AN INTRODUCTION TO STANTEC ZEBRA

STANDARD TELEPHONES AND CABLES LTD.

INFORMATION PROCESSING DIVISION,

NEWPORT, MON.

1-6/59.

INTRODUCTION

The basic operations performed by a digital computer are the arithmetical ones of addition and subtraction. Some machines have built-in facilities for multiplication and division but others, like Stantec Zebra, use a combination of addition and subtraction and right and left shifting.

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.

As electronic computers are able to perform vast quantities of arithmetic quickly, they make it possible to tackle problems which would previously have taken a prohibitively long time. Since speed is one of the computer's main attractions it is natural for users to want it to be as fast as possible - but high speed usually means high cost. In Zebra a successful combination of medium-high speed and low cost has been achieved by an ingenious logical design which, in conjunction with a unique instruction code, enables up to 9 machine operations to be performed at one time.

The first part of this booklet describes the main features of Zebra's functional design, and the second part is a brief description of the basic parts of the computer and the way they work.

GENERAL DESIGN FEATURES OF ZEBRA

Stantec Zebra is an electronic automatic digital computer designed and built in accordance with the following principles:

Technical Simplicity and Economy of Equipment is achieved by using:

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

Optimum Speed of Operation is achieved by:

(a) The provision of a quick-access store
(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
(c) A combination of functional design and instruction code which makes it possible to perform as many as 9 machine operations at one time. Zebra's basic word-time is 312 microseconds, and thus two numbers may be added or subtracted in that time, but at the same time several other operations can be taking place. Therefore any assessment of Zebra's actual speed must be based on the parallel nature of its operations, inherent in its instruction code.

Extreme Flexibility of the Instruction Code is achieved by using an instruction word having a large number of function digits, most of which are not decoded. The separate action of these digits makes possible the parallel operation already referred to, and means that an average of five operations may take place for each instruction. There are 15 function digits, and this means that thousands

of different instructions may be formed.

Reliability has been increased by designing Zebra as a computer of medium size and medium electronic speed, and this is for two reasons. First, the components in any machine have a certain failure rate, however low improved manufacturing techniques may make them, and the smaller the total quantity of equipment the lower the failure rate of the machine as a whole. Second, 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.

Ease of Maintenance is enhanced by a relatively simple functional design, and the fact that only a few different types of functional unit are used.

Of great assistance in maintenance is Zebra'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).

Adequate Storage Capacity is provided by a main store of more than 270, 000 binary digits, and a high speed store of about 400 binary digits.

Ease of Programming is provided by the Simple Code. The Normal Code, because of its power and flexibility, requires a certain amount of skill and experience in the programmer to make the fullest use of it. The Simple Code, on the other hand, can be learned and applied quickly, and although some operation speed is lost in its use it makes the machine available to the comparatively unskilled

programmer. Other valuable uses of the Simple Code are - for the writing of one-off programs, which can be written more quickly in Simple Code than in Normal Code; for calculations in which it may be impossible to ensure that numbers remain fractional or integral throughout the calculation; for avoiding the necessity of scaling. The last two facilities are available because all numbers in Simple Code calculations are represented in floating-point form.

Simplicity and Economy of Input and Output results from the use of punched paper tape. For input, a Ferranti photoelectric tape reader is used, operating at a speed of 100 characters per second. For visual output a Creed teleprinter prints at 7 characters per second, and for punched tape output a Teletype punch operates at a speed of 50 characters per second.

