BOOLEAN ALGEBRA

IN

COMPUTER DESIGN

BOOLEAN ALGEBRA IN COMPUTER DESIGN

Boolean algebra is an algebra of logic. It was invented by George Boole in the middle 1800's. Almost one hundred years later, Claude E. Shannon began using it in the design of switching circuits. Today, it is essential in computer design.

This project is a BASIC computer program that simplifies boolean expressions. It applies the following simplification theorems:

$XY+X\bar{Y}=X$

$X+XY=X$

$X+\bar{X}Y=X+Y$

$X+X=X$

$XY+\bar{X}Z+YZ=XY+\bar{X}Z$

This program also prints truth tables for simplified expressions. This makes it easy to check the program's accuracy, and get out bugs.

On display is one digit base 2 to base 8 converter designed with the help of this program. It is constructed from discrete components. It contains 28 transistors, 20 diodes, and numerous resistors.

INSTRUCTIONS FOR USE

If the computer is not already on, dial 285-1020 on the telephone, wait for a beep, and turn on the coupler. When you hear a beep on the phone, insert it into the coupler. Hit the "return" key. The computer will print "PLEASE SIGN ON WITH HELLO." Type "HELLO H5R1080/JZA8" and return. The computer is now on.

The computer will print some nonsense and then it will print a greater-than sign. Type "OLD BOLII" and return. Then type "NHR" and return.

The computer will print "INSTRUCTIONS?". If you want the computer to type instructions, type "YES". If not, type "NO".

When the computer types "AGAIN" you may use one of several commands.

HELP	Gives help.
YES	Starts new problem.
TRUTH	Prints truth table for simplified expression.
CON	Simplifies expression again.
NO	Stops program.
STOP	May be used at any time, not just after "AGAIN", to stop program.

HOW THIS PROGRAM SIMPLIFIES EXPRESSIONS

This program simplifies boolean expressions in a way very similar to the way one would do it by hand. First, it picks two terms to be compared and simplified. Then it counts the number of variables that are different from each other and in what way. It stores the information as follows:

Y=number of variables that are inverts of each other. (ABCD, A$\bar{B}$D)

X1 = number where first is blank, second full. (ABCE, ABCDE)

X2 = number where first is full, second blank. (ABCDE, ABCE)

X=X1+X2

Then it applies the following chart:

THEOREM TO BE APPLIED	Value of: Y	X1	X2	X
$XY+\bar{X}Y=X$	1	0	0	0
$X+X=X$	0	0	0	0
$X+XY=X$	0	1	0	1
$XY+X=X$	0	0	1	1
$X+\bar{X}Y=X+Y$	1	1	0	1
$\bar{X}Y+X=X+Y$	1	0	1	1

After it has compared all the terms with each other, it selects three terms and applies

$$XY+\bar{X}Z+YZ=XY+\bar{X}Z$$

whenever possible.

For a more complete explanation, ask or examine the program card that reads "SIMPLIFYING" on top.

HOW THE PROGRAM WORKS

This is the list of the fundamental part of the program:

STEP (IN BASIC LANGUAGE)	PART OF PROGRAM ***	FUNCTION
00440 IF A$="NO" THEN STOP	*1*	
00450 IF A$="CON" THEN GO TO 520	*2*	
00460 IF A$="YES" THEN GO TO 540	*3*	
00470 GO SUB 12000	*4*	Input number of terms and variables
00480 GO SUB 11000	*5*	Clear, open workspace
00490 GO SUB 8000	*6*	Print heading
00500 GO SUB 1000	*7*	Input expression
00510 GO SUB 2000	*8*	Print expression
00520 GO SUB 6000	*9*	Simplify
00530 GO SUB 2000	*A*	Print expression
00540 IF A$="TRUTH" GO SUB 4000	*B*	Print truth table if requested
00550 IF A$="HELP" GO SUB 13000	*C*	Give instructions
00890 INPUT IN IMAGE "AGAIN? ":A$	*D*	Input command
00900 GO TO 440		

THE BASIC LANGUAGE

Most of the commands in the BASIC language are easy to understand, but a few are not so obvious. Here are a few of these commands:

GO SUB ( )...................... This command goes to a certain step. As soon as the computer reaches a "RETURN" command it goes to the step directly after the "GO SUB" command that sent it.

