DECEMBER 1951

RADIO-ELECTRONICS

LATEST IN TELEVISION • SERVICING • AUDIO

HUGO GERNSBACK, Editor

LIGHT-SENSITIVE ELECTRONIC BEAST

SEE ELECTRONICS

30¢ U.S. and CANADA

In this issue: **Measuring Distortion • Scotsman's Super • New Life for Old Radios**

Some important lessons for radiomen are carried by this

Light Sensitive . . . ELECTRONIC BEAST

By EDMUND C. BERKELEY

On the cover of this issue of Radio-Electronics is a picture of a small robot which has four sensing organs, three acting organs, and a small electronic and relay brain. His name is Squee, the Robot Squirrel. What he does is roll along the floor, hunt for "nuts," pick up a "nut" in his scoop, take it over to his "nest," leave it there, and then hunt for more nuts.

Although Squee is not a very clever robot, he does have a small amount of memory and of reasoning ability, and he is a close cousin of his predecessor, Simon, the Midget Electric Brain. Simon was the main subject of a series of thirteen articles in Radio-Electronics from October, 1950, to October, 1951, by Robert A. Jensen and this author.

There are a number of interesting things about Squee, himself, but the most important of them is that he is in many ways a good illustration of a powerful new method for the design of circuits for mechanical brains and robots. This method is the *algebra of logic*, also called *Boolean algebra*. The engineers at Northrop Aircraft Co. in California, who designed the electric brain Maddida, say they have given up drawing circuit diagrams in many places because Boolean algebra does a better job.

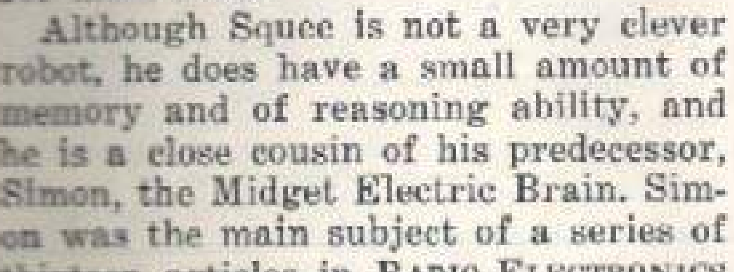


Fig. 1—The electronic sensory organs of Squee.

In this series of two articles the main emphasis will be on Boolean algebra: what it is, how you can calculate with it, and how it can be used in practice. The secondary emphasis will be on Squee. But first a few more words about Squee.

Why did Edmund C. Berkeley and Associates build Squee? Well, last year Bob Jensen and I read some articles about a mechanical tortoise made in Bristol, England, by Dr. W. Grey Walter at the Burden Neurological Institute. We said to ourselves, "Let's make a robot like that—but have him do something a little more clever." So we came up with the idea of a squirrel gathering nuts, and decided to make a robot squirrel.

Squee was constructed mainly through the efforts of three men—Robert A. Jensen (until he re-entered the Air Force in June, 1951), William Szabo, and Jack Koff. Bob Jensen made the skeleton, a framework holding a front wheel for driving, a pivoted column holding and steering the wheel, and two rear wheels rolling free. He mounted tubes, relays, and batteries. By October, 1950, the machine was responsive to light (but only to one kind of light). It still had no "hands." In March, 1951, we made a commitment to exhibit Simon and Squee at the Minnesota State Fair, August 25 to September 3, under the auspices of The Dayton Co., a large department store of Minneapolis. The project that had started as fun became good business. We undertook in earnest the work of making the machine sensitive to a second kind of light, and to give it the needed hands. The scheme for the scoop and the nest light was the main contribution of Szabo, and the final completion, testing, and modi-

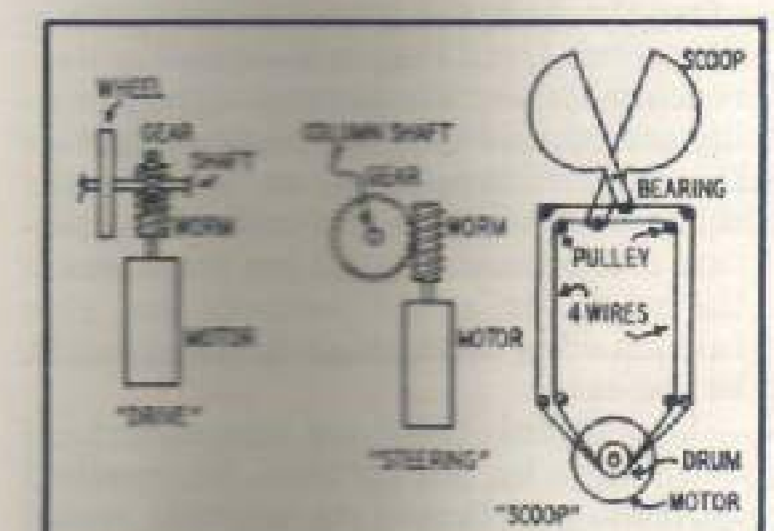


Fig. 2—These are the squirrel's motor organs.

fication of the machine was due to Jack Koff. Squee was exhibited ten hours a day for ten days in Minneapolis, and probably 50,000 people saw him.

The design of Squee

The physical design of Squee is based on hitching together the sensing organs (see Fig. 1) and the acting organs (see Fig. 2) with appropriate hardware. The logical design of Squee is based on the Boolean algebra relating the conditions expressed by the sensing organs and the conditions expressed by the acting organs.

The first two sensing organs are the *right eye* and the *left eye*, the two photocells. They enable Squee to scan the surrounding environment; as Squee turns his steering column, the photocells look in one direction after another for nuts or nest. A nut (currently a golf ball) is lighted from above by a steady light, a d.c. light. The nest (currently a 12 x 18-inch sheet of aluminum) is lighted by a 60-cycle a.c. gas-filled lamp giving 120 flickers a second.

In Squee the physical circuit using electronic tubes connected to each photocell reports at any time three logical conditions. These are: darkness; a.c. light; and either d.c. or a.c. light or both a.c. and d.c. light. The possible logical reports from the circuit, for each phototube are:

Condition	*Darkness*	*A.C.*	*Any light*
No.	*Report*	*Report*	*Report*
1	1	0	0
2	0	1	1
3	0	0	1

Here the 1 designates "yes" or "reported" or "on," and the 0 denotes "no" or "not reported" or "off." Notice particularly that this circuit, which we called the Amplifying Circuit, was unable to report "d.c. and not a.c."; there will be a lot to say about this point later.

The third sensing organ of Squee is a contact-reporting switch taken from a vending machine, and installed at the base of the scoop. We called this the "tongue." When the nut (ball) entered the scoop, it would roll against this contact switch and close a relay, thus telling Squee that it had taken hold of a nut.

