T.W.J. POWELL

DEPARTMENTAL MEMORANDUM

FROM: M.J.Wingstedt

DATE: 29th May, 1962.

NEWMAN STREET MJW/AK

TO: Supervising Computer Engineers.

REPORTING FAULTS

When total fault repair time exceeds one hour in any one day, a report must be made immediately by telephone to your Area engineer. It is every engineer's duty to see that this is done and not left to the senior engineer who may not be available at the time.

In addition, if a computer has been working under limited conditions, (e.g. only one tape reader, less than full complement of tape decks, I.A.S. not operational) the above rule still applies.

If the Area engineer cannot be contacted then a report can be made to Mr. D.W. Potts at West Gorton (EAST 1301) or Miss A.P. King at Newman Street (MUS 5040).

The data required when reporting the fault is as follows:

(a) Machine Type
(b) Location
(c) Date and Time of Breakdown
(d) Date and Time Reported
(e) Details of Fault

On completion of repair the following data should be reported:

(f) Date and Time of Completion
(g) Total Operating Time Lost
(h) Cause of Breakdown

If breakdown occurs out-of-normal factory working hours, a report can be made to the Duty Engineer, Newman Street (MUS 0310) during the following hours:

Mon. - Fri.	Up to 22.00 hours.
Sat., Sun.	08.00 - 16.30 hours.

If it is impossible to make a report under any of the above-mentioned arrangements, a report must still be made at the earliest possible opportunity to your Area Engineer.

M J Wingstedt

FERRANTI PEGASUS COMPUTER

VOLUME 1

OPERATING INSTRUCTIONS

FERRANTI LTD.
Head Office: HOLLINWOOD, LANCASHIRE,
ENGLAND

COMPUTER DEPARTMENT

Offices and Works:

WEST GORTON WORKS
THOMAS STREET
WEST GORTON
MANCHESTER, 12

Tel: EAST 1301

London Computer Cent

21, PORTLAND PLACE
LONDON W.1

Tel: LANgham 921

Copy No. 127

SUMMARY

This volume is intended as a general introduction to the rest of the handbook. The system is described in general terms, and the uses of the various controls are explained. Instructions are given for punching tapes that are to be taken in by the standard form of Input, and a rather special programme, the Bootstrap, is explained fully, primarily as an exercise in following the detailed working of the machine.

A glossary of terms and a list of symbols appear at the end of the volume.

First issue AUGUST 1956

Printed in England by Specialised Printing Services Limited, 30/34, Langham Street, London, W.1

CONTENTS

CHAPTER I

INTRODUCTION

CHAPTER I

INTRODUCTION

1. A computer, like any other machine, can operate only by following the instructions of a human agent. But, whereas instructions can be fed into simple machines by manipulating control knobs, levers or switches, a digital computer does not lend itself easily to direct control. If its high speed of operation is not to be vitiated, the computer must have access to instructions as fast as it can obey them - at a rate of some four thousand or so per second for Pegasus. Computers are therefore provided with a 'memory' that enables them to store all the necessary instructions (and other data) at the beginning of a computation so that orders can be obeyed rapidly in rotation. Most of the instructions will be obeyed several times in the course of a single computation.

2. The complete set of instructions and numerical data required to solve a particular problem is called a 'programme'. Any general part of a programme that can be applied in other programmes is called a 'routine' or 'sub-routine'. Individual instructions causing the machine to perform single operations are called 'orders'. Programming consists in devising a method of solving the problem by means of the elementary arithmetical operations that the machine can perform, deciding what orders are necessary to cause it to perform the operations, and formalising the orders in a numerical code appropriate to the machine.

3. The machine 'reads' the programme initially from punched paper tape by means of a photoelectric tape reader. Each row of holes across the tape, according to the number of them and their positions, represents a letter, a numeral or one of the other characters used by the programmer. The form in which the programme is punched on tape is not quite the same as the form in which it is stored in the machine. The punching code has been chosen so as to make the task of writing programmes, and translating them into tape form, as easy as possible, and to enable them to be printed out recognisably by means of a teleprinter interpreter unit. It is left to the machine itself to perform the tedious process of compiling the orders in a form that it can interpret; it does this by obeying a permanently stored routine known as 'Input'. It should be appreciated that the machine never obeys orders direct from tape: orders are always obeyed from store.

