by the potential divider R₁ R₂. When a logical "up" signal is applied to R₄ the transistor switches on and the neon strikes and remains in this condition until the "up" signal is removed.

www.keith-snook.info

(Next month we will begin to consider the overall system of the computer.)

ELECTRONIC MUSIC

IN *Mixtur* the sounds of a woodwind ensemble, a brass ensemble, and two string ensembles—one pizzicato—(seated in four groups around the audience) are picked up by microphones and put into four ring modulators; the four groups of microphones lead to four mixing tables, where sound engineers control the balance of the various microphones and the input levels for the ring modulators (during the last public performance of *Mixtur* at Stockholm in October, 1966, a total of 36 microphones was used, one microphone for each stand with two musicians). Four players, each using a sine wave oscillator with continuous frequency control, produce sine waves with which the instrumental sounds are modulated by the ring modulators. The results, reproduced over four separate loudspeakers, are blended with the orchestral sound. From each instrumental sound there arises a *Mixtur*-sound. (By 'Mixtur', one usually refers, regarding organ stops and also choral and orchestral melodies, to a mixing of parallel pitches. It is then a matter of timbral texture from overtones or parallels of chromatic intervals.) The fifth instrumental group of *Mixtur*, consisting of three percussionists each playing a cymbal and gong, is provided with contact microphones connected to three separate loudspeakers. So a composition of differentiated timbres—which I had heretofore been able to achieve only with electronically produced sounds—becomes possible with the use of instruments."

So writes the composer Karlheinz Stockhausen in the first issue of a new quarterly journal *Electronic Music Review*. Established "to provide a source of information and a means of discourse on all aspects of electronic music," this American publication contains a mixture of semi-technical articles describing equipment and general articles on organizational and aesthetic matters. "The multiplier-type ring modulator" shows how various audio signals, produced from conventional musical sources and from artificial sources such as oscillators and noise generators, may be modulated by each other to obtain various effects; while a group of articles on "programmed control" discusses methods by which sources may be controlled in pitch, amplitude, bandwidth, duration, etc., using punched tape, sequencing devices and computers. As one of the authors says, "the variety of results obtainable is as limitless as the imagin-

ation of the user." It will be interesting to see whether the use that is made of this superabundance of technical means in the years to come will refute the widely held belief that art thrives on restriction of means—or support it!

Electronic Music Review is published by Independent Electronic Music Center Inc., Trumansburg, N.Y.14886, U.S.A.

Conferences and Exhibitions

EDINBURGH

Aug. 5-10

Information Processing Congress

(International Federation for Information Processing, 23 Dorset Sq., London, N.W.1)

LEEDS

Aug. 30-Sept. 6

British Association Annual Meeting

(British Association for the Advancement of Science, 20 Great Smith St., London, S.W.1)

OVERSEAS

Aug. 13-17

Energy Conversion Engineering Conference

(M. Altman, Univ. of Pennsylvania, 113 Towne Bldg., Philadelphia, Pa.)

Aug. 25-Sept. 3

German Radio Exhibition

(Berlin Ausstellungen, 22 Messedamm, 1 Berlin 19)

Aug. 28-Sept. 2

Computation Congress

(International Association for Analogue Computation, Wasserwerkstrasse 53, CH 8006 Zurich)

Aug. 29-31

High Frequency Generation and Amplification

(H.F. Conference Committee, School of Electrical Engineering, Phillips Hall, Cornell University, Ithaca)

Miami Beach

Berlin

Lausanne

Ithaca, N.Y.