AN OUTLINE OF THE FUNCTIONAL DESIGN OF THE STANTEC ZEBRA COMPUTER

PLEASE ADDRESS ALL INQUIRIES
TO
Intelex Systems
INCORPORATED
Associate of International Telephone and Telegraph Corporation
22 THAMES STREET, NEW YORK 6, N. Y.

AN OUTLINE OF THE FUNCTIONAL DESIGN OF THE STANTEC ZEBRA COMPUTER

STANDARD TELEPHONES AND CABLES LTD.

INFORMATION PROCESSING DIVISION.

NEWPORT, MON.

9/58

Introduction

The following outline of the functional design of Zebra is divided into three parts:

Part One is intended to provide some background to computer operations for those new to computer work.

Part Two is a description of the principles on which the functional design of Zebra is based.

Part Three is a description, in general terms, of the functional design of Zebra. It is intended to provide a basis for the study of the actual 'gating' arrangements used to perform the various machine operations; in other words it is a preliminary study of the 'logical' design of the machine.

Part One

Part Two

Part Three

Part One

1. COMPUTER OPERATIONS

The fundamental operations performed by a digital computer are the arithmetical ones of addition and subtraction. Some machines have built-in facilities for multiplication and division. Others, like Zebra, use a combination of addition and subtraction and right and left shifting, which will be explained later.

A computer can be used to solve complex mathematical problems, because most of these problems can be broken down into a sequence of simple arithmetical operations. The technique of programming is concerned with breaking down a problem in this way, and converting it into a language the machine can recognise.

What has been said above refers to the use of the computer primarily in mathematical and scientific applications. These applications are characterised by a large amount of computation on a relatively small amount of data. In the past few years computers have become increasingly used for other purposes such as industrial control, office management and other data processing systems. A data processing device accepts numerical data (usually in large quantities), processes it in different ways - adds, subtracts, sorts, compares, files, abstracts etc., then takes some action like printing out information resulting from its operations. It will be realised that in these computer applications the emphasis is on storage capacity and input/output facilities, rather than on mathematical calculation.

2. BINARY ARITHMETIC

Zebra, in common with most digital computers, works with numbers in binary notation, and in order to understand the operations that take place in the machine it is necessary to know something about binary arithmetic.

In binary notation any number is represented by using only the digits 1 and 0. Zero is represented by 0, one by 1, two by 10, three by 11, four by 100, and so on. Table 1 gives the binary equivalents of the decimal numbers 0 to 16. These binary numbers may be obtained by starting at zero and adding one successively, according to the rules of addition, which are given in Table 2.

Table 1

Binary equivalents of some decimal numbers

decimal number	binary number
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

Binary notation is useful for computers because it is often more convenient to use computer devices in such a way that they only have to adopt one of two stable states rather than one of many stable states. For example, it is much easier to arrange that a thermionic valve is either conducting or not conducting, than to arrange that it is conducting by a certain amount. Moreover, the most commonly used media for computer input and output, punched paper tape and punched cards, are inherently two-state devices, where information is conveyed by the presence or absence of a hole in a certain position. Such two-state devices can be made to represent either a 1 or a 0.

In the following description of binary arithmetic operations the methods given for multiplication, division and conversion, are those which may be used in paper and pencil calculations and are not the methods actually used in Zebra. The way these operations are performed in Zebra is described fully in the Program Manual.

2.1 Binary Addition

Table 2

Rules of binary addition

$0 + 0 = 0$
$1 + 0 = 1$
$0 + 1 = 1$
$1 + 1 = 0$ and carry 1

A corollary of these rules is that

$1 + 1 + 1 = 1$ and carry 1

Here is an example of the addition of two binary numbers:

1 0 1 1 0 0 1 1	179
1 1 1 0 1 0 1	117
1 0 0 1 0 1 0 0 0	296

2.2 Binary Subtraction

Table 3

Rules of binary subtraction

$0 - 0 = 0$
$1 - 1 = 0$
$1 - 0 = 1$
$0 - 1 = 1$ and borrow 1

Here is an example of the subtraction of two binary numbers:

1 0 1 0 1 1 0 0	172
1 0 0 0 0 1 1 1	135
1 0 0 1 0 1	37

2.3 Binary Multiplication

The rules of binary multiplication are simpler than those of decimal multiplication.

Table 4

Rules of binary multiplication

$$\left.\begin{array}{l} 0 \times 0 \\ 1 \times 0 \\ 0 \times 1 \end{array}\right\} = 0$$

$$1 \times 1 = 1$$

In this example of the multiplication of two binary numbers, we start with the right hand digit of the multiplier and proceed as with decimal numbers:

$$\begin{array}{rr} 1101 & 13 \\ \underline{1011} & 11 \\ 1101 & \\ 1101 & \\ 0000 & \\ \underline{1101} & \\ 10001111 & 143 \end{array}$$

2.4 Binary Division

This operation is also similar to that using decimal numbers, as the following example will show:

$$\begin{array}{lll} 10001 & |11011000011| & 110011 \\ & \underline{10001} & \\ & 10100 & \\ & \underline{10001} & \\ & 11001 & \\ & \underline{10001} & \\ & 10001 & \\ & \underline{10001} & \\ & \cdot\cdot\cdot\cdot\cdot & \end{array}$$

2.5 Positional Notation

In decimal numbers we are familiar with the idea that the value of a digit depends on its position in the number, as the following series indicates:-

. . . . Thousands Hundreds Tens Units . Tenths Hundredths Thousandths . . .

If the number is moved one place to the right it is divided by ten, and if it is moved one place to the left it is multiplied by ten. In binary numbers the value of a digit also depends on its position in the number, and a binary point is used to separate whole numbers from fractions:-

. . . . Sixteens Eights Fours Twos Ones . Halves Quarters Eighths . . . .

Here each digit position represents a power of 2 instead of a power of 10. If a binary number is moved one place to the right it is divided by two, and if it is moved one place to the left it is multiplied by two.

2.6 Conversion from Binary to Decimal

In paper and pencil calculations a binary number may be converted to decimal form by calculating the value of each digit according to its position in the number, and then finding the sum of the values. The following example is a conversion of the numbers used in the division example of Section 2.4:

Dividend (binary)	1	1	0	1	1	0	0	0	1	1	
(decimal)	512	256		64	32				2	1	Sum = 867
Divisor (binary)						1	0	0	0	1	
(decimal)						16				1	Sum = 17
Quotient (binary)					1	1	0	0	1	1	
(decimal)					32	16			2	1	Sum = 51

2.7 Conversion from Decimal to Binary

A decimal number may be converted to binary form by dividing it successively by two and noting the carry, 1 or 0, at each stage.

These carries then make up the binary number, as in the following example:

2	86		
2	43	+	0
2	21	+	1
2	10	+	1
2	5	+	0
2	2	+	1
2	1	+	0
	0	+	1

The bottom digit is the most significant, so that 86 is represented in binary notation by 1 0 1 0 1 1 0.

2.8 Negative Numbers

In paper and pencil calculations with decimal numbers a negative number is usually indicated by prefixing an absolute number with a minus sign, and this can be done when using binary numbers. In machine operations, however, using a machine which can represent only the digits 1 and 0, the sign of a number will also have to be represented by a 1 or a 0. Zebra uses the complementary system for representing negative numbers, and in this system the sign digit of a negative number is always a 1.

3. 'LOGICAL' OPERATIONS AND DEVICES

One of the most important actions that takes place in an electronic digital computer is that of switching, and in Zebra the most important devices used for this purpose are 'gates', bi-stable units and buffer-inverters.

3.1 Gates

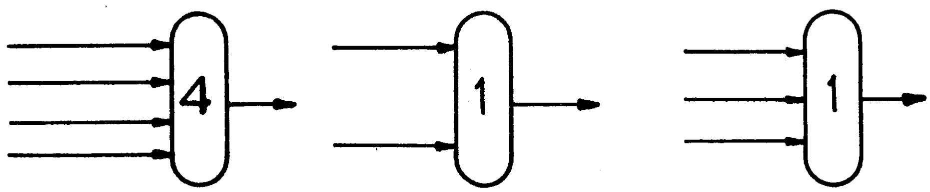
In Zebra two types of gate are used called 'and' and 'or'. These names are derived from the similarity between the operations performed by these gates and operations performed in the subject of symbolic logic.

Information is passed from one part of the machine to another in the form of electrical pulses, and gates will be open or closed to these pulses according to whether or not their operating conditions are fulfilled.

An 'and' gate will be open, and produce a pulse on its output terminal, only when pulses occur on <u>all</u> of its input terminals.