4. Operations in electronic digital computers are carried out in the 'binary' scale of numeration, in which numbers are expressed in terms of the powers of two, instead of, as in the common 'decimal' scale, the powers of ten. Thus two is expressed in the binary scale as 10, four as 100, eight as 1000 and so on. Five in the binary scale would be 101 (four plus one), seven would be 111, and ten 1010. Binary numbers are represented in the machine by sequences of electrical pulses, the presence of a pulse representing a 'one' and its absence a 'nought'. Binary information is the only kind of information that the machine can interpret; consequently orders must be stored as binary numbers, though of course they are not written by the programmer in this form. An introduction to binary arithmetic will be found in Volume 2 of this handbook.

6. The main store in Pegasus is a rotating drum coated with magnetic material on which digital information is retained in the form of a magnetisation pattern. Information is 'written on' or 'read from' the drum by means of magnetic recording heads. This type of store has quite a large capacity; but it suffers from the disadvantage of a long access time. Whenever information is to be transferred to or from the main store, the machine must wait for a period that may be quite long in terms of computer operations until the part of the magnetisation pattern required comes under the recording heads. Consequently the main store is supplemented by an immediate-access 'computing' store, which contains the orders and numbers in current use, and which can be replenished from the main store when its contents have been exhausted in a particular context.

7. Orders are stored in pairs, a pair of orders constituting a 'word' which is the standard unit of information in the computer. Numbers, too, generally occupy one word each. The computing store is made up for the most part of one-word registers. These are of the type in which digit pulses are transmitted in sequence down a nickel-wire line in the form of a compression wave. The pulses are detected, amplified and reshaped at the remote end of the line and are then fed back again to the input end. The length of the line is adjusted so that one word-length of pulses can be kept circulating in this way until the information is required for use elsewhere in the computer.

8. Every one-word location in the main store and the computing store is allotted a number known as its 'address'. The programmer can then call for an operation on a particular number in store by specifying its address in an appropriate order. Thus a typical order might say, 'Take the number in address x, add it to the number in address n, and replace the resulting sum in address x'. As all addresses must be expressed ultimately in binary form, the most economical way of grouping storage locations must be according to some power of two. In fact the storage addresses in Pegasus are arranged in blocks of eight; three binary digits, the eight possible combinations of which will represent the numbers from nought to seven, are then necessary to specify a particular location within a block. Addresses in Pegasus are generally expressed as a block number with a position number.

9. A schematic diagram of the Pegasus system will be found at the end of this volume. The computing store, which is shown at the right-hand end of the diagram, will be seen to comprise six blocks of eight 'ordinary' registers, a group of 'special' registers, and seven storage locations known as 'accumulators'. Order pairs are stored only in the ordinary registers. They are transferred in turn to the 'order register' in the control unit of the machine, from which they are able to control operations. Orders are obeyed in a set 'rhythm', which is controlled by the timing circuits shown in the diagram to the left of the order register. This rhythm is made up of 'bars', each comprising three 'beats', C, A and B. The order pair is transferred from store to the order register during beat C. The first order in the pair is obeyed during beat A; it is then deleted and control passes to the second order in the pair, which is obeyed during beat B. Beat C comes up again after the second order has been obeyed, causing this order to be replaced in the order register by the next order pair.

10. Beat C lasts for approximately 127 microseconds; this is the time interval that would be occupied by one word-length of digit pulses, and is referred to as a 'word-time'. The durations of beats A and B depend on the operations to be performed,

but are always an integral number or word-times. The first word-time in beat A or beat B is referred to as word-time AD or BD; during this word-time, the number or numbers to be operated on (the 'operands') are normally being routed from the computing store to the part of the machine where the operation is to be performed. (Because of the way in which the timing circuits are constructed, beat C can be referred to as word-time CD.). The last word-time in beat A or beat B is referred to as AE or BE; during this word-time, usually, the result of the operation is being routed back to store. During AE the first order of the pair is deleted and the second order is 'shuffled' so that it takes up the timing of the first order.