The fourth sensing organ of Squee is a "foot," consisting of two copper tips mounted on springs, which trail along the floor. If and when both of them touch a metal plate (the "nest"), a relay is closed, and Squee "knows" that it has found the nest.

The possible logical reports from these two sensing organs are:

Tongue Report	*Foot Report*
1	1
0	0

We come now to the acting organs. After a lot of pondering over various ways of giving energy to the acting organs, and the problems of clutches, we finally decided on the simplest, though crudest method: We hitched a separate motor to each part that had to be moved, and we provided that it could be de-energized, run forward or backward.

For the drive wheel, we mounted a gear on the drive shaft, and turned that gear with a worm wheel mounted on the shaft of the drive motor. For steering, we mounted a gear on the column shaft, and turned the gear with a worm wheel mounted on the shaft of the steering motor. In the case of the scoop, we had a problem. There was room to put a motor at the bottom rear of the chassis. But the scoop was at the front of the chassis, even ahead of the column, and it had to be opened and closed like two cupped hands held together at the wrists. So we ran pulley lines made of light, flexible wire string, from the base of the scoop to the drum mounted on the shaft of the scoop motor; and we adjusted the amount of turning of the motor by means of limit switches, so that there would be two positions and the scoop would be either open or closed.

The possible logical reports about each of these three acting organs is:

Condition	*Motor On*	*Motor Forward*	*Motor Backward*
1	0	1 or 0	0 or 1
2	1	1	0
3	1	0	1

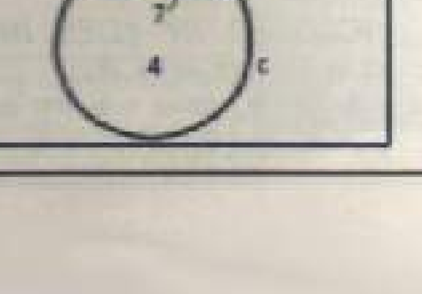
We have now reduced the sensing (or input) of Squee to a set of yes's and no's, or 1's and 0's. We have reduced the acting (or output) of Squee to a set of yes's and no's, or 1's and 0's. We now have left the problem of hitching the input and the output together, so as to express the desired behavior of Squee.

Ordinarily, up to this time, this kind of problem has been solved by the practical method of drawing circuits on paper, using prior rule-of-thumb experience with that method. But there

CHART 1—THE IDEAS OF BOOLEAN ALGEBRA

ELEMENTARY ALGEBRA	BOOLEAN ALGEBRA
QUESTION 1. What symbols are used to stand for any things being talked about?	
a, b, c, . . ., x, y, . . .	a, b, c, . . ., x, y, . . .
2. What can the things be that are talked about?	
Numbers, like: 4, 8.57, —3, ⅓, $\sqrt{2}$, 3.14159. . ., . . .	(A) Classes, like: "Horses, Animals, Cows, Mammals, . . ." (B) The truth values (i.e., yes, no; or 1, 0) of statements such as: "Motor A is off." "Motor B is on." "Photocell C registers light."
3. What operations are there?	
PLUS: $a + b$ MINUS: $a - b$ TIMES: $a \times b$, ab DIVIDED BY: $a \div b$, a/b ROOT: $\sqrt{a}$, etc.	OR: $a \vee b$ AND: $a \cdot b$, ab NOT: a', $\sim a$ EXCEPT: $a \cdot b'$ OR ELSE: $a \wedge b$
4. What special constants are there?	
ZERO, 0, such that $a + 0 = a$ for every a (and $a \cdot 0 = 0$, $a \neq \infty$) ONE, 1, such that $a \cdot 1 = a$ for every a, $a \neq 0$ INFINITY, ∞, such that $a + \infty = \infty$ for every a (and $a \cdot \infty = \infty$ for $a \neq 0$)	NULL CLASS, 0, such that $a \vee 0 = a$ for every a (and $a \cdot 0 = 0$) UNIVERSE CLASS, 1, such that $a \vee 1 = 1$ for every a (and $a \cdot 1 = a$)
5. How many are all the things that are talked about?	
INFINITY, ∞	2, or 4, or 8, or 16, or . . ., or ∞
6. What is an example of a rule?	
"The reciprocal of the reciprocal of a number is the number itself." $1/(1/a) = a$	"The truth value of the denial of the denial of a statement is the truth value of the statement itself." $(a')' = a$
7. How do you represent graphically the things talked about?	
(A) By points on an infinite line: See drawing on page 48 (B) By tables of numbers:	(A) By areas in a finite rectangle: (See drawing below) The areas No. 1 to 8 are: No. 1: $ab'c'$ " 2: abc' " 3: $a'bc'$ " 4: $a'b'c$ " 5: $ab'c$ " 6: $a'bc$ " 7: abc " 8: $a'b'c'$ The null class has no location. (B) By tables of truth values:

a	b
0	0
1	1
2	4
3	9
4	16
5	25
6	36



a	b	c	d
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

is beginning to be a change: Designers have begun to use the methods of Boolean algebra to effect the logical design of circuits to express behavior. For connecting the input and output yes's and no's, 1's and 0's, is a problem in Boolean algebra. What is Boolean algebra?

Boolean Algebra

Boolean algebra is a kind of algebra named after a great English mathematician, George Boole, who lived 1815 to 1864. He wrote a famous book called *The Laws of Thought*, in which he laid out quite completely the design of a new algebra. It was somewhat like ordinary algebra but was adapted to the ideas and operations of logic, of reasoning.

Other mathematicians and symbolic logicians have since then considerably improved and extended the algebra which Boole devised.

Ideas of Boolean algebra

What are the ideas of Boolean algebra? In Chart 1 is a comparison of the main features of:

Elementary algebra, which we all learn in school, for handling numbers, and which is essential for all computations in radio, electronics, electricity, etc., and

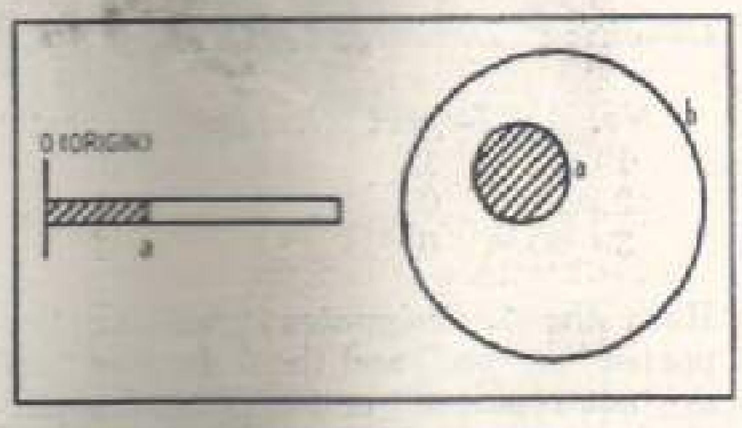
Boolean algebra, the newer algebra, which is useful for handling statements, classes, conditions, and circuit elements.