RETURN .... Command used with "GO SUB" command. Combination of these two can be used to define complex commands, as in above program.

FOR J=A TO B .... Starts loops. J may be replaced by any letter; A and B may be replaced by any letter or number.

NEXT J .... Terminates above loop.

" " .... Print whatever is in the quotes.

Sample Run of Program With Parts Labeled

```
RUN

BØLII         13:52:58    23 FEB 77

INSTRUCTIØNS?NØ                 Not shown
TERMS,VARIABLES
   3,3                                  Part 4
                                             5
       A B C                                 6
TERM 1? 1,2,1
TERM 2? 0,2,1
TERM 3? 0,0,2                                7
CHANGE?NØ

 -   -   -
ABC+BC+C                                     8

                                             9
 - -
B+C                                          A

AGAIN?  CØN                                  D
                                         2 -> 9
 - -
B+C                                          A

AGAIN?  TRUTH                            D -> B
  A B C
  1 0 0   1
  0 1 0   1
  1 1 0   1
  0 0 1   1                                  B
  1 0 1   1
  0 1 1   0
  1 1 1   0
  0 0 0   1
AGAIN?  YES                                  D
TERMS,VARIABLES                              4
   2,2                                       5
                                             6
       A B
TERM 1? 2,1
TERM 2? 2,1                                  7
CHANGE?NØ

-  -
AB+AB                                        8

                                             9
-
AB                                           A

AGAIN?  NØ                                   D
```

BOOLEAN ALGEBRA AND COMPUTER CIRCUITS

Suppose we set up a circuit as in Figure A. The output has only two possible states, positive (+) and ground.

The output voltage will be positive only when

1) Switch A is closed,
or 2) Switches A and B are closed,
or 3) Switches B, C, and D are closed.

If we define our variables as follows:

A=1 if A is closed,
B=1 if B is closed,
etc.
Q=1 if output voltage is positive,

then the statement can be written

$$Q=A+AB+BCD$$

This can be simplified to

$$Q=A+BCD$$

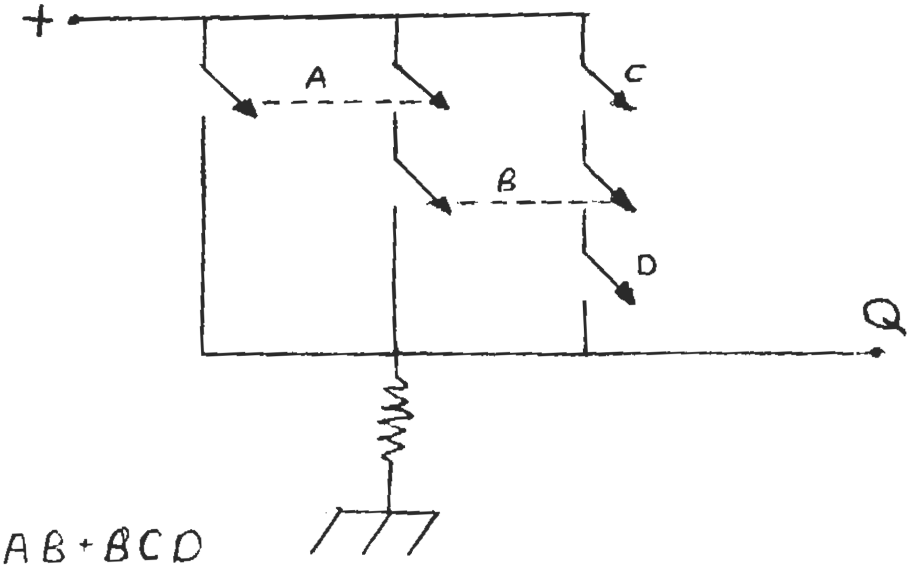
The equivalent circuit for this equation is shown in Figure B. If we try all the switch combinations, we find that Figure A is equivalent to Figure B. Boolean algebra has been used to simplify a switching circuit.

Much more complicated circuits can be simplified using boolean algebra.

The logic described above is called positive logic because a positive voltage is represented by a '1'. In negative logic, positive voltages are defined as ground and negative voltages represent '1's. Negative logic is rarely used.