11. Each order is simply a binary number 19 digits long; but it is interpreted by the machine as it is written in the programme, as four separate numbers. The first number, consisting of seven binary digits, and the second number, consisting of three binary digits, are usually to be interpreted as addresses in the computing store. The third number, consisting of six binary digits, prescribes, according to a special code, the 'function', i.e. the type of operation that the computer is to perform. The last three digits are used in connection with a special facility for 'modifying' orders.

12. When an order is obeyed, the pulses representing the digits in the order are made to 'set' special circuits called 'staticisers', which store the information contained in the order, digit by digit, for as long as it is required. The outputs of these staticisers, and combinations of the outputs, are then made to open electronic 'gates', through which the operands can pass from their locations in store to the arithmetical circuits, and through which the result of the operation can pass back to store. The routing gates cannot be controlled direct from the order register, as the digit pulses there are in circulation. It is necessary to 'staticise' the information, i.e. make a stationary copy of it, so that the gates can be held open continuously. The process of generating 'gating' signals from the digital information in orders is called 'decoding'.

13. The first (seven-digit) number in the order is usually referred to as the N-address. It may be any address in the computing store. The second (three-digit) number, referred to as the X-address, is usually one of the accumulators, which are therefore sometimes referred to as X-lines. The N-digits and X-digits are usually interpreted by the machine as addresses in the way described, but not always. The way in which the machine interprets them depends ultimately on the six function digits (F-digits) in the order.

14. The function code used in Pegasus has been designed to combine ease of programming with the greatest possible economy of equipment. To this end, the functions have been arranged in eight groups with up to eight functions in each, the first three F-digits in the order specifying the group, and the last three the particular function within the group. Functions are provided for simple arithmetical operations (addition, subtraction, multiplication, division and 'shifting', i.e. multiplying or dividing by moving the position of the point) and for certain logical operations that are used principally in collating orders. Transfers between the main store and the computing store, and between the accumulators and other parts of the computing store are under the control of orders, and the necessary functions are provided. There is a group of 'jump' functions, which enable the machine to take alternative courses of action

according to circumstances. There are also functions for use in the special applications of 'double-length' and 'floating-point' arithmetic, and a function enabling the machine to make a selection between several input and output devices.

15. Function 77 says 'stop'. This does not mean that the power to the machine is switched off; but merely that the beat-timing circuit is 'inhibited', i.e. made inoperative, so that the machine cannot obey any more orders. There is also an 'optional' type of stop under the control of the programme. The digit place preceding the first order of a pair is normally occupied by a binary 'one'; but the programme can be arranged so that this place is occupied by a 'nought'. When this is so, the machine will stop after the order pair has been transferred to the order register, and before the first order of the pair is obeyed.

16. Automatic 'internal' stops have been provided to guard against programme errors: the machine will stop if an order contains a function to which no 'meaning' has been allocated, or if a transfer to the main store is called for after an out-of-range number, i.e. a number longer than the standard word-length, has been formed as a result of 'overflow' after an arithmetical operation. The machine will also stop if the 'parity' of (the number of 'ones' in) a word in store has been changed owing to an electrical fault, and, momentarily, if an input or output operation is called for while part of the input or output equipment is 'busy'. The optional stop can be inhibited with a switch on the programmer's panel on the desk; the other stops can be inhibited with switches on the engineer's panel, which is recessed below a flap on the desk top.

17. Orders are normally obeyed in the sequence in which they are stored; this is assured by means of the 'order number', which gives the address (one of the ordinary registers) of the order pair currently being obeyed. The order number is stored in the 'order-number register', which can be seen above the order register in the schematic diagram, and is fed into the order register during word-time BE through the 'order adder' where its value is increased by unity. It is then decoded in the same way as the N-address of an order, to produce signals that will open gates associated with the appropriate register. Increasing the order number by unity before it is decoded ensures that the next order pair to take control will come from the next address and will therefore be the next order pair in the sequence. There is an exception to this procedure in 'jump' orders, which will now be described.