A great deal of information has been packed into this chart, and it is worth much attention.

A reader may say: "There seem to be two ways in which Boolean algebra can be represented, by classes and by the truth values of statements." Yes—and there are more ways besides. Boolean algebra is an interesting mathematical framework that applies to quite a variety of different situations.

For example, take the number 30 and its factors 1, 2, 3, 5, 6, 10, 15, 30; let a, b, c, . . . be any factors; let $a \vee b$ mean the least common multiple of a and b; let $a \cdot b$ be the highest common factor of a and b; and a' be 30/a. You will find this to be a Boolean algebra.

For another example, consider sets of contacts, switches or relays; let a, b, c, . . . be any switch contacts; $a = b$ if a is closed when b is closed and open when b is open; $a \vee b$ means a and b in parallel; $a \cdot b$ means a and b in series; and a' is any contact open when a is closed and closed when a is open. This is typical Boolean algebra. How it is applied in practical electric circuits will be shown later.



The drawings above indicate what *a* and *b* mean in standard and Boolean algebra.

CHART 2—THE RULES OF BOOLEAN ALGEBRA

TOPIC	ELEMENTARY ALGEBRA	BOOLEAN ALGEBRA
1. Commutative Law	$a + b = b + a$ a PLUS b = b PLUS a $a \cdot b = b \cdot a$ a TIMES b = b TIMES a	$a \vee b = b \vee a$ (a OR b) = (b OR a) $a \cdot b = b \cdot a$ (a AND b) = (b AND a)
2. Associative Law	$(a + b) + c = a + (b + c)$ a PLUS b PLUS c is the same, whatever order you take them. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ a TIMES b TIMES c is the same, whatever order you take them.	$(a \vee b) \vee c = a \vee (b \vee c)$ a OR b OR c is the same, whatever order you take them. $(a \cdot b) \cdot c = a(b \cdot c)$ a AND b AND c is the same, whatever order you take them.
3. Distributive Law	*ONE LAW:* $a \cdot (b + c) = ab + ac$ a TIMES (b PLUS c) = (a TIMES b) PLUS (a TIMES c)	*TWO LAWS:* (1) $a \cdot (b \vee c) = a \cdot b \vee a \cdot c$ a AND (b OR c) = (a AND b) OR (a AND c) (2) $a \vee (b \cdot c) = (a \vee b) \cdot (a \vee c)$ a OR (b AND c) = (a OR b) AND (a OR c)
4. Combining	$a + a = 2a$ a PLUS a = TWO a $a \cdot a = a^2$ a TIMES a = a SQUARED There are numerical coefficients and exponents.	$a \vee a = a$ a OR a = a $a \cdot a = a$ a AND a = a There are *no* numerical coefficients or exponents.
5. Special Elements	0, ZERO; 1, UNITY $a + 0 = a$ $a \cdot 0 = 0$ $a + 1 = a + 1$ $a \cdot 1 = a$	0, NULL, NOTHING; 1, UNIVERSE, ALL $a \vee 0 = a$ (a OR NOTHING = a) $a \cdot 0 = 0$ (WHAT IS BOTH a AND NOTHING = NOTHING) $a \vee 1 = 1$ (a OR ALL = ALL) $a \cdot 1 = a$ (BOTH a AND ALL = a)
6. Opposites	*TWO OPPOSITES:* $a + (-a) = 0$ a PLUS (MINUS a) = 0 $a \times (1/a) = 1$ a TIMES (RECIPROCAL OF a) = 1 $-(-a) = a$ $1/(1/a) = a$	*ONE OPPOSITE:* $a \vee a' = 1$ a OR NOT a = ALL $a \cdot a' = 0$ BOTH a AND NOT a = NOTHING $(a')' = a$ (NOT-NOT-a = a) $1' = 0$ $0' = 1$
7. Laws Involving Opposites	$a + b = -(-a -b)$ $a \cdot b + a(-b) = o$ $-(a + b + c \ldots) = (-a) + (-b) + (-c) + \ldots$	$a \vee b = (a' \cdot b')'$ a OR b = NOT (NOT-a and NOT-b) $a \cdot b = (a' \vee b')'$ $a \cdot b \vee a \cdot b' = a$ $(a \vee b) \cdot (a \vee b') = a$ $(a \vee b \vee c \ldots)' = a' \cdot b' \cdot c' \ldots$
8. Absorption	$a + ab = a(1 + b)$ $a(a + b) = a^2 + ab$	$a \vee (a \cdot b) = a$ a OR (a AND b) = a $a \cdot (a \vee b) = a$ a AND (a OR b) = a
9. Relation of Less Than or Included In	*LESS THAN OR EQUAL:* $a \leqq b$ if and only if $a +$ (zero or some positive number) $= b$	*INCLUDED IN (LIES IN):* $a \subset b$ if and only if: $a \vee b = b$, or $a \cdot b = a$, or $a \cdot b' = 0$, or $a' \vee b = 1$

Rules of Boolean algebra

But the ideas of Boolean algebra, interesting though they may be, are not enough: we also need the rules. We could work out the rules on the basis of ordinary reasoning. In fact, most ordinary reasoning is Boolean algebra, and is done by means of words and experience. But the rules of Boolean algebra expressed in mathematical form are potent and helpful. They are given in Chart 2, set into comparison with the rules of ordinary elementary algebra.

Here then is an introduction to the ideas and rules of Boolean algebra. Some of the ways to use them for dealing with circuit elements that can be "on" or "off," and some of the ways to use them for connecting input and output to express the behavior of a robot or mechanical brain, will be explained in the next article.

—end—

Algebra in Electronic Design

By EDMUND C. BERKELEY

IN OUR previous article, "Light Sensitive Electronic Beast," we talked about Squee and about Boolean algebra, and introduced the ideas and rules of Boolean algebra. We said that Boolean algebra had a number of important applications in the design and simplification of circuits, and that it was used in the design of Squee and proved very useful. Some of the use of Boolean algebra in the design of Squee will be given in this article; but unfortunately there will be no space here to give all the construction information for Squee. However, we shall be glad to try to help any reader who wants to learn about or construct Squee (or other small robots).

What is Boolean algebra?