INFORMATION IN THE MACHINE

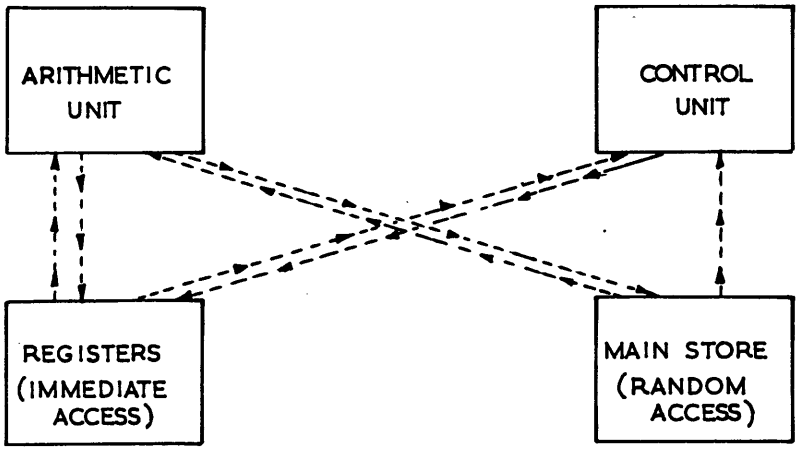
For solution by a computer, mathematical problems are broken down into a sequence of simple arithmetical operations. In the solution of any problem, or in the processing of data, the machine has to be provided with the data on which operations are to be performed and instructions which control these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

In the machine both numbers and instructions are in the form of binary digits. That is, they are made up of the digits 1 and 0. The binary digits are themselves represented by trains of electrical pulses, by patterns of magnetisation, or by electronic devices called staticisers. Whether a number is large or small, or an instruction long or short, it will be represented by a set number of binary digits - a unit of information. This unit of information is called a word, and in Zebra the length of a word is 33 digits.

Each digit in a binary number represents a power of 2. A binary number of 33 digits is equivalent to a decimal number of 9 - 10 digits, and can represent integers in the range +4000 000 000 to -4000 000 000 or fractions in an equally wide range.

The code letters present in a written instruction are represented by 1's, and those absent are represented by 0's.

Words may be stored in the machine for an indefinite time, and may be passed from one part of the machine to another. A computer has three basic parts - an arithmetic unit, a control unit and a store. Zebra has both a random access store and an immediate access store, and the machine is best thought of as consisting of four basic parts:



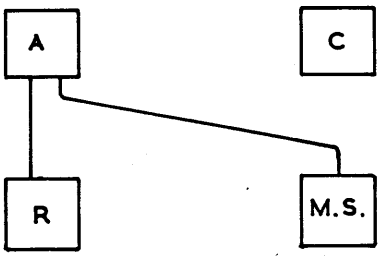
Any word transferred to the Main Store or Registers is automatically stored in a particular location, specified by the instruction causing the transfer. Any word transferred to the Arithmetic Unit is either added to or subtracted from the existing contents, or it replaces the existing contents. Any word transferred to the Control Unit will be an instruction, and if it includes test digits its execution will depend on certain conditions in the machine.

THE ZEBRA INSTRUCTION WORD

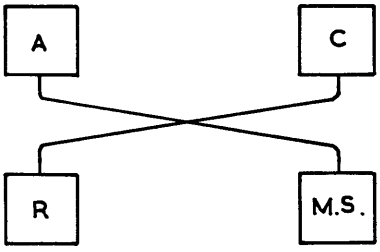
An instruction, when contained in the machine is in the form of a 33 binary digit word. So that particular store locations may be specified, provision is made for two addresses - a Main Store address and a Register address. These addresses are written by the programmer as decimal numbers, and are converted in the machine to binary numbers, taking up 18 of the 33 digits.

The remainder of the instruction word is made up of 15 function digits. When writing instructions, the programmer may write the letters representing the function digits in any order, and he omits the letters not required. A is an exception to both these rules, as it must be written first, and if not required must be replaced by X. This is to enable the machine to separate one instruction from another when reading a list of them from the input device.

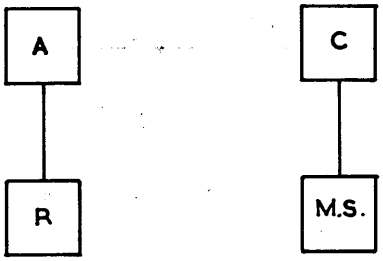
In an instruction word the function digits corresponding to the letters written in any instruction represent 1's, and the digits corresponding to the letters omitted represent 0's. In this way the function digits control the switching actions that cause the various machine operations to take place. For example, the interconnection of the four basic units, and the direction of flow of information between them, is controlled by A, K, D and E. When A = 1 the Main Store is connected to the Arithmetic Unit, and when A = 0 the Main Store is connected to the Control Unit. When K = 1, the Registers are connected to the Control Unit, and when K = 0 the Registers are connected to the Arithmetic Unit.