An 'or' gate will be open, and produce a pulse on its output terminal, when a pulse occurs on <u>any</u> of its input terminals.

In logical diagrams gates are shown thus:



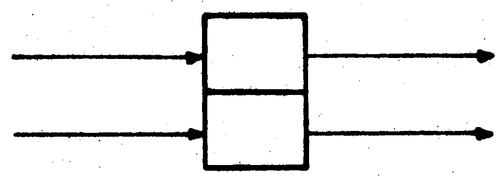
The two kinds of gate are distinguished by the number shown inside the logical symbol. 'And' gates have a number corresponding to the number of inputs, whereas 'or' gates always contain the number 1 regardless of the number of inputs.

3.2 The Bi-stable Unit

In the form in which it is used in Zebra the bi-stable unit, often called a flip-flop, consists of two valves connected in such a way that when either one is conducting the other is not. It follows that each 'side' of a bi-stable unit is always in one of two stable conditions: these conditions are referred to variously as - set or unset; down or up (referring to the voltage at the anode of the valve); or, by convention, 1 or 0. When either side of a bi-stable unit is set it stays set until the other side is triggered by a pulse, whereupon it changes to the unset state.

In logical operations the bi-stable unit is used mainly for its ability to represent, and indicate the presence of, a 1 or a 0. In some applications it is useful for its characteristic of causing a one-digit-period delay between pulses on its input and output terminals.

The bi-stable unit is shown in diagrams thus:

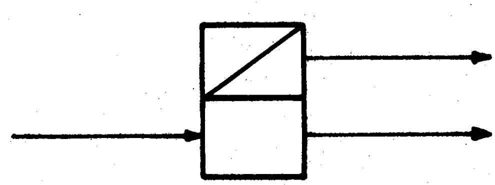


3.3 The Buffer-inverter

A buffer-inverter has one input terminal and two output terminals. The two outputs may be called normal and inverted. A pulse representing a 1 on the input terminal produces a 1 on the normal output terminal and a 0 on the inverted output terminal. A pulse representing a 0 on the input terminal produces a 0 on the normal output terminal and a 1 on the inverted output terminal.

The action of the buffer-inverter differs from that of a bi-stable unit in that the output follows the input; and there is no delay between the input and output pulses. The output of a buffer-inverter also provides more power than does the output of a bi-stable unit.

In diagrams the buffer-inverter is shown thus:



4. STORAGE

If we consider what is the essential difference between an automatic digital computer and, say, a desk calculating machine, we find that it lies in the function of storage. An electronic computer can perform arithmetic operations many times faster than a desk machine, but if it were necessary for a human operator to instruct the machine step by step, and make a note of partial results as they were obtained, its speed would be largely wasted. This step by step control is not necessary because a computer is able to store within itself instructions, numbers to be operated on, and partial results.

All storage in Zebra takes place on the surface of a magnetic drum, and this will be described in some detail.

4.1 The Magnetic Drum

This is a hollow aluminium drum 6" in diameter and 15" high which revolves about a vertical axis at 6000 r.p.m. The surface is coated with a thin layer of nickel, which is a convenient magnetic material.

Information is recorded on the drum surface by means of read/write heads (in computers recording is referred to as writing, and the extraction of information is referred to as reading). In writing, an alternating current is passed through the head, and this causes a pattern of magnetisation to be formed on the drum surface as it passes beneath the head. The phase of the writing current waveform is made to represent binary digits, which are reproduced by the magnetic pattern. In reading, the pattern of magnetisation is converted into a voltage waveform, which is then amplified and decoded to produce the binary digits it represents. An advantage of magnetic storage is that information may be read from a store without destroying it, yet may easily be erased when no longer needed.

The surface round the circumference of the drum which passes beneath a head is called a track. By spacing heads along the length of the drum a large number of tracks are formed and used for the following purposes:-

Number of Tracks	Purpose
256	Main Store (Section 10.3)
12	Registers (Section 10.4)
2	Arithmetic Accumulators 'A' and 'B' (Section 10.2.1)
1	Control Register 'D' (Section 10.1.2)

Number of Tracks	Purpose
1	P. N. or 'current address' track (Section 10. 3. 2)
1	T. W. or Timing Wave track. The frequency of the Timing Wave is 128 kc/s.

The P. N. and T. W. tracks control all timing in the computer.

4. 1. 1 Word storing

In Zebra the basic word-time (the time taken to read or write a word) is approximately 312 microseconds, so that 32 words can be written into a main store track. In computers a word is an ordered set of digit symbols which can represent either a number or an instruction. The number of digits in a word is usually constant in a given machine, and in Zebra it is 33.

Part Two

5. GENERAL DESIGN FEATURES OF ZEBRA

Stantec Zebra is a general purpose automatic digital computer designed and built in accordance with the following principles:

5.1 Technical Simplicity and Economy of Equipment

This is achieved by using:-

(a) Binary operation
(b) Serial rather than parallel working
(c) An arithmetic unit designed to perform (basically) addition and subtraction only; multiplication and division being programmed externally
(d) A simple control circuit
(e) The use of a magnetic drum for storage.

5.2 Optimum Speed of Operation

This is achieved by:-

(a) The provision of a number of registers, the contents of which are immediately accessible
(b) A functional design which gives elegant means of performing counting, instruction modification, repetition of instructions, double length arithmetic and floating point operation, without additional equipment. Some sample operation times are:-

Addition or subtraction	312 microsecs.	
Multiplication	11 millisecs.	
Division	35 "	
Extraction of square root	80 "	
Calculation of cosine	80 "	
Floating point operation	30 "	
Floating point matrix inversion	$110n^3$ "	where n is the degree of the matrix.

By making full use of the flexibility of the instruction code an average operation time (measured over numerous programs) was found to be two milliseconds, which compares favourably with the performance of machines having a faster basic operation time.

5.3 Extreme Flexibility of the Instruction Code

This is achieved by using an instruction word in which each function digit controls only one operation within the machine, and a logical design that enables several operations to be performed at the same time.

There are fifteen function digits in the instruction word and each one can be either 1 or 0, i.e. present or absent in an instruction. Thus a possible 2^{15}, or more than thirty-two thousand different instructions may be used.

One of Zebra's outstanding features is the ability to perform several operations at the same time. For this reason the machine compares favourably in its true speed of operation with machines having a much faster electronic speed. Zebra can perform as many as nine operations simultaneously, and in a typical program it may perform an average of five operations for each instruction.

5.4 Reliability

One of the advantages of designing Zebra as a computer of medium size and medium electronic speed is increased reliability, and this for two reasons. First, every component in the machine has a certain failure rate however low improved manufacturing techniques may make it, and the smaller the total quantity of equipment the lower the failure rate of the machine as a whole. Secondly, it is found that as the operating speed of a machine is increased so the safety margin which allows for such things as alterations in component characteristics is decreased, and there is a greater chance of error.

A very useful contribution to reliability is provided by marginal checking where this is part of a regular maintenance routine. Operating conditions, e.g. voltages, are varied until the machine is working close to the safety margin. Programs designed to test the various machine functions are then run through, and in this way incipient component failures may be detected, and the failing component replaced.

5.5 Ease of Maintenance

The relatively simple functional design and the fact that only a few different types of functional unit are used make the machine easier to understand, and therefore easier to maintain.

A cathode-ray-tube monitor is provided on which can be displayed at the same time the contents of four stores. These are:-

(i) The A accumulator
(ii) The B accumulator
(iii) Any of the registers 4 to 15, or either of the control registers C and D
(iv) Any one of the main store locations.

Of great assistance in maintenance is Zebr's plug-in unit construction. A spare unit can be used to replace a faulty one, which may then be repaired away from the computer. This method, as well as making maintenance easier, reduces the loss of machine time (the time during which the machine is usefully employed).

5.6 Adequate Storage Capacity

Zebra's main store has a capacity of more than 270,000 binary digits. In addition, there is a high-speed store of about 400 binary digits.

Development work is being done on auxiliary magnetic storage devices, in the form of ferrite core matrices and magnetic tape.

5.7 Ease of Programming

Programming is the technique of preparing a sequence of instructions for a computation. As has already been mentioned, the Zebra instruction code is extremely flexible and powerful. In order to make full use of it, however, a good deal of skill and experience is required in the programmer, and there will be occasions when the machine could provide a useful tool to people who have not had extensive training in computer programming; engineers in a research laboratory for example. To make this possible a special instruction code known as the 'simple code has been devised, which works in conjunction with interpretive routines stored within the machine. This code allows day to day problems to be programmed after only brief training, and it is a remarkable achievement that speed of operation in simple code is one-sixth to one-fifth as fast as in 'normal code' and in some applications, e.g. matrix inversion, half as fast.