In the last article we gave three interpretations of Boolean algebra: classes of things; factors of a number such as 30; and relations of contacts of switches or relays, as shown again in Fig. 1 here.

What are a, b, and c in Fig. 1? They are not actually contacts or wires; they stand for states, conditions, or reports about contacts or wires. They stand for reports such as "Current flowing" or "No current flowing." They always have just one out of two values, such as *yes* or *no*, *all* or *nothing*, *true* or *false*, *1* or *0*. They are called *binary variables*, variables which have only two values. Any letter labeling a contact or a wire has the value 1 if its contact is energized or closed or if its wire is carrying current. It has the value 0 if its contact is open or not energized, or if its wire is not carrying current. Contacts are always drawn in the unenergized (double-contact relays) or open (single-contact relays) position. This is referred to as the "not" position and is written a′, b′, etc.

The algebra of 1 and 0

This leads us to a fourth interpretation of Boolean algebra that is most important for our purposes. This is the algebra of propositions or statements. Suppose that P, Q, R stand for propositions or statements such as "Motor A is on," "Photocell B registers light," "Relay C is energized," etc. Let T(...), where the space ... is filled with a statement, stand for the "truth value of ...," equal to 1 if the statement is true and 0 if the statement is false. Then the 1's and 0's of truth values are a Boolean algebra.

We write T(P) = p, T(Q) = q, T(R) = r. In other words, a convenient abbreviation for the truth value of a statement represented by a capital letter is the corresponding small letter.

Now it can easily be shown that the following rules hold:

$$T(P \text{ AND } Q) = p \cdot q$$
$$T(P \text{ OR } Q) = p \vee q = p + q - pq$$
$$T(\text{NOT} - P) = p' = 1 - p$$

In the form of tables, we can list all the cases:

I

p	q	p·q
0	0	0
0	1	0
1	0	0
1	1	1

II

p	q	p v q	p + q − pq
0	0	0	0 + 0 − 0
0	1	1	0 + 1 − 0
1	0	1	1 + 0 − 0
1	1	1	1 + 1 − 1

III

p	p′	1 − p
0	1	1
1	0	0

How do we convince ourselves of these tables and formulas? Let us first ask "When is the statement P AND Q true?" Now we know from our use of AND that this is only true if P is true (p = 1) and Q is true (q = 1). The table shows 1 for p · q only in that case. That is what we mean by AND when we put it between statements.

By OR we ordinarily mean "... OR ... OR BOTH." Sometimes we mean OR ELSE; but in the connection of switch and relay contacts in parallel and often elsewhere, the inclusive-OR is more useful than the exclusive-OR. The table above shows 1 for p v q when one or the other or both of p and q is 1.

By NOT we mean that when P is true, NOT-P is false, and when P is false, NOT-P is true. And this relationship the table of p and p′ accurately summarizes.

As we see in the tables, we can easily write down formulas of ordinary elementary algebra using plus, minus, times, 1, 0, which will do the same work as Boolean formulas. In fact AND and TIMES are indistinguishable. The algebra formulas $p \vee q = p + q - pq$ and $p' = 1 - p$ are interesting and occasionally useful, but most of the time the operators v (or) and ′ (not) of Boolean algebra are more compact and fit more neatly with the expression of circuits.

Interpreting rectifiers

But there are many other types of circuit elements and other mechanisms that have just two states, on and off, closed or open, moved to one side or moved to the other, positioned forward or positioned back. And it is easy, logical, and efficient to represent these two-fold conditions by Boolean algebra also. Each new type of element leads to another interpretation of Boolean algebra.

For example, a fifth interpretation of Boolean algebra is in terms of rectifiers (see Fig. 2). Under the term rectifiers we include vacuum tube diodes, germanium and selenium crystal diodes, etc., any circuit element in which current flows in one direction only.

In Fig. 2, 0 represents "low voltage" and 1 represents "high voltage." In the OR circuit, the output line c will have a high voltage if and only if either a or b or both have a high voltage, because then the potential drop will be all across resistor R. In the AND circuit, the output line c will have a high voltage if and only if both a and b are at a high voltage; for only in that case is there no drop across the resistor.

There is no direct representation of NOT-P using rectifiers; but if the inputs to a rectifier network include all the

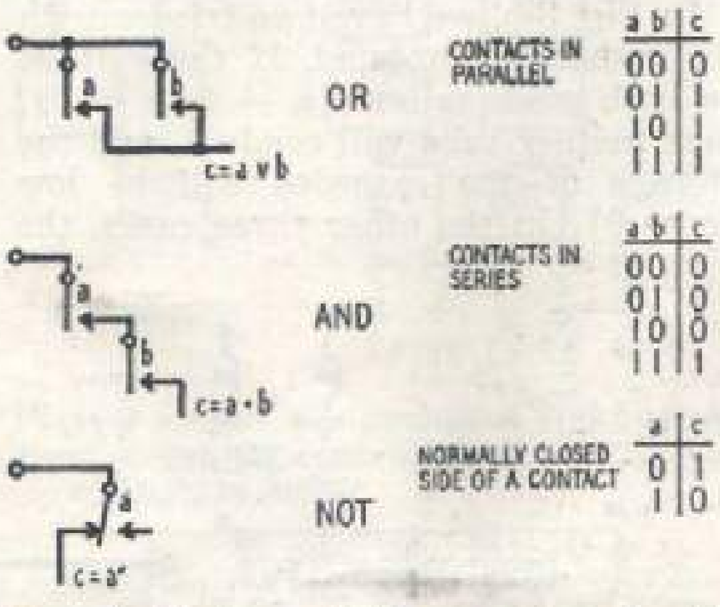


Fig. 1—Basic switch or relay circuit relations expressing Boolean algebra.

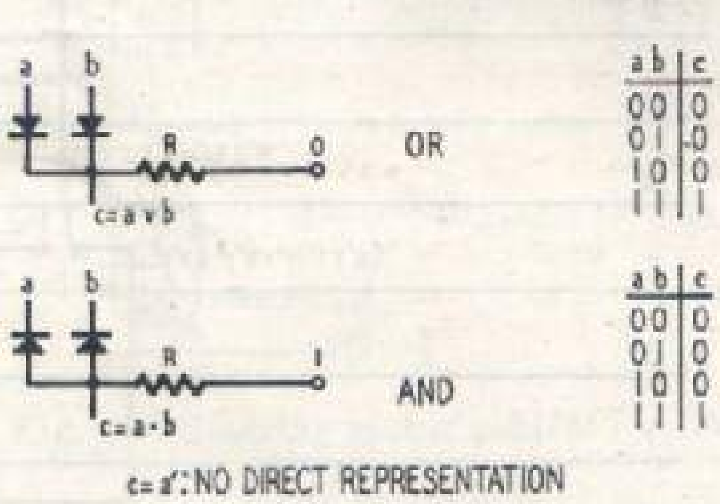


Fig. 2—Boolean relations in rectifiers.