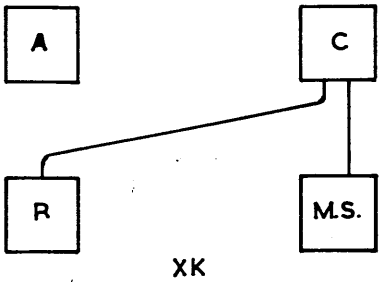
A



AK



X



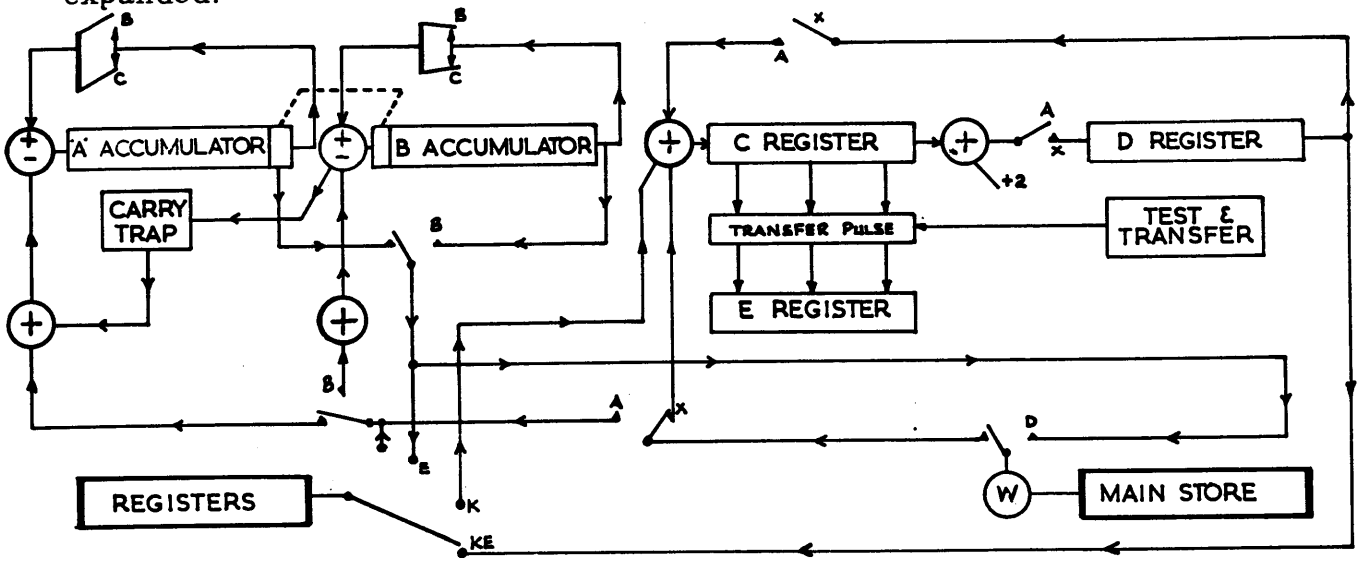
XK

In each of these four combinations 'reading' from the stores has been taking place. 'Writing' into the Main Store occurs when D = 1, and 'writing' into the Registers occurs when E = 1.

There are two accumulators in the Arithmetic Unit, which are called 'A' and 'B'; when B = 1 the 'B' accumulator is selected, and when B = 0 the 'A' accumulator is selected. When Q = 1 a binary 1 is added into the least significant position of the 'B' accumulator. This is in addition to any other arithmetic that might be taking place in 'A' or 'B'. When I = 1 subtraction instead of addition takes place in the Arithmetic Unit. When L or R = 1 left or right shifting of the accumulators takes place. This has the effect of multiplying or dividing the contents of the accumulators by 2. When C = 1 the accumulator specified by B is cleared. The four V digits are decoded into 15 combinations, four of which are used to test (1) the sign of the number in 'A', (2) the sign of the number in 'B', (3) whether or not the number in 'A' is zero, (4) whether or not the least significant digit of the number in 'B' is a 1. Other combinations are used in association with manual keys. An instruction in which W = 1 is executed without any delay, and connection between the Main Store and other units is broken.