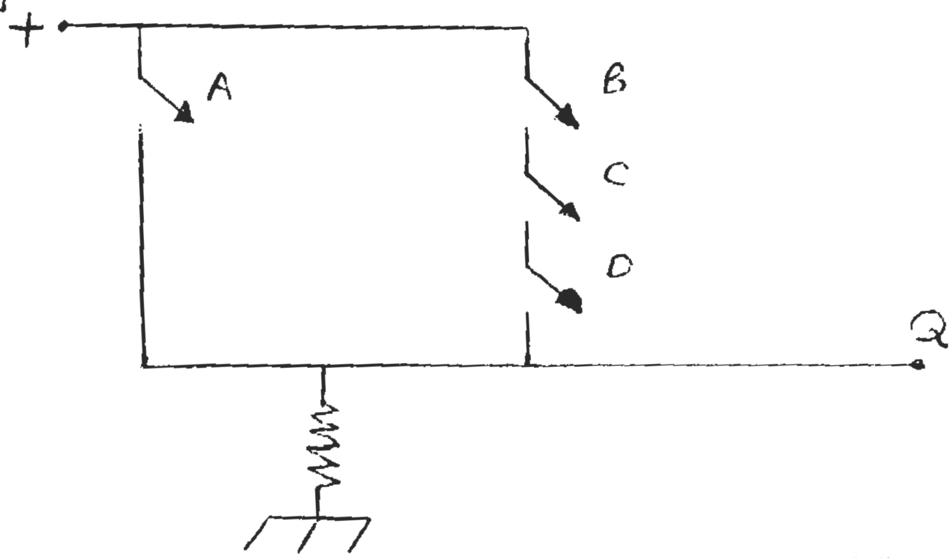
In the design of modern computers, boolean algebra is used to simplify switching circuitry. This switching circuitry is usually TTL, CMOS, or MOS.

A



$Q = A + AB + BCD$

B



$Q = A + BCD$ This is equivalent to "A."

SOME THEOREMS OF BOOLEAN ALGEBRA

1a) $0*X=0$

2a) $1*X=X$

3a) $XX=X$

4a) $\bar{X}X=0$

5a) $XY+XZ=X(Y+Z)$

6a) $XY+X\bar{Y}=X$

7a) $X+XY=X$

8a) $X+\bar{X}Y=X+Y$

9a) $XY+\bar{X}Z+YZ=XY+\bar{X}Z$

Aa) $\overline{X+Y}=\bar{X}\bar{Y}$

1b) $1+X=1$

2b) $0+X=X$

3b) $X+X=X$

4b) $X+\bar{X}=1$

5b) $(X+Y)(X+Z)=X+YZ$

6b) $(X+Y)(X+\bar{Y})=X$

7b) $X(X+Y)=X$

8b) $X(\bar{X}+Y)=XY$

9b) $(X+Y)(\bar{X}+Z)(Y+Z)=(X+Y)(\bar{X}+Z)$

Ab) $\overline{XY}=\bar{X}+\bar{Y}$

Theorems 2b, 3b, 4b, 6a, 72, 8a, and 9a are applied extensively in this project's program.

The 'a' and 'b' theorems are complements of each other; if one can be proved, the other can be proved. This is because every postulate has a complementary postulate. If a theorem can be proved with postulates, then its complement can be proved in exactly the same way with complementary postulates.

PROOFS OF SOME THEOREMS

Theorems 1-4 can be proved by substitution.

5a) This theorem must be proven by substitution:

Y	Z	XY	XZ	XY+XZ		X(Y+Z)	Y+Z
0	0	0	0	0	=	0	0
0	1	0	X	X	=	X	1
1	0	X	0	X	=	X	1
1	1	X	X	X	=	X	1

Therefore, $XY+XZ=X(Y+Z)$

6a)

STATEMENT	REASON
$XY+X\bar{Y}=X(Y+\bar{Y})$	5a
$=X(1)$	4b
$XY+X\bar{Y}=X$	2a

7a)

STATEMENT	REASON
$X+XY=X(Y+1)$	5a
$=X(1)$	1b
$X+XY=X$	2a

8a) STATEMENT	REASON
$X+\bar{X}Y=(X+\bar{X})(X+Y)$	5b
$=(1)(X+Y)$	4b
$X+\bar{X}Y=X+Y$	2a