5.8 Simplicity of Input and Output

Punched paper tape is used as the basic medium of input and output because of its simplicity and economy, and the fact that standard equipment is available for its use. For input, a Ferranti photo-electric tape reader is used, operating at a speed of 100 characters per second. For visual output a Creed teleprinter prints at a speed of 6.7 characters per second (soon to be replaced by one operating at a speed of 10 characters per second), and for punched tape output a Teletype punch operates at a speed of 50 characters per second.

For particular applications, punched cards and magnetic tape may be used for input and output.

Part Three

6. MACHINE OPERATIONS

For computer solution mathematical problems are broken down into a sequence of simple arithmetical operations. In the solution of any problem, or in the processing of data, the machine has to be provided with the data on which operations are to be performed and instructions which control these operations step by step. In a typical computation, instructions and data are fed into the machine and stored in numbered locations: when the computation begins, instructions are executed one after another in the order prescribed directly or conditionally by the programmer until the computation is finished.

6.1 Serial Operation and Digit Timing

There are two different methods by which binary words may be transmitted from one part of the computer to another. One method requires a separate channel for each digit of the word, all digits being transmitted in parallel, i.e. simultaneously. The other method requires only a single channel and the digits of the word are transmitted serially, i.e. one after another. With the exception of the parallel transfer of a word from the C control register to the E control register (10.1.1) Zebra works in the serial mode.

The binary digits of a word in Zebra are represented by a train of electrical pulses. The significance of each digit in the word is governed by the time at which it occurs in relation to timing pulses generated by the machine. These timing pulses are produced in groups of forty, each pulse being distinguishable as the first, second, third and so on. The first timing pulse is called P_{33}, the second P_{32}, the third P_{31} and so on to the last which is P_{-6}, the cycle then re-commencing with P_{33}. Fig. 1 shows one group of timing pulses.

All words transferred serially from one part of the machine to another travel least significant digit first.

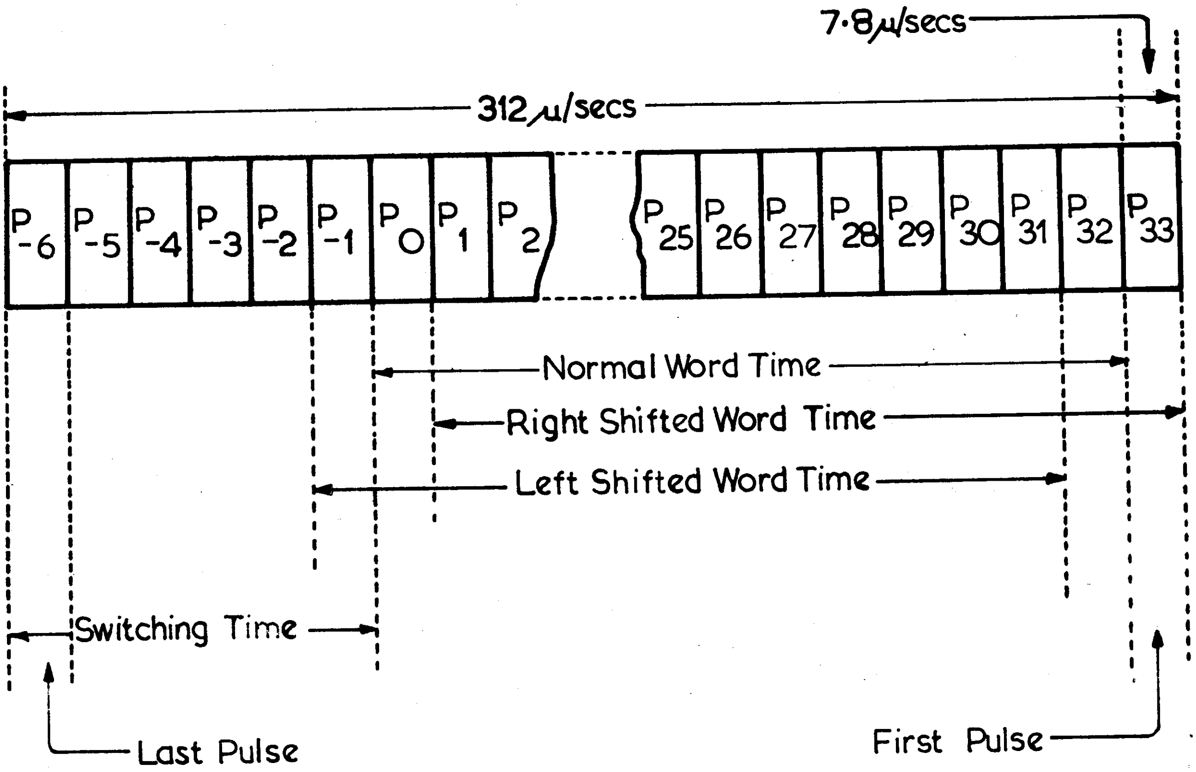


Fig. 1

7. NUMBER WORDS

In the machine numbers normally consist of 33 binary digits, designated 32 to 0 from the least to the most significant. The most significant digit is the sign digit, being 0 in a positive number and 1 in a negative number. In the A accumulator (10.2.1) numbers are provided with an extra sign digit, which is called a_{-1}.

In a computation numbers may be regarded either as all fractions with a binary point assumed between the sign digit and the next most significant digit, or as all integers. This is called fixed point operation, in which numbers have to be scaled by the programmer so as to remain either fractions or integers throughout the computation.

Zebra is primarily regarded as a fixed-point machine but it can be programmed to use a form of number representation called floating point, in which a number consists of two parts called mantissa and exponent. For example, 81060 would be represented in floating point form as 0.8106×10^5, where the mantissa is 0.8106 and the exponent is 5. This form, although reducing the speed of arithmetic operations, avoids the necessity for scaling.

8. INSTRUCTION WORDS

An instruction, when contained in the machine, is also in the form of a 33 binary digit word. It is made up of three parts - a thirteen digit Main Store address, a five digit Register address, and fifteen Function digits.

Function Digits															Register Address					Main Store Address												
																				Track Selection								Drum Position				
A	K	Q	L	R	I	B	C	D	E	V	V_4	V_2	V_1	W	R5	R4	R3	R2	R1	S8	S7	S6	S5	S4	S3	S2	S1	P5	P4	P3	P2	P1

Fig. 2.

Machines with an instruction word containing two addresses can be used in different ways. If one address is used in performing arithmetic on a number and the other address is used to specify the location of the next instruction, the machine is known as a 1 + 1 address machine. If both addresses are used in performing arithmetic on numbers then instructions are executed in sequence, and the machine is known as a 2 address machine. Zebra can operate in either of these ways at different times during a computation.

9. FUNCTION DIGITS

A Functional Diagram of Zebra's four basic units will be found at the end of the book, and with the aid of this diagram the action of most of the function digits can be followed. The letters TP shown by two of the switches in the control unit are not function digits but stand for the words Transfer Pulse. The switching actions shown in the diagram are performed in the machine by the 'gates' and other logical elements described in Section 3.

It will be seen from the diagram that the interconnection of the four parts of the machine, and the direction of transfer of words between them, is controlled by the digits A, K, D and E.

9.1 The A Digit

When A = 1:

(a) the main store and the arithmetic unit are connected

(b) a transfer path is provided from the D control register to the C control register

(c) a transfer path is provided from register 4 to the D control register.

When A = 0 = X:

(a) the main store and the control unit are connected

(b) a transfer path is provided from the C control register to the D control register.

When the programmer wants the machine to perform a particular operation he writes the appropriate function digit in the instruction; the function digits concerned with operations he does not want performed being left out. The A digit is made an exception to this rule for the following reason. When the machine is receiving a list of instructions from the input device the A digit, being in the most significant position, indicates the end of each instruction, thereby enabling the machine to separate one instruction from another. If the A digit is not required in an instruction the programmer writes the letter X instead, in the most significant position, and the machine recognises this also as signifying the end of an instruction. So an instruction in which the A digit is absent is known as an X instruction.

9.2 The K Digit

When K = 1 : the registers and the control unit are connected.
When K = 0 : the registers and the arithmetic unit are connected.

9.3 The D Digit

When D = 1 : writing into the main store from either of the accumulators occurs. It is not possible to write directly into the main store from the control unit.