THE BASIC FUNCTIONAL UNITS

We come now to a short description of the four basic units. It will help in visualising the actions taking place in the Arithmetic and Control units if those parts of the block diagram are expanded:



CONTROL UNIT

The functions of arithmetic, control and storage are equally important, but it is control allied to storage that distinguishes an electronic digital computer from previous forms of calculator. If the Store may be considered as the 'memory' of the machine, the Control unit is the 'director of operations'. In some machines it is only possible to execute programmed instructions in a fixed sequence from the beginning to the end of the computation, but the power and flexibility of Zebra make it possible to:-

(a) Execute instructions in sequence or in any desired order

(b) Make the execution of an instruction conditional upon tests on the contents of the Arithmetic unit

(c) Automatically provide an instruction to replace one which fails to meet a specified condition

(d) Modify an instruction (by adding another to it) during computation

(e) Repeat an operation a specified number of times before going on to the next instruction.

To enable it to do its work the Control unit is provided with 3 registers - C, D and E, and an adder. The D register has a counter associated with it.

The C Register

After an instruction has entered the Control unit there are two things to be decided about it, (a) whether it is to be executed at all, and if it is (b) when it is to be executed. The first decision is necessary because some instructions are conditional, and their execution depends on the contents of the Arithmetic unit, or on the position of keys set by the operator. The second decision is necessary because the Main Store is located on the surface of a rotating cylinder, and if the instruction requires access to the Main Store it may be necessary to wait until the cylinder is in the appropriate position. While these decisions are being made the instruction is held in the C register.

If the instruction is to be executed, then at the appropriate time it is transferred in parallel, i.e. all the digits at the same time and therefore almost instantaneously, to the E register.

The E Register

The E register is the execution register, and an instruction transferred to it remains there for one word-time, i.e. 312 microseconds. During this time each of the function digits in the instruction causes its particular operation to be performed in the machine.

The D Register and Associated Counter

We have seen that the absence of A in an instruction is denoted by X, and that X connects the Main Store to the Control unit. As in this case the flow of information is in one direction only, from Main Store to Control unit, X can be interpreted as 'take your next instruction from', so that X100, for example, would mean 'take your next instruction from the Main Store location whose address is 100'.

Every time an X instruction is transferred to the E register for execution it is also transferred serially to the D register, and on the way its Main Store address is increased by 2. Now if at any time during a computation the Control unit is not 'told' where to take the next instruction from, the last instruction to have entered D returns to C.

A instructions are not transferred to D because they do not specify the address of a new instruction and are therefore valueless as 'return' instructions. In fact, one of the reasons for having the D register is to provide a new instruction when an A instruction is currently being executed. The other reason for wanting a 'return' instruction available is that conditional instructions which are not carried out are rejected, and without some provision like the D register the Control unit would be left without a new instruction.

The D register is also used in 'repeat' instructions, passing the 'repeat' instruction back to C until it, and the A instruction it has been bringing in, have been executed the required number of times. At this point the address of the 'repeat' instruction is automatically increased, so providing the address of a new instruction.

The C Adder

The C adder enables an instruction from a Main Store location, or the D register, to be modified by the addition of an instruction from a Register. This is known as the 'B line' or 'order modification' facility.

MAIN STORE

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. This is a hollow brass 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 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, and by spacing heads along the length of the drum a large number of tracks may be formed.

Most of these tracks, in fact 256 of them are used by the Main Store. As 32 words can be got into a track, the Main Store has a capacity of over 8,000 words. Some of this storage capacity is normally used to store the Normal Input, Output and other interpretive programs, subroutines etc., and the rest of the space is available for ordinary working.

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 designated, for selection by the programmer, by numbers from 0 to 8191; the first thirty-two are written with three figures (000 to 031) to distinguish them from register addresses.

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 are decoded to produce 256 different combinations, and are used to select the required track. The other five digits are decoded to produce 32 different combinations, and are used to select the required word position.

REGISTERS

The average access time to the Main Store is 5 milliseconds, and although 'waiting for the drum' can be reduced by optimum programming, there is a need for a small high speed store. The need is met in Zebra by using special tracks on the drum, but although the Main Store and Registers both make use of the drum, it is better to regard them as separate entities.