9a) STATEMENT	REASON
$XY+\bar{X}Z+YZ=XY+\bar{X}Z+(1)YZ$	2a
$=XY+\bar{X}Z+(X+\bar{X})YZ$	4b
$=XY+\bar{X}Z+XYZ+\bar{X}YZ$	5a
$=XY+XYZ+\bar{X}Z+\bar{X}YZ$	...
$XY+\bar{X}Z+YZ=XY+\bar{X}Z$	7a

Aa) This theorem is best proven by subsitution:

X	Y	X+Y	$\overline{X+Y}$		$\bar{X}\bar{Y}$
0	0	0	1	=	1
0	1	1	0	=	0
1	0	1	0	=	0
1	1	1	0	=	0

Therefore, $\overline{X+Y}=\bar{X}\bar{Y}$

If the 'b' version theorems were to be proven directly, each 'b' version proof would have the same form as its corresponding 'a' version proof. The postulates used in the 'b' version proof would be complements of the ones used in the 'a' version proof. This is the proof of theorem 8b. It is quite similar to the proof for 8a.

8a) STATEMENT	REASON
$X(\bar{X}+Y)=X\bar{X}+XY$	5a
$=(0)+XY$	4a
$X(\bar{X}+Y)=XY$	2b

The theorems used to prove 8b and 8a are complements of each other. The statements in the theorems are complementary; the 'and' symbols have been replaced by 'or' symbols and the 'or' symbols have been replaced by 'and' symbols. '+'s have replaced '*'s; '*'s have replaced '+'s. Parentheses have been added to preserve the order of operations.

SYMBOLS OF BOOLEAN ALGEBRA

Boolean algebra is an algebra of logic that can be used to simplify true-or-false statements. Letters are used to represent simple statements; '+' symbols are used to represent 'or'; multiplication symbols are used to represent 'and', and a line above a letter negates the statement the letter stands for.

It is agreed, as in regular algebra, that 'and' is done before 'or'; '*' before '+', unless parentheses direct otherwise. Here is a list of requirements that might appear on an insurance application.

Issue Policy No. 3 only if the applicant

1) Is a male under 25,

or 2) Is married and over 25,

or 3) Is a married female.

We can define our variables as follows:

X=1 if applicant is male
Y=1 if applicant is under 25
Z=1 if applicant is married

This statement can be written:

$P=XY+\bar{Y}Z+\bar{X}Z$

A statement such as 'X' above must be either true or false. By definition:

If 'X' is true, then X=1
If 'X' is false, than X=0

Following are the postulates of boolean algebra.

POSTULATES OF BOOLEAN ALGEBRA

X=1 or else X=0

1*1=1	0+0=0
1*0=0*1=0	0+1=1+0=1
0*0=0	1+1=1
$\bar{1}=0$	$\bar{0}=1$

With these postulates, many useful theorems may be proven.

In ordinary algebra, it is not usually possible to prove a theorem by substituting all possible values of the variables since there may be an infinite number of values.

In boolean algebra, since the variables can have only two values, 0 and 1, theorems can be proven by substituting all possible values for the variables.

COMPARISON OF TTL, CMOS, AND MOS LOGIC

CHARACTERISTIC	TTL	5 VOLTS CMOS	10 VOLTS CMOS	P-MOS	N-MOS
Propagation delay	10 ns	40 ns	20 ns	70 ns	45 ns
Quiescent power	10 nW	10 nW	10 nW	Varies--usually high	
Noise immunity	1 V	2 V	4 V	Varies--2 - 4 V	
Fan-out	10	50 or more, depending on allowable delay			
Power supply in volts	4.7-5.3	1.5-15		Two power supplies often required. Voltage varies widely.	
Input resistance	Medium	Infinite	Infinite	Infinite	Infinite
Input impedance	Resistive	Capacitive		Capacitive	
Output resistance	Low	High (unless buffered)	High	High	High
Standardized	Yes	No		No	

MOS and CMOS are more compact than TTL.

P-MOS and N-MOS are used widely in large, complex, repetitive functions, such as high capacity memories, clock modules, calculator IC's, and microprocessors. Gates and basic logic circuits have not become popular, because of their high output impedance. They are much more compact than TTL.