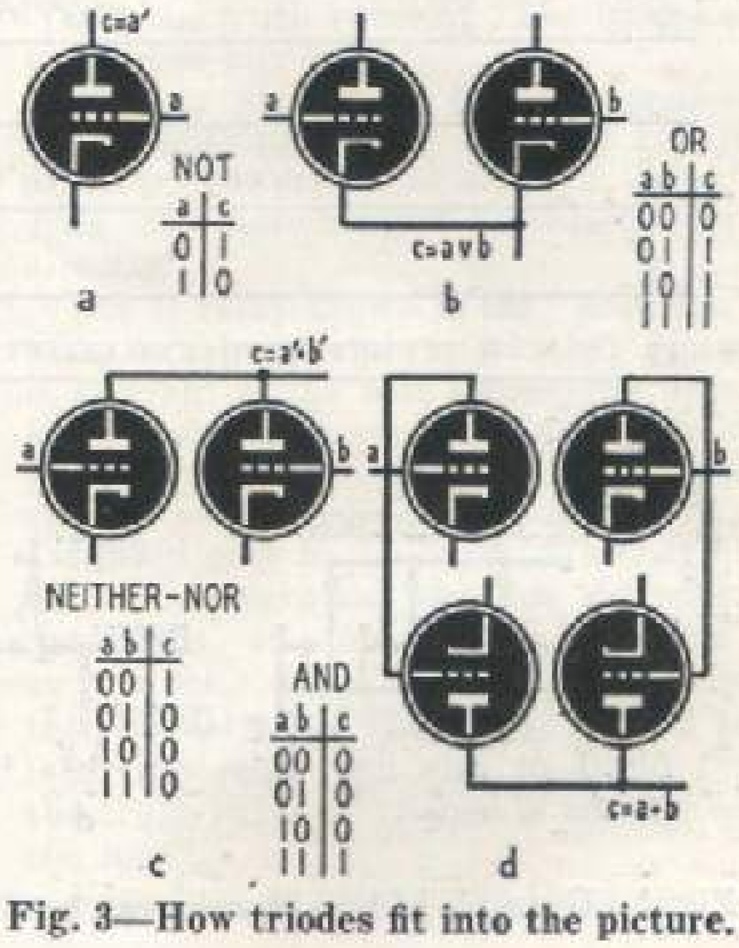


Fig. 3—How triodes fit into the picture.

binary variables needed, a, b, ..., and their individual negatives, a′, b′, ..., then the rectifier network can yield all Boolean functions needed.

A good example of the use of Boolean algebra in a rectifier network to change one set of signals into another set of signals is given in Part 12 of the series "Constructing Electric Brains," by Berkeley and Jensen, appearing in the September, 1951, issue of RADIO-ELECTRONICS, pp. 45-46.

Interpretation of triodes

A sixth interpretation of Boolean algebra is in terms of *triodes* (see Fig. 3). Fig. 3-a is a triode with a signal *a* on the grid, and a signal c on the plate. If the voltage of the grid is high (a = 1), the tube will conduct and the voltage of the plate will be low (c = 0). If the voltage of the grid is low (a = 0), the tube will not conduct and the voltage of the plate will be high (c = 1). The circuit expresses c = a′.

In Fig. 3-b, we have two triodes with the cathodes connected. If the voltage on both grids is low (a = 0, b = 0), then neither tube will conduct, and the voltage of the cathodes will be low (c = 0). In the other three cases, the voltage of the cathodes will be high (c = 1). The circuit expresses c = a v b.

In Fig. 3-c, we have two triodes with the plates connected. The voltage on the plates will be high (c = 1) if and only if the voltage on both grids is low (a = 0, b = 0). The table shows the situation. What is the equation for c?

There is a useful, general rule of Boolean algebra (which we shall call general rule No. 1) that we can use:

1. Suppose we have a complete table of 1's and 0's showing the behavior of some binary variables.
2. Suppose y is the dependent variable and a, b, c, ..., are the independent variables.
3. Note all the cases where y = 1.
4. If any independent variable a = 0, write a′; if any independent variable b = 1, write b.
5. For each case, associate a′, b, ..., with AND.
6. For all the cases, associate them with OR.

For example, suppose we have the following table:

Case	a b c	y
1	0 0 0	0
2	0 0 1	1
3	0 1 0	0
4	0 1 1	0
5	1 0 0	0
6	1 0 1	0
7	1 1 0	1
8	1 1 1	0

Now y = 1 in two cases: case 2 where a = 0, b = 0, c = 1, and case 7 where a = 1, b = 1, c = 0. Applying the rule, y = a′ b′ c v a b c′

The present instance is Fig. 3-C, and we have:

c = a′ • b′ = (not–a and (not–b)

In ordinary English, c here equals "neither a nor b."

How do we connect triodes to get a AND b? In Fig. 3-d, a manner of connection is shown. The outputs of the two upper triodes have to be negated in the two lower triodes (plate-connected) in order to give c = a • b. Obviously, therefore, if we are using triodes and want to economize, we should prefer to work with OR and NEITHER-NOR relations, instead of OR and AND relations.

Enough has been said, perhaps, to show that many different kinds of circuit elements may be used to express AND, OR, NOT, EXCEPT, and other relations of Boolean algebra. For example, pentodes could be used to represent relations of Boolean algebra. But how can Boolean algebra simplify circuits?

Let us now take an example of a circuit and its simplification using Boolean algebra.

Suppose we have the circuit shown in Fig. 4-b, which energizes relay Z by means of contacts of relays W, X, and Y shown in 4-a. Our problem is to simplify this circuit.

Looking at 4-b, we see seven (two-way) relay contacts in this circuit, and six wire connections. This makes a total of thirteen *events* that modify information carried in the wires of the circuit. Wherever there is an *event* of this type, the effect on the information is a Boolean algebra operation, as may be seen in the isolated examples shown in Fig. 4-d, 4-e, and 4-f. In either the forking contact of 4-d or the associating contact of 4-e, change in the position of the armature may change the information in the output. In the junction of Fig. 4-f, changing the input from one conductor to the other is the "event" which changes the information in the output.