18. The 'jump' or 'conditional transfer of control' is one of the facilities that makes for the great adaptability of electronic computers, enabling them to choose their future operations according to the results of past operations. 'Jump' orders instruct the machine to test for a particular condition, e.g. to test whether a number is negative, or whether it is non-zero; if the condition is fulfilled, the machine is to obey next an order in an address specified in the 'jump' order; if the condition is unfulfilled, the machine is to continue with the normal sequence of orders. The machine can be made to obey a series of orders repetitively by placing at the end of the series a 'jump' order transferring control back to the first order in the series. In this way a 'loop' will be set up, which will be 'cycled' until the 'jump' condition is no longer satisfied. The N-address in a 'jump' order in Pegasus specifies the next order to be obeyed if the 'jump' condition is satisfied; the N-digits are routed into the order-number register at the end of the 'jump' order and thus take the place of the old order number.

19. The tape reader operates at a theoretical maximum rate of 200 characters per second. As there are, on the average, some ten characters to an order for normal input, the machine will take in orders at a rate of about 20 per second, which is slow in comparison with the rate at which orders are obeyed. A great deal of 'input' time can be saved, however, by the facility that the machine has for 'modifying' orders. For example, if the machine is required to carry out the *same* group of operations on a series of numbers stored consecutively, it is not necessary to repeat the whole group of orders for *each* number in the series; instead it can be arranged for the addresses in the orders to be 'modified' before the orders are obeyed, to make them specify each number in turn.

20. Modification, then, requires a number to be added to the address digits of an order before that order is obeyed. This number, called a 'modifier', is held in one of the accumulators in the computing store, where it will have been placed by a previous order. A modifier is called for by the last three binary digits of an order, the M-digits, which give its address. The M-digits are decoded just before the order beat to give gating signals that route the modifier from store on to the accumulator output line, the X/M-bus, whence it passes to the M-bus, and thence to the order adder. As modifiers are sometimes used to extend the length of an order, they are delayed by three digit-times or six digit-times, according to the function in the order, before addition.

21. Modifiers are usually used in conjunction with a special type of 'jump' order called a 'unit modify' order to make the machine carry out the same series of operations on several numbers stored sequentially. The unit-modify order is used in the usual way to set up a loop; but it also causes the modifier to be increased by unity so that the next time the loop is cycled it will be the next number in the sequence that will be operated on. When the loop has been cycled eight times, a complete block of numbers will have been 'used up'; the unit-modify order then does not cause a jump, but allows control to pass to the next order in the normal way, enabling a new block of numbers to be brought from the main store into the computing store.

22. Associated with unit-modify orders are 'unit-count' orders. These operate on a counter, which is a number stored in an accumulator and used to count repetitive processes. Unit-count orders, being 'jump' orders, can be used to set up a loop in the usual way; they also cause the counter to be reduced by unity before the jump takes place. When the counter has been reduced to zero, there is no jump, and control passes to the next order in the programme.

23. The main arithmetical unit of the machine is called the Mill. This is shown at at the bottom centre of the schematic diagram. The Mill contains facilities for addition and subtraction, for multiplication and division by powers of two by 'shifting', and for testing for overflow and 'jump' conditions. It will accept numbers from the N-bus (from any location in the computing store) or from the X-bus and X/M-bus (from one of the accumulators). It will also accept 'counter' numbers, which are prescribed in orders and therefore derived from the order register, and, in special circumstances it will accept numbers from the multiplier-divider. The Mill 'operands' (numbers to be operated on) are selected by a gating circuit, which is controlled by the F-decoding signals.

24. The multiplier-divider in Pegasus is built around accumulators 6 and 7. An auxiliary register is provided to hold the multiplicand or divisor. Multiplication and division orders are timed by the beat counter, which works in conjunction with a special timing circuit in the multiplier-divider.