When D = 0 : reading from the main store into the control unit or the arithmetic unit occurs.

9.4 The E Digit

When E = 1 : writing into the registers from the control unit or the arithmetic unit occurs.

When E = 0 : reading from the registers to the control unit or the arithmetic unit occurs.

E and $\overline{E}$ (not E, or E = 0) are used with certain pseudo registers (10.4.2) to activate input/output equipment.

9.5 The Q Digit

When Q = 1 : a 1 is added to or subtracted from (depending on the I digit) the least significant end of the number in the B accumulator, independently of any other arithmetic operation occurring at the same time. This operation only refers to the B accumulator.

9.6 The L Digit

When L = 1 : the number in each accumulator is shifted one place to the left.

The extra sign digit a_{-1} of the number in the A accumulator is lost, and what was the digit a_0 becomes the new a_{-1}. The space left at the least significant end of the A accumulator is filled with what was the most significant digit of the B accumulator.

When it is required to shift the number in one accumulator without shifting the number in the other, it can be done in the following way. The accumulator containing the number not to be shifted is cleared, by breaking the circulation path; at the same time the cleared number is re-written into the accumulator from register 2 if it is the A accumulator, or register 3 if it is the B accumulator.

9.7 The R Digit

When R = 1 : the number in each accumulator is shifted one place to the right. The least significant digit of the number in the B accumulator is lost. The space left at the most significant end of the

B accumulator is filled with what was the least significant digit of the A accumulator. The space left at the most significant end of the A accumulator is filled with a copy of what was the a_{-1} digit.

We can right shift the number in one accumulator without shifting the number in the other by the same method as that described for the left shift operation.

9.8 The I Digit

When I = 1 : the number entering the arithmetic unit is subtracted from, instead of added to, the accumulator selected by the B digit.

9.9 The B Digit

The B digit determines whether the A or B accumulator is selected when writing into, or reading from, the arithmetic unit.
When B = 1 : the B accumulator is selected.
When B = 0 : the A accumulator is selected.

9.10 The C Digit

When C = 1 : the accumulator selected by the B digit is cleared.

It is possible to clear a number from an accumulator and, at the same time, to write the number into a main store location or register.

9.11 The V Digits (V, V4, V2, V1)

When the programmer wishes to impose a condition on the execution of an instruction (a typical condition being 'execute this instruction only if the contents of the 'A' accumulator are negative') he uses the V digits. These are decoded in the test-and-transfer box to produce 15 combinations as shown in Table 5. Some of the combinations are used as tests and the remainder are used for other purposes as described in the following subsections. The letters A and B in Table 5 refer to the A and B accumulators.

Table 5

Decoding of V Digits.

V	V4	V2	V1	Combinations
1	0	0	0	= V Releases any carry/borrow from B to A
1	0	0	1	= V1 Test A Negative
1	0	1	0	= V2 Test B Negative
1	0	1	1	= V3 Test A zero
1	1	0	0	= V4 Test least significant digit of B
1	1	0	1	= V5 May be used in conjunction with external equipment
1	1	1	0	= V6
1	1	1	1	= V7
0	0	0	1	= U1 Used in conjunction with manual keys on control panel
0	0	1	0	= U2
0	0	1	1	= U3
0	1	0	0	= U4
0	1	0	1	= U5
0	1	1	0	= U6
0	1	1	1	= U7

9.11.1 V

When arithmetic takes place in the B accumulator a carry/borrow digit may be produced. This is trapped in the 'carry trap' and is only released into the A accumulator by an instruction which contains the combination V.

9.11.2 V1 - Test A Negative

An instruction containing the combination V1 is executed only if the number in the A accumulator is negative, i.e. when the sign digit is a 1.

9.11.3 V2 - Test B Negative

An instruction containing the combination V2 is executed only if the number in the B accumulator is negative.

9.11.4 V3 - Test A Zero

An instruction containing the combination V3 is executed only if the number in the A accumulator contains a 1; in this case the extra sign digit a_{-1} is disregarded.

9.11.5 V4 - Test Least Significant Digit of B

An instruction containing the combination V4 is executed only if the least significant digit of the number in the B accumulator is a 1.

9.11.6 V5, V6, V7

These combinations are not used as tests, but may be used to activate input/output equipment.

9.11.7 The U1 to U7 Combinations

These combinations are used in conjunction with manual keys, called U1 to U7, located on the control panels. An instruction containing any U combination is executed only if the U key corresponding to it is depressed. The U1 to U7 combinations are also used with stored programs to enable certain instructions to be given to the machine.

9.12 The W Digit

This is not normally provided directly by the programmer, but is supplied automatically within the machine in certain circumstances.

If an instruction requires a word to be read from or written into a main store location its execution must be delayed until the drum has rotated into the appropriate position; and in this case the W digit is absent.

When W = 1 : the main store reading and writing circuits are inhibited and the instruction containing it is executed immediately, i.e. without waiting for the drum. This is used in 'repeat' instructions in which the main store address, made redundant by the inclusion of the W digit, can be used as a counting device.

9.13 LR

The instruction LR cannot be executed literally, but is used to perform a special operation known as the multiplication facility, which has two actions:-

(i) The numbers in the A and B accumulators are right shifted;
(ii) The A accumulator has added to it or subtracted from it:
(a) the contents of register 15 if b_{32} (least significant digit of the number in the B accumulator) = 1: or
(b) zero if $b_{32} = 0$.

9.14 XD

XD causes the addition or subtraction of the contents of register 15 to the A or B accumulator. The XD facility is used in the division process.

9.15 Some typical instructions

Apart from A or X, which must come first, it does not matter in what order the rest of the function digits are written. The main store address is normally written before the register address but in 'repeat' instructions the register address is written first and the W digit is then automatically inserted into the instruction by the machine. Addresses may be separated by function digits or by a point. Main store addresses less than 32 are written with three digits, from 000 to 031, to differentiate them from register addresses. If it is not required to use a main store location, or a register, in any instruction, the corresponding address is omitted when the instruction is written. When no main store address is written the machine inserts the W digit into the instruction.

Instruction as written	Meaning of instruction
X100BC5	Take your next instruction from main store location 100. Add the contents of register 5 into the cleared B accumulator. (In this type of instruction, one address is used to specify the location of the next instruction and the other is used to do arithmetic. This is an example of 1 + 1 address operation).
X327BCE5	Take your next instruction from main store location 327. Write the contents of the B accumulator into register 5. Clear the B accumulator.

Instruction as written	Meaning of instruction
X569LQK7	Take your next instruction from main store location 569, modified by the addition of the contents of register 7. (This kind of operation is known as the 'B' line facility). Left shift both accumulators. Add 1 into the least significant position of the B accumulator.
X641KE4	Take your next instruction from main store location 641. Write the contents of the D control register into register 4.
A2119.9	Add the contents of main store location 2119 and register 9, and add the sum into the A accumulator. Your next instruction will come from the D control register. (In this type of instruction both addresses are used to do arithmetic. This is an example of 2 address operation).
AD289I9Q	Write the contents of the A accumulator into main store location 289. Subtract the contents of register 9 from the A accumulator. Subtract 1 from the least significant position of the B accumulator. Your next instruction will come from the D control register.
AD4015BCE9	Write the contents of the B accumulator into main store location 4015 and into register 9. Clear the B accumulator. Your next instruction will come from the D control register.
AC262K6	Add the contents of main store location 262 into the cleared A accumulator. Your next instruction will come from the D control register, modified by the addition of the contents of register 6.
X101.5V2	If the contents of the B accumulator are negative, take your next instruction from main store location 101, and add the contents of register 5 into the A accumulator. If the contents of the B accumulator

Instruction as written	Meaning of instruction
	are zero or positive, do not execute this instruction but take your next instruction from the D control register.
AB3570IV3	If the contents of the A accumulator are not zero, subtract the contents of main store location 3570 from the B accumulator. If the contents of the A accumulator are zero, do not execute this instruction. Your next instruction will come from the D control register whether this instruction is executed or not.
X1856LBC3	Take your next instruction from main store location 1856. Left shift the A accumulator. (Here, although the L digit indicates a shift of both accumulators, the BC digits indicate that the B accumulator is to be cleared. However, as register 3 is equivalent to the B accumulator, the unshifted contents of the B accumulator are replaced after shifting and clearing have taken place).
X5000RC2	Take your next instruction from main store location 5000. Right shift the B accumulator. (This is an analogous instruction to the previous one, register 2 being equivalent to the A accumulator).