Because of their infinite input resistance, static electricity can destroy them.

CMOS gates consume practically no power and produce practically no heat. They cost little more than TTL. They operate in a very wide temperature range. CMOS has good noise immunity.

CMOS is not standardized. It uses new, unfamiliar pinouts. Many are not systems oriented.

TTL IC's are known for their high speed. They are available at very low prices. They are compatible with many other logic families. TTL is the most popular logic family.

A variety of other minor logic families are on the market. RTL, DTL, HTL, and ECL find some use in computer applications.

If speed is not important, and power consumption is important, then the best family of gates is CMOS.

BIPOLAR TRANSISTORS, MOSFET'S, AND INTEGRATED CIRCUITS

Two types of bipolar transistors are shown in A. They are the NPN and PNP transistors. The transistor has three terminals: the emitter, the base, and the collector. A very small base current can be used to control a very large collector current.

Figure B is an NPN transistor connected to a battery and a variable voltage source. When the base-emitter voltage is 0 or positive, little or no collector current flows. When base-emitter voltage is negative, electrons from the emitter region are injected into the base region. Most of them reach the collector region. A PNP transistor operates the same way, with holes replacing electrons, and all voltage polarities reversed.

Very many transistors can be produced on a single silicon wafer. Resistors, capacitors, and conductors (wires) can also be produced on the same wafer. Using these components, it is possible to construct a complete, complex circuit on a single silicon wafer. This circuit is called an integrated circuit, an IC, or a chip. IC's containing TTL circuits (transistor transistor logic) can perform all boolean functions and many other functions. TTL chips require a 5 volt power supply. They are currently very popular.

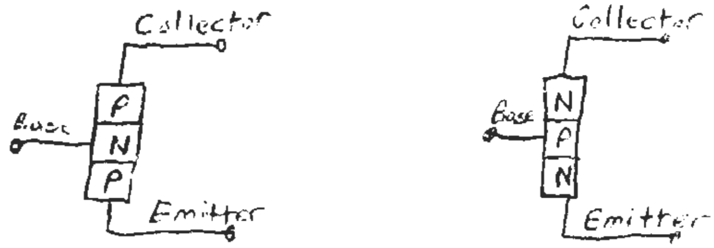
MOSFET stands for Metal Oxide Semiconductor Field Effect Transistor. Two types of MOS transistors are shown in Figure C. They are the P-channel and the N-channel MOSFET's. The MOSFET has three or more terminals: the source, the drain, and one or more gates. A small gate voltage can be used to control a relatively large drain current.

Figure D is an N-channel MOSFET connected to a battery and a variable voltage source. When the gate voltage is 0 or negative, no drain current flows because the drain junction is reverse biased. When the gate voltage is positive, electrons are attracted to the gate and all silicon near the gates becomes N-type because of the high electron concentration. Now, a large drain current flows. Many MOSFET's can be constructed on a single silicon wafer, and MOS and CMOS logic chips can perform all boolean functions, as well as several other functions. MOS and CMOS are becoming increasingly popular.

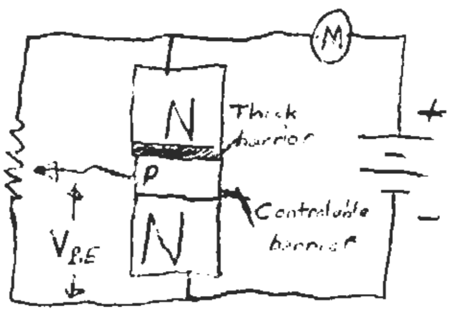
Using MOS technology, more functions can be put in smaller chips at lower cost than for any other technology.

In CMOS technology, N-channel and P-channel MOSFET's are produced on the same silicon wafer. The equivalent circuit of a basic CMOS inverter is shown in Figure E. Using this basic inverter, more complicated gates can be built. CMOS chips consume almost no power because a transistor is always turned off, blocking current flow. Because of their infinite input resistance, one CMOS chip can drive 50 or more other CMOS chips. CMOS chips operate on a very wide voltage range, 1.5-15 volts.