25. Transfers between the main store and the computing store can be made in single words or in blocks of eight words. The required part of the drum surface is selected by a switching arrangement, which brings into operation the recording heads appropriate to the selected 'track' and by a 'coincidence' circuit, which enables the transfer to take place as soon as the angular position of the drum is correct. The coincidence circuit tests for equivalence between part of the address in the order and a number derived from the 'address track' of the drum. A facility is provided whereby characters representing drum addresses can be punched on the output tape to give the operator some idea of the stage that the computation has reached.

26. The digit frequency in Pegasus is locked to the rate of rotation of the drum with the aid of 'clock' and 'reset' waveforms, which are derived from a 'clock track' on the drum, suitably shaped, and applied to the 'delay' packages throughout the computer. The 'word' frequency is controlled by a 'commutator', which is locked to the signal derived from the address track. The rate of rotation of the drum is standardised by comparing the clock frequency with that of a crystal oscillator, and using the error to control the drum-motor supply.

27. Information in Pegasus is generally stored with odd parity, i.e. with an odd number of 'ones' in each word, an extra 'one' being added if necessary at the end of the word. Parity-count circuits attached to the buses correct the parity of new words fed into store and check the parity of words routed out, causing the machine to stop if the parity of a word has changed owing to an electrical or mechanical fault during storage. The tape characters normally used in programmes also have odd parity (an odd number of holes), enabling a fault in the reading mechanism to be detected. The parity counter on the input bus is arranged to correct the parity of information fed to the punch.

28. The input and output units are treated in orders as addresses (special registers) in the computing store. Output is by punched tape; but this can be fed into an interpreter unit, which causes the information to be printed out on a teleprinter. An 'external-conditioning' function is provided to enable the machine to make a selection among several input and output devices. The selection is done by relays, which are controlled by N-decoding signals. A set of handkeys is provided to enable a single order to be fed into the machine - to modify a programme or for testing. The handkey information can be fed direct into the order register (manual operation), or it can be 'read' by a suitable order that treats the handkeys as a special register in the computing store.

29. A built-in monitor is provided, with two oscilloscope displays. One of these, the programmer's display, will show the content of either the order register or the order-number register: the other, the engineer's display, will show information derived from many more places in the machine. The timebases of these oscilloscopes run from right to left so that the digits in numbers and orders are shown as they would

conventionally be written. The time-base-triggering arrangement enables the condition of the machine to be studied in any pre-selected word-time.

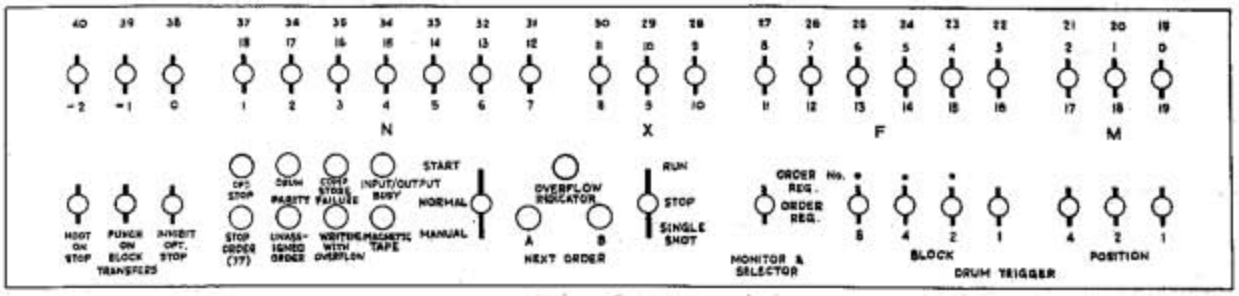
30. The computing equipment is contained in a three-bay cabinet, on the end of which is fixed the control desk. The computing-store registers and the arithmetical and control circuits are built in the form of standard 'packages', of which there are a relatively small number of types. The packages are supported in racks in the front part of the bays, and plug into sockets fitted to laterally mounted plates. The interconnections between packages in the same bay, and some of the connections between components of the same package, are made by direct wiring between socket tags in the back portion of the cabinet. Most of the interconnections between bays are made by plugs, identical with the plugs on the packages, which fit into sockets at the ends of the package rows.