10. THE FOUR BASIC UNITS

Neglecting input and output, to perform computations or to process data a computer needs three basic parts: an Arithmetic part for performing the necessary arithmetic operations; a Control part for controlling the execution of instructions; and a Store part for storing instructions and data. In Zebra there are two kinds of store, (a) a large store, called the Main Store, from which there may be some delay in extracting information and (b) a small high-speed store consisting of what are called Registers (sometimes immediate-access registers) where the stored information is immediately available. We may say, then, that the machine consists of four basic functional units. The following description of these units is based on the Functional Diagram, which will be found at the end of the book.

10.1 The Control Unit

The purpose of the control unit is to enable the machine to carry out programmed instructions. In some machines the control unit simply executes stored instructions in a fixed sequence from the beginning to the end of the program, but Zebra can do more than that. By means of the Zebra control unit it is possible to:-

(a) Execute instructions in sequence
(b) Execute instructions in any desired order
(c) Make the execution of an instruction depend upon a specified condition.
(d) Automatically provide an instruction to replace one which fails to meet a specified condition
(e) Modify an instruction (by adding another to it) during computation
(f) Repeat an operation a specified number of times before going on to the next instruction.

To perform its operations the control unit has three registers called, C, D and E, an adder, and a device called a test-and-transfer box. The D register has a counter associated with it.

10.1.1 The C register and the test-and-transfer box

The purpose of the C register is to hold incoming instructions while any V test digits are examined, and until the drum is in the required position for reading a word from or writing it into a particular main store location.

The register is made up of bi-stable units connected in such a way that digits entering at one end shift along from one unit to the next until stopped at the appropriate time. The register also has a circulation path associated with it.

An instruction, which may come from a main store location, a register, or the D control register, enters serially into the C register and shifts along until it is wholly contained in the register at time P_0. By this time the test-and-transfer box has already examined that part of the instruction indicating the required position of the drum. Any V test digits present are then examined, and if the condition specified by them is satisfied, and the drum is in the required position, the test-and-transfer box produces a transfer pulse which causes the instruction to be transferred in parallel to the E register for execution. If any test is satisfied but the drum is not in the required position the instruction circulates until the drum is in the required position, and is then transferred to the E register. If an instruction contains the W digit, and any specified test is satisfied, the instruction is transferred to the E register immediately.

When the condition specified by a V digit test is not satisfied the instruction containing the test is not transferred to the E register but is erased, and the next instruction enters the C register from the D register.

An X instruction, as well as being transferred in parallel to the E register, passes serially through the counting device into the D register and has its drum address increased by two.

10.1.2 The D register and associated counting device

The D register, which is a track on the drum similar to those used for the arithmetic accumulators (10.2.1), performs three functions:

(a) it provides the next instruction when an A instruction is currently being executed
(b) it provides the next instruction when the test on an instruction in the C register fails
(c) it makes it possible to repeat an instruction a specified number of times.

On the right of the C register will be seen a counter represented by a plus sign inside a circle, two switches marked A and TP and a box representing the D register.

When an X instruction is transferred from the C register to the E register, the switch A on the left of the D register will be up and switch TP down, enabling the instruction to pass through the counting device into the D register. In passing through thecounting device the drum position part of the main store address is increased by two. The reason for increasing the drum position by two is that the instruction in the D register, when required, takes one word time to pass into the C register; so that if the address had been increased by only one, the appropriate drum position would have just passed and it would be necessary to wait ten milliseconds for it to come round again.

If an instruction in the D register is about to be transferred to the C register, Switch A on the left of the C register will be up to provide a transfer path between the two registers. The A switch on the left of the D register will be down and a new instruction will enter the D register from register 4. The reason for providing the D register with a new instruction is that otherwise, an A instruction followed by the test failure of the instruction taken from the D register would leave the machine without an instruction, and cause it to stop computing.

The D register, in conjunction with a 'repeat' instruction, may be used to repeat an A instruction a specified number of times. To do this the main store address of the 'repeat' instruction, made redundant by the automatic inclusion of the W digit, is used as a counting facility and increased by 2 each time the instruction passes to the D register. When the A instruction has been repeated the required number of times the main store address of the 'repeat' instruction overflows into the register address position and provides the address of a new instruction.

When a word is transferred from the control unit to a register, it is transferred via the D register.

10.1.3 The E register

The E register is the execution register. An instruction transferred to the E register remains static for one word-time, and during that time each of the function digits present in the instruction causes a particular operation to be performed in the machine.

The decoded V combinations V1 to V4, and the drum position part of the main store address, have already been examined in the C register and are therefore not transferred. The W digit is transferred because it has a function to perform in the E as well as the C register. An instruction transferred to the E register may therefore consist of - any of the digits A to E; the V combinations V, V5, V6, V7; the W digit; and the eight track selection digits.

10.1.4 Timing in the C and E registers

The timing of events in the C and E registers is as follows:

The least significant digit of an instruction entering the C register enters at time P_{32}, and at time P_0 the whole instruction is in the register and stops shifting.

During the last five normal-word-time P pulses, P_4 to P_0, the five least significant digits of the instruction (those selecting the required drum position) are examined as they shift into the register; so that when the instruction stops shifting, the machine knows whether or not the drum is in the required position.

At time P_0 the track selection digits of the main store address are copied into the E register, replacing the previous track address. The track selection digits are taken before the other function digits in order to give the transistor switches, which select the appropriate track, sufficient time to operate.

At time P_{-1} the test-and-transfer box examines the V digits and determines whether any specified test is satisfied or not. The W digit is also examined because when this is present there need be no wait for a particular drum position.

Also at time P_{-1} the previous contents of the E register are cleared, (except for the newly changed track address) and the instruction AW is set up (10.1.5).

At time P_{-2}, if any specified test is satisfied, and the drum is in the correct position - or there is no need to wait for the drum - the remainder of the instruction in the C register is transferred to the E register.

In the time P_{-3} to P_{33} the E register operates necessary switches, and at time P_{32} the actions specified by the instruction start to take place.

10.1.5 Setting up AW in the E register

When the test on an instruction in the C register is not satisfied, the E register remains in the AW condition set up at time P_{-1}. The A digit puts switches in the correct position for the next instruction to be transferred from the D to the C register, and the W digit, by inhibiting the drum reading and writing circuits, prevents any unwanted action taking place. AW also remains set up when an instruction in the C register circulates while waiting for the required drum position. Here again the W digit prevents any unwanted action taking place during the waiting time.

The setting up of AW in the E register could be made dependent upon the absence of a transfer pulse, but a considerable amount of circuitry is saved by setting it up automatically with the timing pulse P_{-1}. If either A or W are not required in the instruction transferred from the C register they are overwritten at the time of the transfer, i.e. at time P_{-2}.

10.1.6 The 'C' adder

Any instruction entering the C register from a main store location, a register, or the D control register, has to pass first through the 'C' adder, shown as a plus sign inside a circle. It is a simple adding device which, when adding two words, is required to produce sum and carry digits according to the rules of binary addition. Its function is to enable an instruction from a main store location or the D control register to be modified by the addition of an instruction from a register.

10.2 The Arithmetic Unit

To enable it to perform its operations the arithmetic unit consists of two accumulators called the A accumulator and B accumulator; each having associated with it a circulation path, a pre-adder and an adder/subtractor.

10.2.1 The accumulators

The main purpose of the accumulators is to hold partial results, and when two numbers are added or subtracted the result is placed in one of the accumulators. This result may then be transferred to a store location; or it may be circulated in order to be right or left shifted or to have another number added to or subtracted from it.

There are only a few differences between the A and B accumulators. One of the differences is that a number in the A accumulator is provided with an extra digit at the most significant end, which is called the extra sign digit. This digit prevents errors which could otherwise occur in certain shifting operations.

Another difference is that a 1 is added to or subtracted from the least significant end of only the B accumulator when the Q digit is present in an instruction.

The last difference is the 'carry trap', which has a one-way path from the B to the A accumulator. The two accumulators can be linked together and used as a double-length accumulator capable of holding a 67 digit number. In the addition or subtraction of two double-length numbers, which is performed half in the A accumulator and half in the B accumulator, it is possible that a carry/borrow digit will be produced at the most significant end of the B accumulator. This digit is trapped in the carry trap, and is only released to the A accumulator by an instruction containing the function digit V.

For convenience each accumulator is located in a track on the magnetic drum. An ordinary track has a capacity of 32 words and an average access time of five milliseconds; but writing the same word continuously round the track effectively creates a one-word register of immediate access. In case it is required to read the word before it is written right round the track, a separate reading head is provided behind the writing head at a distance of one word length. As the last digit of a word is being written the first digit is being read and circulated back to the writing head. A word circulates in this way continuously, even when it is copied into a store location, and only stops circulating when the accumulator is cleared by the function digit C, which breaks the circulation path.