So we go back to 4-b, and draw blue lines across the circuit, in such a way as to isolate each event, from No. 1 to No. 13. Now we go down through the circuit, calculating the information which is in each wire of the network. Event 1 is the contact w; hence the right-hand output wire contains w (contact actuated), and the left-hand one not-w, or w′ (contact not actuated). Event 2 is the contact x. Hence the right-hand output wire contains wx and the left-hand one wx′. Event 3 is a join. Hence the output wire contains w′ v wx′, which reduces to w′ v x′, by Boolean algebra. This modifies the information

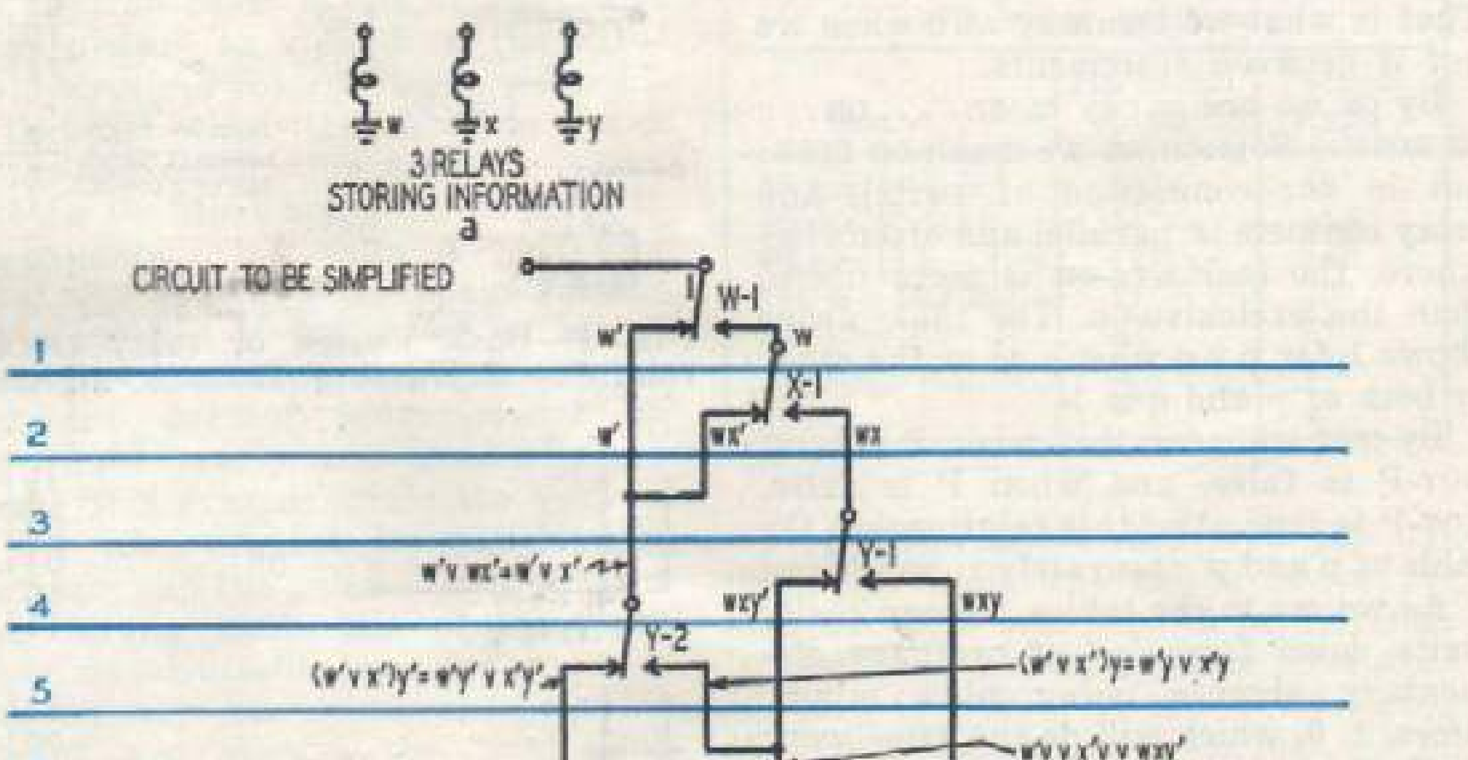


Fig. 4—Example of how circuits can be simplified with the help of Boolean algebra.

in the incoming wires, but in this present type of case, where we want to calculate what information energizes the relay W at the bottom of the diagram, it is not necessary to go back and modify the descriptions of information in the earlier wires. There are, however, circuits where this step is necessary—this is the problem of "back-circuits."

And so we may go through the whole network, at each event computing the information in each output wire, and finally after event 13 reach the wire which energizes relay Z. The expression for the information in this wire is

$$w'y \mathrm{\ v\ } wxy' \mathrm{\ v\ } x' \mathrm{\ v\ } wxy$$

which simplifies into

$$w \mathrm{\ v\ } x' \mathrm{\ v\ } y.$$

This expression of course represents a very simple circuit, of three contacts in parallel, and is shown in Fig. 4-c. The circuit of 4-b (provided the contacts have the functions expressed by the labels) reduces to the circuit of 4-c.

The process we have illustrated here is a powerful, general method, and can be applied in many kinds of situations.

Example of circuit design

Boolean algebra may be used in more ways than just simplifying circuits. It may for example be used in the design of circuits, because it can express the "yes-no" elements in the word language describing the problem, just as well as it can express the yes-no elements in the electrical language describing the mechanism.

For an example, let's take the problem of designing the circuits for controlling the steering of Squee.

a. Associating sensations and behavior

We can work through this problem in stages. The first stage is associating the sensations of Squee with Squee's behavior.

The sensations of Squee for steering purposes are those that are derived from the two phototubes, the "right eye" and the "left eye." In this first stage of the problem, we shall ignore distinctions about kind of light, a.c. or d.c. So, let R equal the truth value of "Squee's right eye sees light," and let L equal the truth value of "Squee's left eye sees light."

The behavior of Squee for steering purposes consists of three states: steering clockwise, steering counterclockwise, and no steering at all. Let:

C equal the truth value of "Squee is steering clockwise,"

U equal the truth value of "Squee is steering counterclockwise," and

N equal the truth value of "Squee is not steering."

Squee must choose between these three kinds of behavior depending on Squee's sensations. How do we associate behavior with sensations? In Fig. 5 we see the various cases displayed; and we can see what we want to arrange. This is summarized in the following table:

Case	R	L	C	U	N
1	0	0	1	0	0
2	0	1	0	1	0
3	1	0	1	0	0
4	1	1	0	0	1

Using our general rule No. 1, we have:

$C = R'L' \mathrm{\ v\ } RL'$ which reduces to L',
$U = R'L$, and
$N = RL$.

Translating these equations into words: Squee should steer clockwise when the left eye does not see light; Squee should steer counterclockwise when the left eye sees light but the right eye does not see light; and Squee should not steer at all when both eyes see light.