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, designated 0 - 31, 12 accord with this definition, being stores of one-word capacity and immediate access. They are designated 4 - 15.

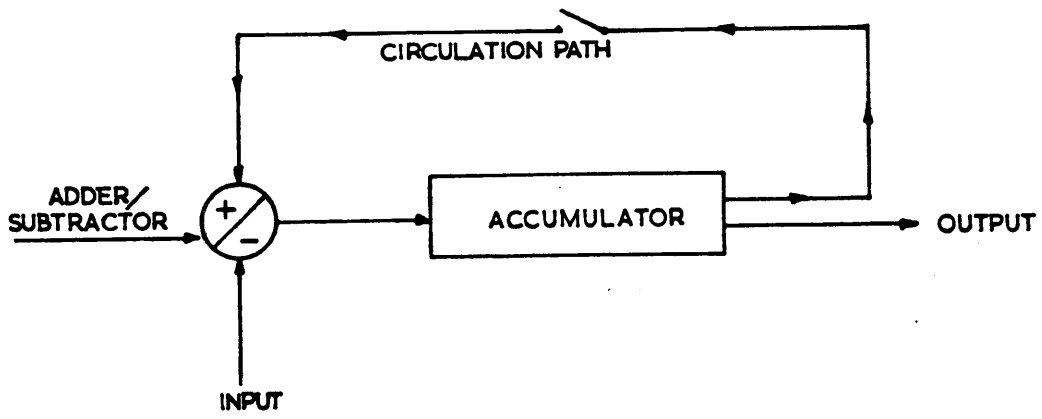
The other twenty registers, called pseudo registers, are of two kinds. Some resemble registers in that information, such as constants, can be read from them immediately. The remainder may perhaps be better called 'register functions' because their selection in an instruction causes operations relating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

ARITHMETIC UNIT

The essential requirement of the Arithmetic unit is that numbers may be added and subtracted in it. This is easily managed by connecting an electronic adder/subtractor to what is in effect a register, the combination being called an accumulator. To this simple arrangement several refinements are made.

Clearing and Shifting

There should be a facility for bringing a number into an accumulator so that it replaces the existing contents, instead of being added or subtracted. Like the Registers, the accumulators use special drum tracks, and are such that information in them is continuously circulating.



To clear an accumulator it is simply necessary to break the circulation path, and write in either zeros or new information.

The circulation of information is also put to use in another way. First it must be explained that each digit in a

word, whether instruction or number, has its own particular significance because the train of pulses representing the word is referred to a group of standard timing pulses. A number in the circulation path of an accumulator may be advanced or delayed so that it shifts to right or left in relation to the standard timing. This shift is equivalent to moving a decimal point or, as Zebra works with numbers in binary notation, a binary point to left or right. The maximum shift in one word-time is one digit position, so that in each word-time a number may be multiplied or divided by 2. This shifting and adding process is used in the full multiplication program, provided by a standard subroutine which is normally stored with the Normal Input program and may be called in at will. There is a similar subroutine for division.

Double-length Arithmetic

The two accumulators are linked together and may be used to perform arithmetic to 'double-length' accuracy. In shifting operations, digits which would otherwise be lost overflow from one accumulator to the other, although this overflow can be prevented if desired. When arithmetic is done in 'B', it is also possible to control the transfer of any 'carry' into 'A'. It is not always desired to allow this 'carry' to pass over to 'A' so it is trapped, and only allowed into 'A' when the appropriate instruction is given.

Each accumulator is equipped not only with an adder/subtractor but with a pre-adder, which allows numbers which come from a Main Store location and a Register at the same time to be added, and the sum then to be added to or subtracted from the contents of the accumulator.

CONCLUSION

In this booklet we have tried to give some idea of how Zebra works. Further information about the machine and its use can be got from the several other publications available.

To sum up, Zebra is a serial/binary digital 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 from one to another of its four basic units - Arithmetic, Control, Main Store and Registers.

Zebra adds or subtracts in 312 micro-seconds. 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 in nearly every word-time several operations will be performed. 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 cleverly designed that operations are about one fifth as fast, sometimes even half as fast, as operations in Normal Code.