10.2.2 The circulation paths

The fact that numbers in the accumulators are able to circulate is made use of in the operations of clearing, right and left shifting, and addition and subtraction.

The circulation paths shown above the accumulators contain switches that perform the clearing operation; and associated with them is the B digit, which specifies the appropriate accumulator.

10.2.3 Word shifting

In a normal number the pulse representing the least significant digit coincides with the timing pulse P_{32} and the pulse representing the most significant digit coincides with the timing pulse P_0. If we arrange that the least significant digit pulse occurs at time P_{33} and the most significant digit pulse occurs at time P_1, i.e. each digit pulse in the number occurring one timing pulse earlier, the number will have been effectively right shifted, and therefore halved. This right shift may be more clearly imagined if we think of the number being written, least significant digit first, into a location which is moving from left to right. If we start writing the first digit at time P_{33} and finish writing the number at time P_1 then at time P_0, one timing pulse later and the time at which we would normally have finished writing the number, the location and the number will have moved one digit place to the right. In a left shift operation the writing of the first digit is delayed until time P_{31}, the last digit then being written at time P_{-1}. Each digit will then have moved one place to the left, effectively doubling the number.

A word, having been written into an accumulator, emerges from the reading circuit after a lapse of only 39 timing pulses, i.e. one timing pulse early and therefore right shifted. This right shifted word may then be written into a main store location or register, or be circulated along one of three paths and re-written into the accumulator. These paths are (i) an undelayed path, leaving the word right shifted, (ii) a one-digit delayed path, making the word normal, (iii) a two-digit delayed path, making the word left-shifted.

10.2.4 The pre-adders

Any word entering an accumulator from a main store location or a register has to pass first through a pre-adder, which is a similar device to the 'C' adder and is also shown as a plus sign inside a circle. Words entering an accumulator from a main store location and a register at the same time are first added together in a pre-adder, and the sum then passed on to an adder/subtractor.

10.2.5 The adder/subtractors

All numbers have to pass through an adder/subtractor, shown as a plus and minus sign inside a circle, before entering an accumulator.

A number is added to, or subtracted from, a number already contained in an accumulator in the following way. At the beginning of a word-time the least significant digit of the number in the accumulator, having circulated, enters its adder/subtractor before re-entering the accumulator. At the same time, the least significant digit of the number from the pre-adder also enters the adder/subtractor, and the two digits are added or subtracted and the result passed on to the accumulator. This action is repeated for each pair of digits until the complete numbers have been added or subtracted.

The operation of an adder/subtractor is simplified by the fact that when any two binary digits are added or subtracted the resulting sum or difference digits are the same; the only difference between the two operations being in the carry/borrow digits produced. This means that the same device can both add and subtract, as long as provision is made for all the cases where a carry/borrow digit may occur.

10.3 The Main Store

The main store, located on the surface of the magnetic drum, has a capacity of 8192 words. About 2000 words are permanently recorded in the form of input and interpretive programs, subroutines etc. and the rest of the space is available for ordinary working.

Words are written on to the drum surface in definite positions. The particular track a word is written into depends on which of the read/write heads is selected; and the position a word occupies in any track depends on the time at which it is written, in relation to master timing pulses generated by the drum itself. These timing pulses are produced in groups, thirty-two groups to one revolution of the drum. If the writing of a word coincides with, say, the fifth group of pulses it will occupy the fifth of the thirty-two possible word positions in the track.

The store locations are numbered, for selection by the programmer, from 0 to 8191; the first thirty-two are designated by three figures (000 to 031) to distinguish them from register addresses. Fig. 3 is a representation of part of the Main Store.

The main store address written by the programmer is converted by the machine during input to a thirteen digit binary number. The eight most significant of these digits, designated S8 to S1, are decoded to produce 256 different combinations and are used to select the required track.

The number shown in the corner of each location is the address by which each location is known to the programmer

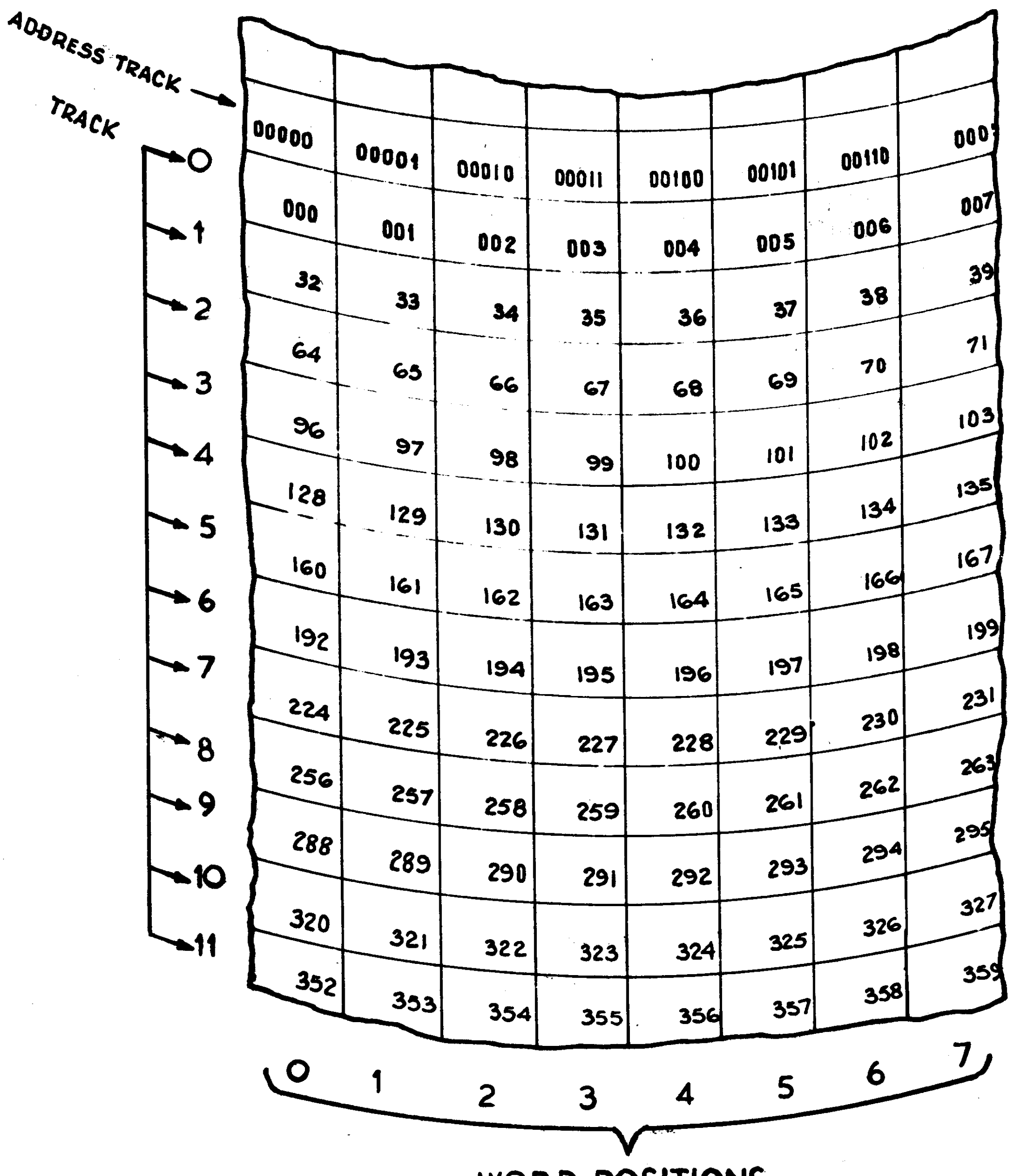


Fig.3.

The other five digits, designated P5 to P1, are decoded to produce 32 different combinations and are used to select the required word position.

10.3.1 Track selection

The binary number formed by the eight track selection digits is divided into two halves, each half producing a pulse which is fed to a matrix arrangement of transistor switches. Fig. 4 shows the selection of track 100 by this means. The eight-digit binary equivalent of decimal 100 is 01100100 and this is divided into two groups. The most significant group produces a pulse in the matrix on the vertical line corresponding to it, and the least significant group produces a pulse on the horizontal line corresponding to it. The transistor switch at the junction of the two pulses is operated and an electrical connection is made with track 100.