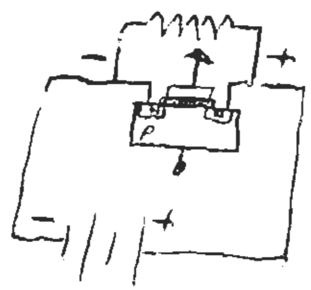
A


Collector
Base
P
N
P
Emitter
Collector
Base
N
P
N
Emitter

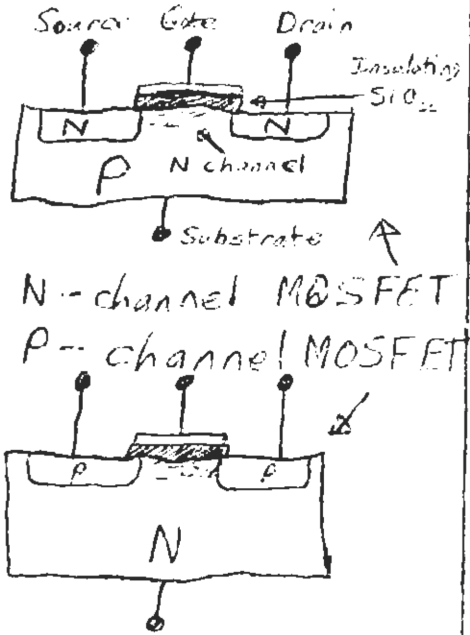
B


M
N
Thick barrier
P
Controlable barrier
N
V_{BE}
+
-

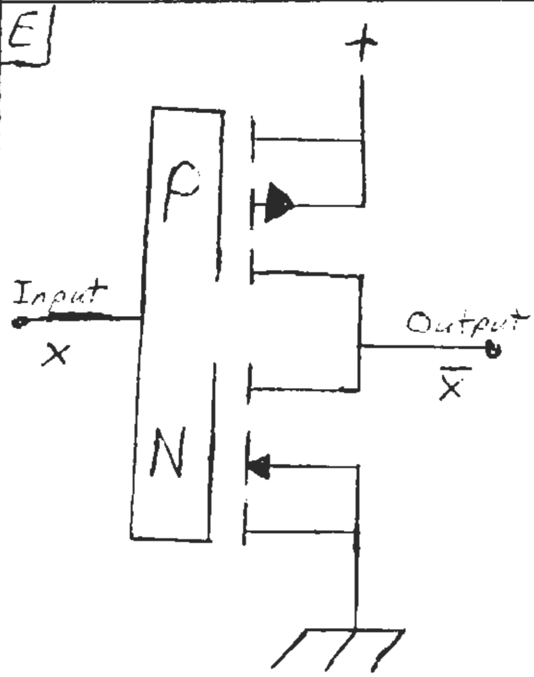
D


-
+
P
-
+

C


Source
Gate
Drain
Insulating SiO_2
N
N
P
N channel
Substrate
N-channel MOSFET
P-channel MOSFET
P
P
N

E


+
P
N
Input
X
Output
$\overline{X}$

SEMICONDUCTOR THEORY

A semiconductor is a material with a resistance of between .1 ohm per cubic centimeter to 100,000 ohms per cubic centimeter. Silicon and germanium are the most popular semiconductors. Silicon is used to make all logic IC's.

A silicon atom has 14 protons, 14 neutrons, and 14 electrons. Silicon has four valence electrons which may be shared with any suitable atom or atoms. (Figure A)