31. Each bay can accommodate up to ten rows of packages, and there can be up to twenty packages in a row. Two coloured plastic buttons are let into the end of each package to indicate its type, and into the racks above each package position to indicate the correct dispositions of the packages. The colour code used is the normal 'resistor' colour code; and corresponds to the package type number. There are ten rows of packages in each of bays 2 and 3 (centre bay and the furthest bay from the desk). Bay 1 contains only four rows of packages, the rest of the space being occupied by the drum and monitor unit.

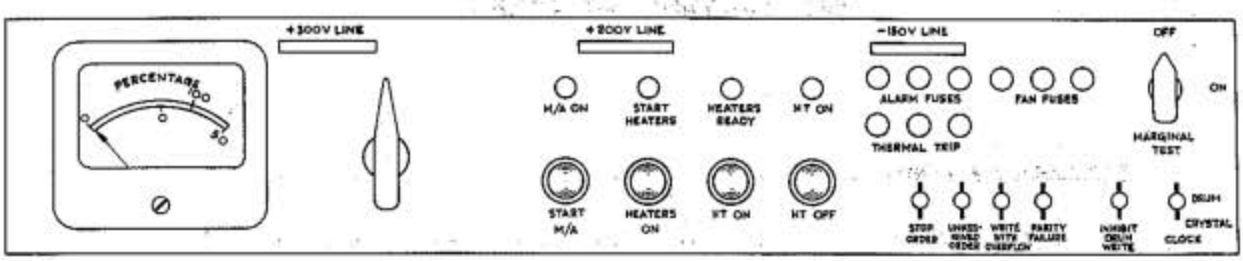
32. The power-supply equipment for Pegasus consists of two main units; a motor-alternator set and a two-bay power-supply cabinet, which may be placed some distance away from the computing cabinet. The motor-alternator set provides a regulated three-phase 415V supply for general requirements and a controlled three-phase supply, nominally at 200V, for the drum motor. The power-supply cabinet contains rectifiers and stabilisers giving three HT supplies (+300V, +200V and -150V) and four bias supplies (+13V, -4V, -10V and -20V) for the packages. The heaters of the valves in the packages are fed from transformers housed in the base portion of the computing cabinet. Unregulated AC is also required for the cooling fans (six in the computing cabinet and one in the power-supply cabinet) and for some of the equipment on the control desk.