10.3.2 Word position selection

In Fig. 3 each word position in the address track is seen to contain a five-digit number, and there are 32 such numbers, running from 00000 to 11111, permanently recorded round the drum. When discussing the C control register we saw that an instruction shifts into it digit by digit. The machine operations are so synchronised that when the digit P1 is passing a particular point in the C register the least significant digit of a number in the address track is also passing a particular point, thus enabling the two digits to be compared. The digits P2 to P5 are then compared in turn with the rest of the digits of the address track number. When each of the five pairs of digits agree, the drum is in the required position for the reading or writing of a word.

10.4 The Registers

In computers a register may be defined as a store, usually of one-word capacity and generally intended for some special purpose or purposes.

Of Zebra's 32 registers, numbered 0 - 31, 12 accord with the above definition.

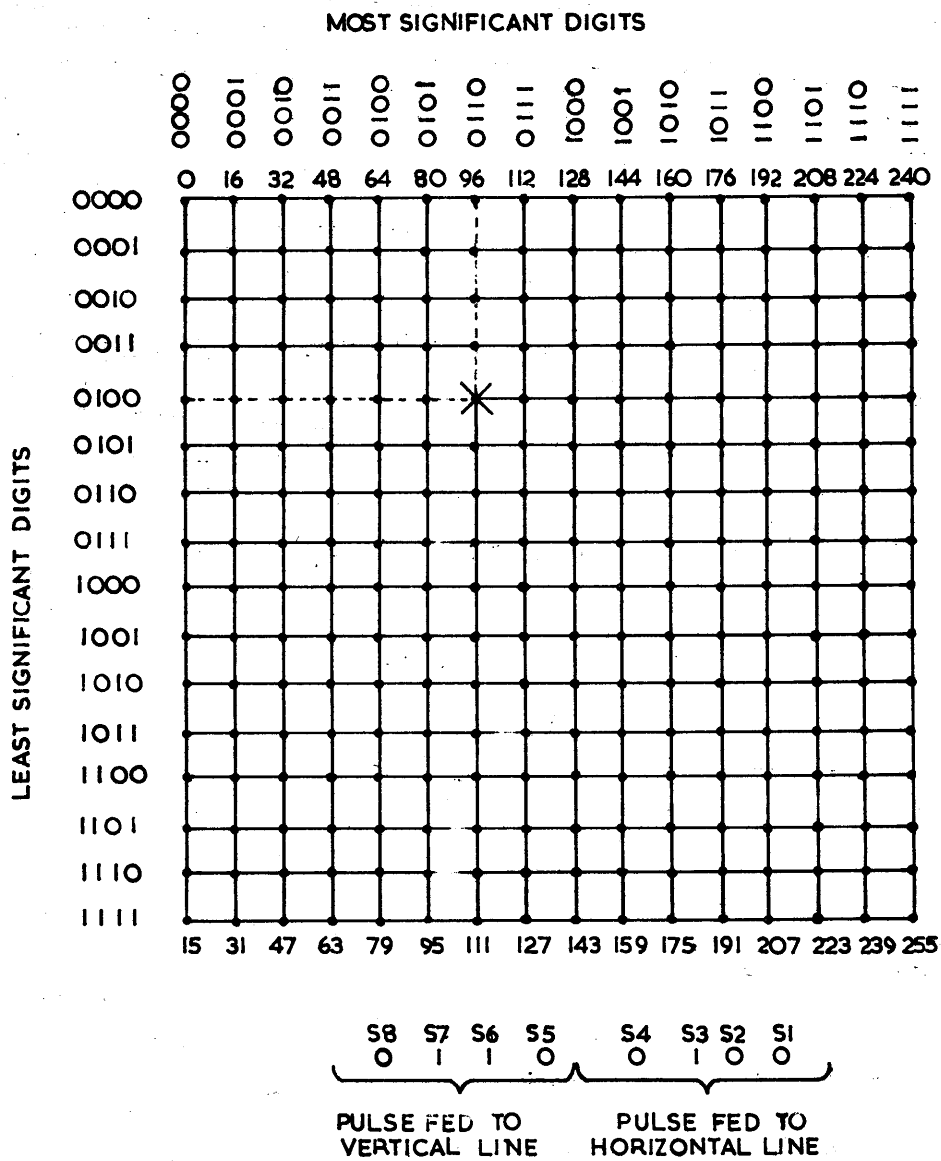

MOST SIGNIFICANT DIGITS
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
0 16 32 48 64 80 96 112 128 144 160 176 192 208 224 240
LEAST SIGNIFICANT DIGITS
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
15 31 47 63 79 95 111 127 143 159 175 191 207 223 239 255
S8 S7 S6 S5 S4 S3 S2 S1
0 1 1 0 0 1 0 0
PULSE FED TO VERTICAL LINE
PULSE FED TO HORIZONTAL LINE

Fig. 4.

10.4.1 Registers

The twelve registers numbered 4 to 15 are stores of one-word capacity and access to them for reading or writing is immediate. Each one is located in a track on the magnetic drum, and the track is converted to a one-word register by the same means as those used for the arithmetic accumulators.

In certain circumstances, the contents of registers 4 and 15 are read automatically. For example, when an instruction in the D register is transferred to the C register the contents of register 4 are transferred automatically to the D register, i.e. without the operation having to be programmed externally. The contents of register 15 are read automatically during the action of the LR and XD combinations.

10.4.2 Pseudo registers

The other twenty registers, called pseudo registers, are of two kinds. Some resemble registers in that information, such as constants, can be read from them immediately. The remainder may perhaps be better called 'register functions' because their selection in an instruction causes operations relating to input and output to be performed.

10.4.3 Register selection

The five digits of the register address, R5 to R1 are decoded to produce thirty-two different combinations, enabling the machine to select any one of the registers.

There are various methods of decoding and we have already seen the matrix arrangement of switches used for main store track selection. For register selection a device known as a 'decoding tree' is used. To illustrate this method we may take as an example the decoding of three binary digits, which produce 2^3, or eight different combinations as follows:-

Binary combination	address
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

REGISTERS AND REGISTER FUNCTIONS SELECTED BY $\overline{E}$

0	read zeros	pseudo registers
1	read least significant one, i.e. a one digit coinciding with time P_{32}	
2	read contents of the A accumulator	
3	read contents of the B accumulator	
4	read contents. Special reading facility	registers
5 to 14	read contents	
15	read contents. Special reading facility	
16 to 21	not yet allocated	
22	read contents of the A accumulator. Set teleprinter signal equal to sign digit of the A accumulator.	pseudo registers
23	read most significant one, i.e. a one digit coinciding with time P_0	
24	read logical product of the A and B accumulators	
25	read contents of the A accumulator. Set teleprinter signal equal to sign digit of the A accumulator.	
26	read 5th hole of tape from staticiser	
27	read 4th hole of tape from staticiser	
28	read 3rd hole of tape from staticiser	
29	read 2nd hole of tape from staticiser	
30	read 1st hole of tape from staticiser	
31	read one row of holes from tape and set up in staticiser. Step tape to next row of holes.	

REGISTERS AND REGISTER

FUNCTIONS SELECTED BY E.

4	write information	registers
5 to 14	write information	
15	write information	
22	set teleprinter signal equal to sign digit of the A accumulator	pseudo registers
24	read logical product of any main store location and register 5	
25	set teleprinter signal equal to sign digit of the A accumulator	
26	set staticiser to punch 5th hole in tape	
27	set staticiser to punch 4th hole in tape	
28	set staticiser to punch 3rd hole in tape	
29	set staticiser to punch 2nd hole in tape	
30	set staticiser to punch 1st hole in tape	
31	punch one row of holes in tape. Step tape to next position.	

Decoded addresses always start at 0. Only in this way can all the possible combinations be produced from a particular number of digits. Eight addresses numbered 1 to 8 would require four digit positions instead of three because 8 in binary is 1000.