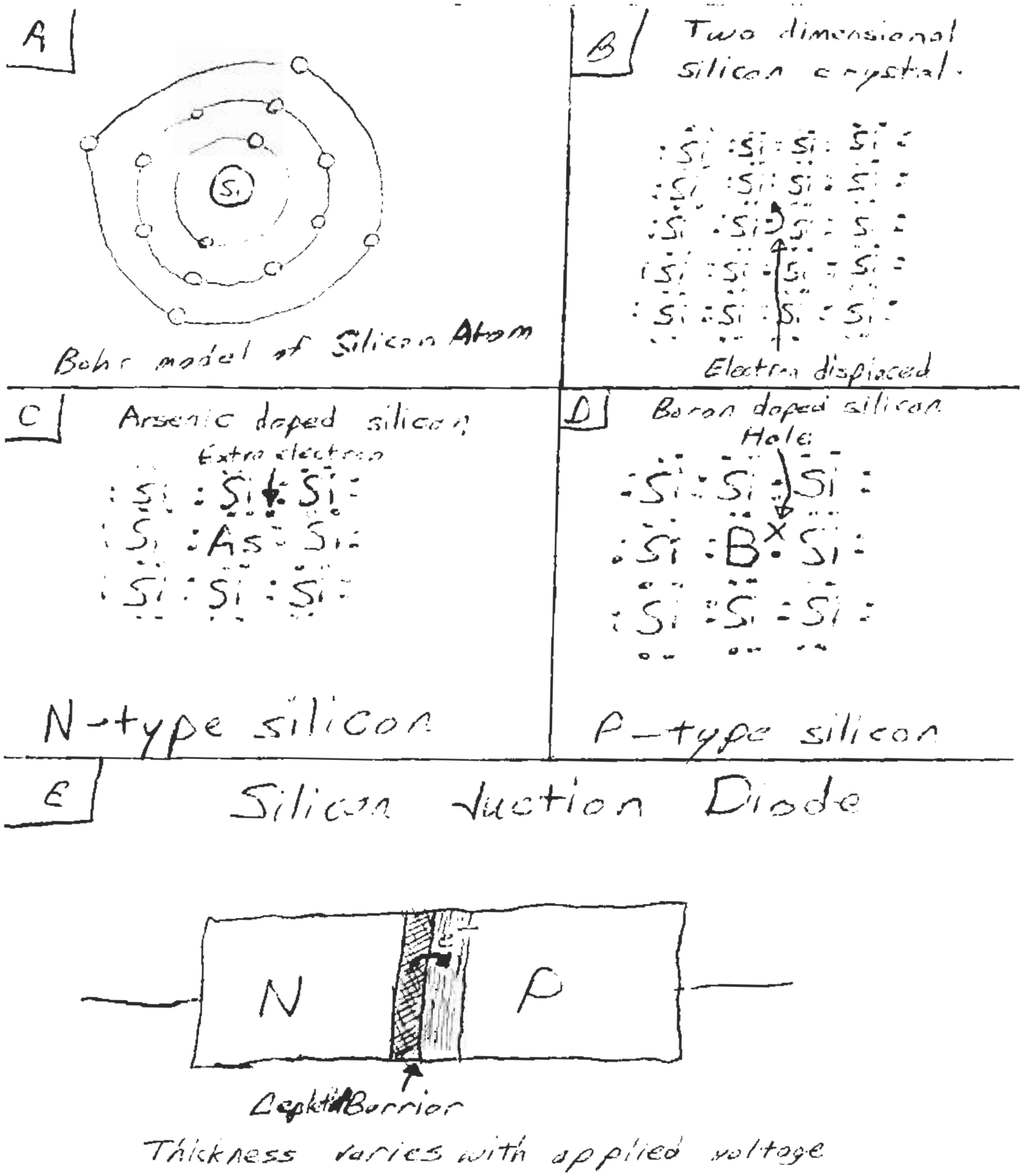
Elementary silicon has a crystal structure in which each silicon atom is surrounded by four other silicon atoms. Each silicon atom shares a pair of electrons with each of the silicon atoms that surrounds it. In this crystal structure, all electrons are tied down in covalent bonds. Although a few electrons are freed from the covalent bonds by heat, the silicon crystal has few free electrons to conduct electricity. (Figure B)

In the manufacture of silicon semiconductor devices, small amounts of boron or arsenic are added to the pure silicon crystal. When arsenic is added to the silicon, the structure in Figure C results. Arsenic has five valence electrons. Four of these are tied down in covalent bonds; the one remaining electron is free to roam the crystal. When boron is added to the silicon, the structure in Figure D results. Boron has three valence electrons, one less than silicon. All of these are tied down in covalent bonds with silicon atoms. Because boron has one less valence electron than silicon, one of the covalent bonds it forms with a silicon atom will be minus an electron. This positively charged "hole" can freely roam the crystal. Thus, a silicon crystal can be made conductive by adding boron or arsenic.