Suppose that we had relays corresponding to C steering clockwise, U steering counterclockwise, and N no steering, the schematic for the behavior of Squee would be as in Fig. 6.

b. Associating behavior and action

But Squee does not have "acting organs" corresponding directly to the three states of behavior. Instead, Squee has a steering motor, whose normal direction is such that Squee steers clockwise, and a relay, by means of which the motor may be run in either direction. See Fig. 7. So, let X equal the truth value of "The motor is running" and let W equal the truth value of "The reversing relay is energized."

How do we associate actions X and W with behavior C, U, N? The logical association is shown in the following table:

R	L	C	U	N	X	W
0	0	1	0	0	1	0
0	1	0	1	0	1	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0 or 1

which expresses the conditions:

(1) The motor is running if and only if C or U;

(2) The motor is reversed if U, and may or may not be reversed if N.

From this table and general rule No. 1, we obtain:

$X = C \mathrm{\ v\ } U = L \mathrm{\ v\ } R'L = R' \mathrm{\ v\ } L'$

$W =$ either U, or $U \mathrm{\ v\ } N =$ either $R'L$, or $R'L \mathrm{\ v\ } RL$, which latter reduces to L. Since we may use either one of the two expressions for W, we can of course use the simpler one, $W = L$. The schematic circuit that corresponds to these equations is shown in Fig. 8.

c. Taking into account a.c. or d.c. light, and homing on the nest or seeking nuts

We have proceeded thus far ignoring the distinction between a.c. and d.c. light, and whether Squee should be homing on the nest, or seeking nuts. But at this stage, we need to take the distinction into account, and we need to translate the previously assumed R and L sensations into appropriate sensations of a.c. and d.c. light depending on Squee's program.

The information reported by the amplifying circuits attached to the photocells is shown schematically in Fig. 9.

Four relays labeled Ra, Rb, La, Lb, are energized in the plate circuits running from the amplifier tubes. The labels are also used for truth values:

Ra: the truth value of "The right photocell sees a.c. light";

Rb: the truth value of "The right photocell sees any light";

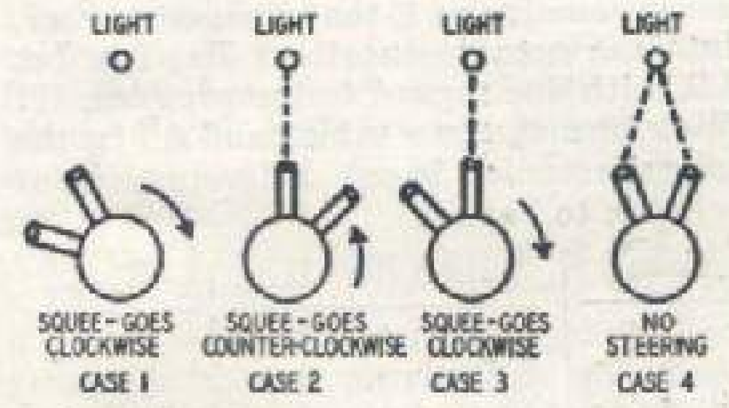
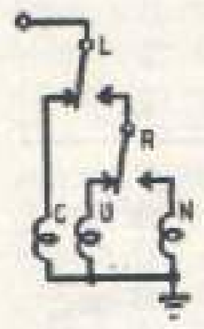


Fig. 5—The sensation circuits of Squee.



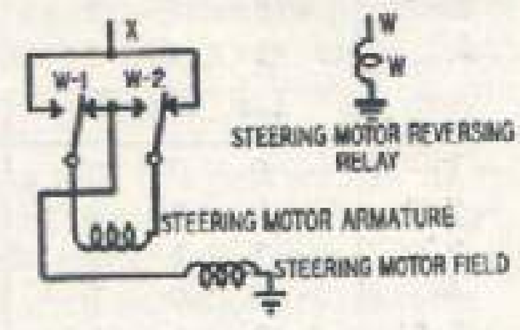
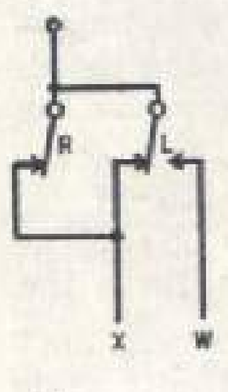
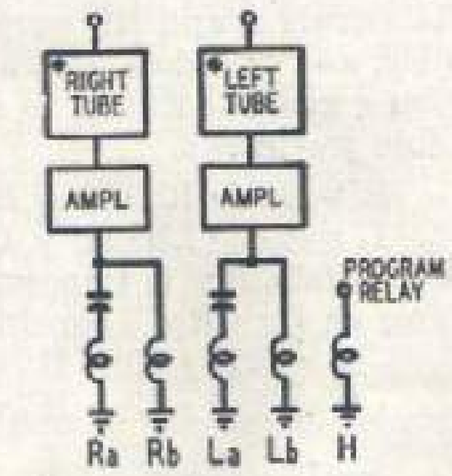


Fig. 6, left—Hypothetical steering circuit. Fig. 7, right—Motor circuitry.





Figs. 8 and 9—Diagrams of light scanning and motor control circuits in Squee.

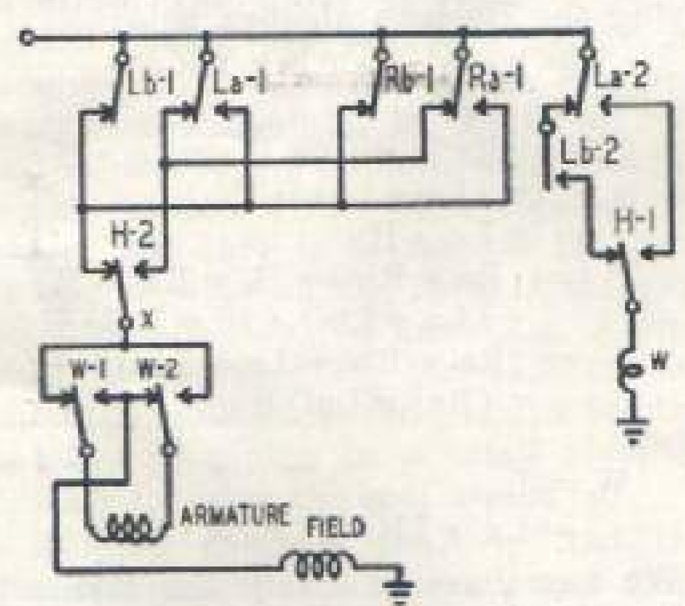


Fig. 10—Steering motor control circuit.