CHAPTER II

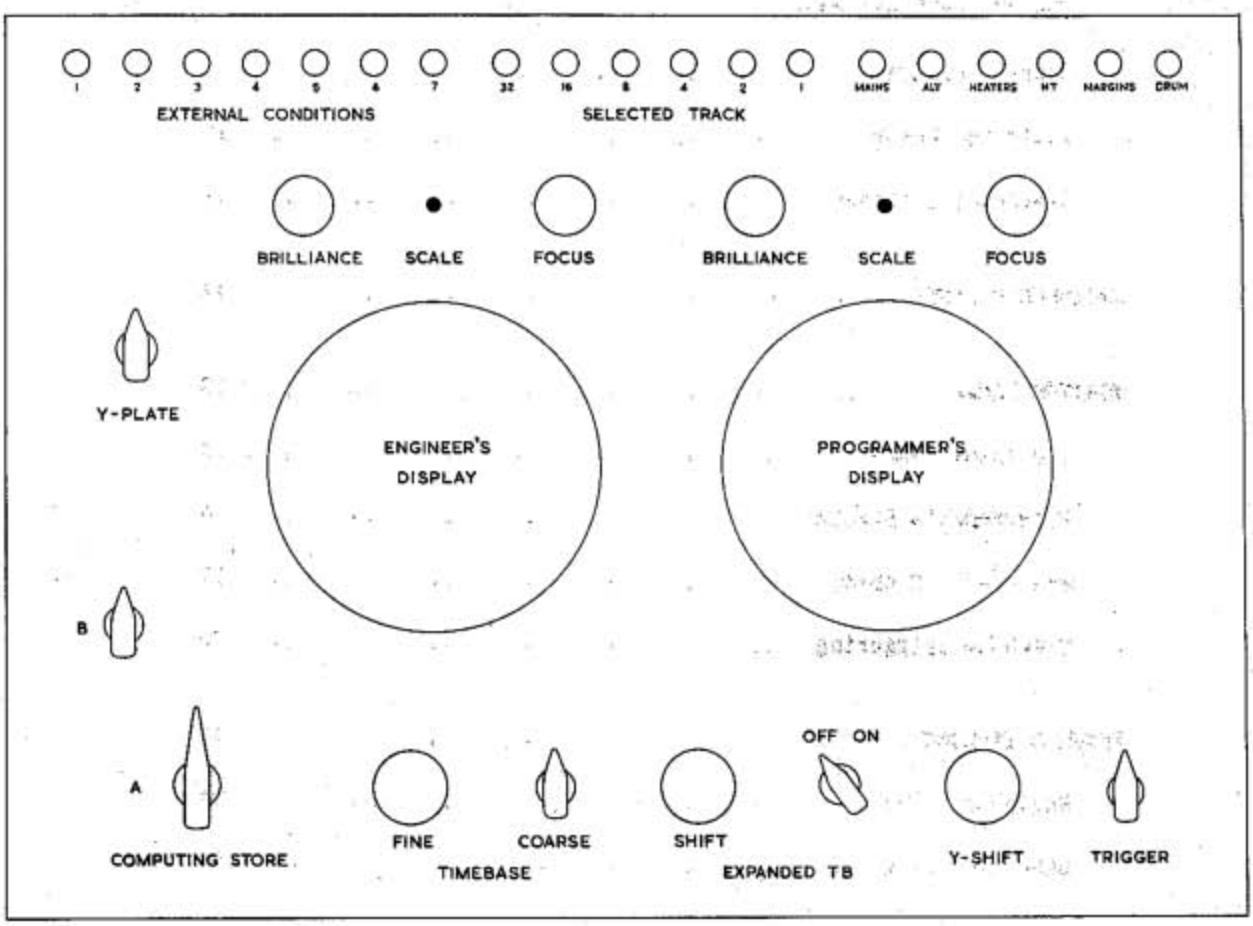
THE CONTROL SWITCHES



PROGRAMMER'S PANEL.



ENGINEER'S PANEL.



MONITOR PANEL.

ISSUED 1961

CHAPTER II

THE CONTROL SWITCHES

PROGRAMMER'S PANEL

1. The normal operation of the machine is controlled from the programmer's panel, which is mounted on top of the control desk. The figure opposite shows the layout. The two prominent switches in the centre of the panel are referred to as the 'start' switch and the stop' switch. During computation, the 'start' switch will be at NORMAL (centre position) and the 'stop' switch at RUN ('up' position).

'Start' Switch

2. Moving the 'start' switch to START sets in the machine certain conditions that should obtain at the beginning of a computation. It also feeds into the order register a permanently wired-in 'start' order pair. The 'stop' switch must be at STOP when the 'start' order pair is being fed in. Moving the 'start' switch to NORMAL and the 'stop' switch to RUN then enables the machine to obey the 'start' order pair, which transfers the first block of Initial Orders from the main store into the computing store and 'jumps' control to the first order in the block.

3. When the 'start' switch is at MANUAL, an order is fed into the order register corresponding to the setting of the row of handkeys at the top of the panel. This order can be obeyed once only by feeding it in with the 'stop' switch at STOP and then moving the 'start' switch from MANUAL back to NORMAL and the 'stop' switch to RUN or SINGLE SHOT; alternatively it can be obeyed repetitively, for testing purposes, by leaving the 'stop' switch at RUN and the 'start' switch at MANUAL.