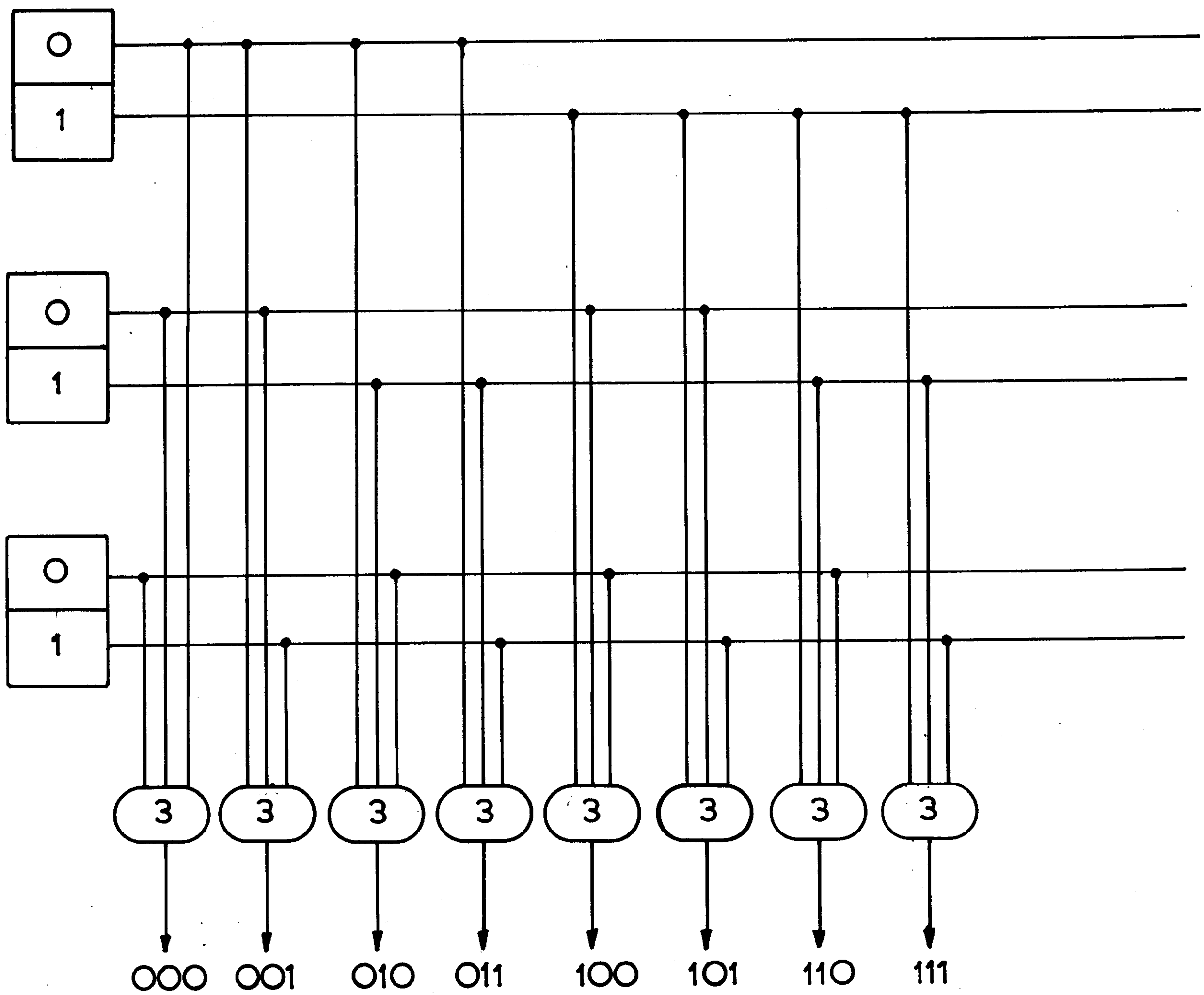


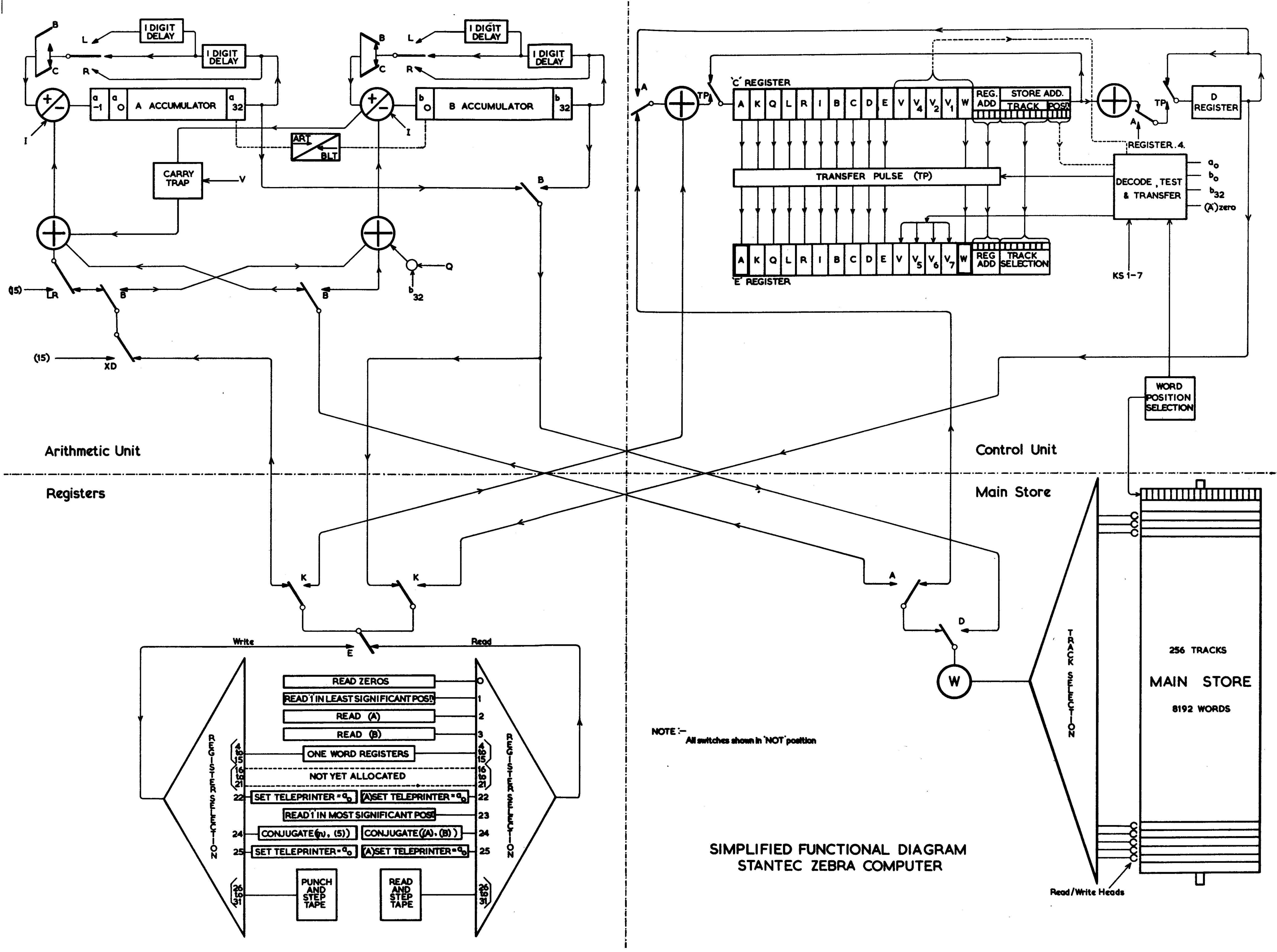
Fig. 6.

Fig. 6 shows a decoding tree with three binary inputs and eight outputs. The three rectangles on the left of the diagram represent binary digits which, of course, may be 1 or 0. Each possible combination of the three digits is made the input to one of eight three-input 'and' gates shown in the lower part of the diagram, and each gate provides an electrical connection to an appropriate point.

11. CONCLUSION

Zebra is a serial/binary computer able to perform mathematical calculations, and process data, with speed and accuracy. It does this by storing data in its magnetic drum main store and registers; and by passing units of this data, in the form of trains of electrical pulses, from one to another of its four basic units - Main Store, Registers, Control and Arithmetic. The technique of switching and general control of these pulses is called the Logic of the machine.

As basic operations, Zebra adds or subtracts in 312 microseconds. Multiplication and division, which are programmed, take 11 milliseconds and 35 milliseconds. However, the important thing to remember in considering the speed of the machine is that several operations nearly always take place in one word time, and that as a result the average time per operation is only 2 milliseconds. Another point to be emphasised is the speed attained in 'simple code' operation. This instruction code, useful when simplified programming or floating point operation are required, is so designed that operations are about one fifth as fast, sometimes even half as fast, as operations in normal code. The last point we wish to make is that Zebra can be integrated with other data processing devices without internal modification. For these reasons Zebra may justly be claimed to be a powerful and versatile machine.



SIMPLIFIED FUNCTIONAL DIAGRAM
STANTEC ZEBRA COMPUTER