La: the truth value of "The left photocell sees a.c. light";

Lb: the truth value of "The left photocell sees any light."

Looking at Fig. 9, we can see that the amplifying circuit does not report directly "the photocell sees d.c. light." This information must be obtained indirectly.

The H relay shown is the "program" relay which remembers whether Squee is homing on the nest or seeking nuts. If the relay is energized, Squee should pay attention only to a.c. light. If the relay is not energized, Squee should pay attention only to d.c. light. Whether or not it is energized depends on other sensations of Squee (the "tongue" switch, etc.). The letter H, taken from the first letter of "homing," is used to label the relay and also to stand for the truth value of "Squee is homing on the nest".

Now, how do we convert the assumed

sensations R and L that we used earlier, into the actual sensations Ra, Rb, La, Lb, with due regard to the program H? We can make two tables and fill in the cases according to our understanding of what is to happen:

Case	Ra	Rb	H	R
1	0	0	0	0
2	0	1	0	1
3	1	1	0	0
4	0	0	1	0
5	0	1	1	0
6	1	1	1	1

Case	La	Lb	H	L
1	0	0	0	0
2	0	1	0	1
3	1	1	0	0
4	0	0	1	0
5	0	1	1	0
6	1	1	1	1

We can see from Fig. 9 that the case Ra equals 1 and Rb equals 0 is impossible, and so it does not have to be listed. Using general Rule 1 to summarize the table we have:

$$R = Ra' \cdot Rb \cdot H' \text{ v } Ra \cdot Rb \cdot H$$
$$L = La' \cdot Lb \cdot H' \text{ v } La \cdot Lb \cdot H$$

Since $Ra \cdot Rb'$ is impossible, Ra is the same as $Ra \cdot Rb$. Therefore,

$$R = Ra' \cdot Rb \cdot H' \text{ v } Ra \cdot H$$
$$L = La' \cdot Lb \cdot H' \text{ v } La \cdot H$$

Now,

$$X = R' \text{ v } L'$$
$$= (Ra' \cdot Rb \cdot H' \text{ v } Ra \cdot H)' \text{ v } (La' \cdot Lb \cdot H' \text{ v } La \cdot H)'$$

Using the Boolean algebra rule:

$$(mk \text{ v } nk')' = m'k \text{ v } n'k',$$

we have:

$$X = (Ra' \cdot Rb)' \cdot H' \text{ v } Rq' \cdot H \text{ v } (La' \cdot Lb)' \cdot H' \text{ v } La' \cdot H$$
$$= (Ra \text{ v } Rb') \cdot H' \text{ v } Ra' \cdot H \text{ v } (La \text{ v } Lb') \cdot H' \text{ v } La' \cdot H$$
$$= (Ra \text{ v } Rb' \text{ v } La \text{ v } Lb') \cdot H' \text{ v } (Ra' \text{ v } La')H$$

Also,

$$W = L$$
$$= La' \cdot Lb \cdot H' \text{ v } La \cdot H$$

We now have precisely the Boolean expressions that we want, to write down a circuit for controlling the steering of Squee. We obtain X by using four R and L contacts in parallel, running to the negative side of an H contact, and two more R and L contacts in parallel running to the positive side of the same H contact. Similarly we obtain W. See Fig. 10.

This brings us to the end of our short introduction to Boolean algebra, and its use in the design and simplification of circuits involving "yes" and "no" elements. We shall be glad to hear from any reader who is interested in Boolean algebra or in the design and construction of small robots or computers.

—end—

BRAIN SENDS MORSE CODE

The *Codetyper*, a 40-tube brain which sends perfect Morse code as the operator types the message on standard typewriter keys, has been announced by N. Dorfman, New York inventor and electronic technician.

Electronic FLAME CONTROL

By THOMAS L. BARTHOLOMEW

Knowledge of flame controls is needed for their maintenance. Here is how they work.

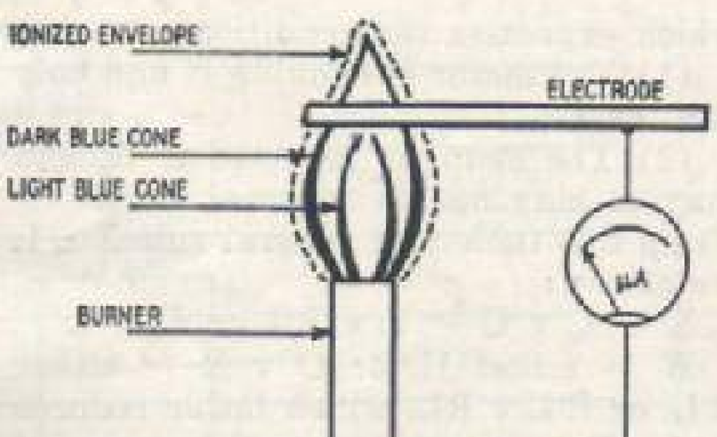


Fig. 1—Diagram of a flame rectifier.

FOR the service man interested in adding maintenance of new electronic devices to his work, here is a device that is right in line with his regular job. With the exception of one new theory, this type of circuit is generally known to most radiomen.

Electronic flame-control devices are found where automatic gas- or oil-fired equipment is used. Their function is to prevent the continued flow of unignited fuel into the combustion space during a flame or ignition failure. An additional important safety function is the prevention of any attempt to relight the burner until a predetermined time has elapsed. This time period allows the fumes to escape and the combustion chamber to be refilled with fresh air—thus preventing an explosion.

The flame is controlled by a flame rod or phototube connected to a twin-triode. The tube in turn operates a relay which—along with a timing arrangement—is the basis of the whole operation.

The rectifying action of a flame electrode depends first of all upon the fact that the chemical action of combustion results in ionizing some of the molecules of gas. The presence of these electrically charged particles enables the flame to conduct a current between two electrodes in contact with it, as indicated in Fig. 1. A steady d.c. voltage is applied across the flame electrode and grounded burner, and a corresponding direct current flows through the flame. Fig. 1 is a schematic diagram intended to suggest, in a very simple way, the way that current flows through the flame. In Fig. 2-a are shown two electrodes of equal size. If electrode X is at a positive potential, and electrode Y negative, the negative ions (free electrons) will be attracted to X and positive ions (molecules positively charged by the loss of electrons) to Y. Electrons reaching X will be absorbed, to replace some of those that have surged through the external circuit to create the negative charge on electrode Y. Positive ions that reach Y will absorb electrons from it and thus become electrically neutral.

With the combustion process continuing, and with a continuous flow of freshly ionized gases through the space between the electrodes, it is apparent that the flow of current in the circuit can also be continuous while an electric