A silicon crystal with arsenic added is called N-type silicon; a silicon crystal with boron added is called P-type silicon. N-type silicon conducts electricity with free electrons; P-type silicon conducts electricity with free, positively charged holes.

A semiconductor diode is made by joining P-type silicon with N-type silicon, and connecting a wire to each type of silicon. (Figure E) Where the two types of silicon meet, electrons move from the N-type silicon to the P-type silicon, forming an insulating barrior. When voltage is applied in direction A, the barrior gets very thick, and little or no current flows. If voltage is applied in direction B, the barrior becomes thin and much current flows. Therefore, a diode conducts in one direction only.

All other semiconductor devices use this type of semiconductor junction.


A
Si
Bohr model of Silicon Atom
B
Two dimensional silicon crystal.
Si
Electron displaced
C
Arsenic doped silicon
Extra electron
As
N-type silicon
D
Boron doped silicon
Hole
B
P-type silicon
E
Silicon Juction Diode
N
P
e-
Barrier
Thickness varies with applied voltage

Sample Run

```
AGAIN?  YES
TERMS,VARIABLES
    3,3

        A B C
TERM 1? 2,1,2
TERM 2? 2,1,2
TERM 3? 1,1,1
CHANGE?NØ

- - - -
ABC+ABC+ABC


- -
ABC+ABC

AGAIN?  YES
TERMS,VARIABLES
    3,3

        A B C
TERM 1? 2,1,2
TERM 2? 2,2,2
TERM 3? 1,1,1
CHANGE?NØ

-  - ---
ABC+ABC+ABC


--
AC+ABC

AGAIN?  YES
TERMS,VARIABLES
    3,3

        A B C
TERM 1? 2,1,2
TERM 2? 2,0,2
TERM 3? 1,1,1
CHANGE?NØ

- - --
ABC+AC+ABC


--
AC+ABC

AGAIN?  YES
```

Sample Run

```
TERMS,VARIABLES
    3,3

        A B C
TERM 1? 2,1,2
TERM 2? 2,0,1
TERM 3? 1,1,1
CHANGE?NØ

-  -  -
ABC+AC+ABC


-   -
AB+AC+BC

AGAIN?  YES
TERMS,VARIABLES
    3,3

        A B C
TERM 1? 1,0,1
TERM 2? 2,1,0
TERM 3? 0,1,1
CHANGE?NØ

   -
AC+AB+BC


   -
AC+AB

AGAIN?  YES
TERMS,VARIABLES
    3,3

        A B C
TERM 1? 0,0,2
TERM 2? 0,0,1
TERM 3? 1,1,1
CHANGE?NØ

-
C+C+ABC


ABC

AGAIN?
```

This "simplification" is wrong. The answer should be "1."

Sample Run

```
AGAIN?  YES
TERMS,VARIABLES
    6,3

        A B C
TERM  1? 2,2,2
TERM  2? 2,1,2
TERM  3? 2,1,1
TERM  4? 1,2,1
TERM  5? 1,1,2
TERM  6? 1,1,1
CHANGE?NØ
```

$\overline{A}\,\overline{B}\,\overline{C}+\overline{A}B\overline{C}+\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC$

$\overline{A}\,\overline{C}+AC+B$

```
AGAIN?  YES
TERMS,VARIABLES
    7,3

        A B C
TERM  1? 2,2,2
TERM  2? 2,2,1
TERM  3? 2,1,2
TERM  4? 2,1,1
TERM  5? 1,2,2
TERM  6? 1,1,2
TERM  7? 1,1,1
CHANGE?NØ
```

$\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+\overline{A}BC+A\overline{B}\,\overline{C}+AB\overline{C}+ABC$

$\overline{A}+\overline{C}+B$

```
AGAIN?  YES
TERMS,VARIABLES
    7,3

        A B C
TERM  1? 2,2,2
TERM  2? 2,2,1
TERM  3? 2,1,1
TERM  4? 1,2,2
TERM  5? 2←1,2,1
TERM  6? 1,1,2
TERM  7? 1,1,1
CHANGE?NØ
```

$\overline{A}\,\overline{B}\,\overline{C}+\overline{A}\,\overline{B}C+\overline{A}BC+A\overline{B}\,\overline{C}+A\overline{B}C+AB\overline{C}+ABC$

$C+\overline{B}+A$