'Stop' Switch

4. When the 'stop' switch is at STOP, the beat counter is inhibited; this means that the order or order pair to be obeyed next is kept circulating in the order register, but can exercise no control on the machine. The position marked SINGLE SHOT is spring-loaded. When the switch is depressed and released, the machine obeys a single order and stops in a condition in which it is ready to obey the next order (taking the next order pair into the order register if necessary). Moving the 'stop' switch to STOP and then back to RUN or SINGLE SHOT will clear an optional stop or a 77 stop.

The 'Manual' Handkeys

5. The top row of 22 handkeys on the panel enables a complete 19-digit order to be set up (a key in the 'down' position representing a binary 'one') with three extra digits to give the effect of extension by modification. The handkey information is fed

into the order register when the 'start' switch is operated as explained in paragraph 3; alternatively it can be read in (less the most significant two extension digits) by a suitable order specifying address 15, so that it can be stored and used later on. The keys are arranged in groups corresponding to the conventional notation of orders, and should be set up to give the binary equivalents of the octal or decimal numbers in orders. Notice, however, in this connection that the first N-digit can be an ordinary-register marker; it must be made a 'one' for ordinary-register addresses, the other six N-digits giving the binary equivalents of the octal block and position numbers.

6. A manual order is normally fed into the order register so that it is obeyed during beat A; but it must be presumed to be followed by a 'null' order that will be obeyed during B: thus two 'single-shot' operations are generally necessary when a manual order is obeyed. The 'null' B-order contains for testing purposes a single 'one' in the ordinary-register-marker position, and therefore causes a transfer from register 0.0 to accumulator 0. A manual order can be obeyed during B if it is fed in with the machine stopped in A; only one single-shot operation will then be necessary. Although three keys are provided for setting up a modifier address, there is no modification in the order adder when the order is obeyed manually only once. If the order is obeyed repetitively from the handkeys (see paragraph 4), it will be modified in the usual way except the first time it is obeyed: orders taken in by address 15 are modified in the normal way when they are obeyed. Notice also that the least significant 'extension' handkey sets the optional-stop condition of an order read in by address 15; when this key is down, the order is obeyed as a 'go' order.

7. A half-length number (more significant half) can be set up on the handkeys and read in by an order specifying address 15. The least significant 'extension' handkey sets the sign digit ('up' for a positive number). The manual handkeys are used in connection with Initial Orders for setting up warning characters (reference should be made to the programming manual for an explanation of this facility) and also as a means of controlling test programmes (which are explained in Volume 4).

8. The three handkeys at the left on the lower row, when in the 'down' position, have the following effects:

(i) To make the hooter below the desk sound when one of the internal stops (except the 'busy stop') occur;

(ii) to make the machine punch out on tape characters representing the relevant main-store block address during block-transfer orders;

(iii) to inhibit the 'optional' stop.

Unless it is inhibited, the optional stop occurs whenever an order pair containing a 'nought' (before modification) in digit place 0 is fed into the order register from the computing store. When a 'full stop' character is punched on tape after an order, the order pair of which it forms a part becomes an optional-stop order pair. One of the main functions of optional-stop order pairs is to stop the machine at a pre-selected point in a programme so that additional information can be fed in manually or on an auxiliary tape.

9. To the right of the 'stop' switch is a handkey with which the operator can select the content of the order register or that of the order-number register for display on the right-hand (programmer's) oscilloscope. The other seven handkeys are used in connection with monitor triggering, and will be dealt with when the monitor panel is described.

Indicator Lamps

10. The eight lamps to the left of the 'start' switch are 'stop' indicators, showing when one of the internal stops operates. The two 'busy' indicators will be flashing continually during input or output routines, and will give a steady indication only if one of the devices 'sticks'. The three indicators between the two control switches, too, will normally be flashing during a programme, and will give a steady indication only during a stop. The top lamp of these three indicates that overflow has occurred and that it has not yet been corrected by 'justification' or by a sub-routine introduced by an 'overflow-test' jump order. The other two lamps indicate after a stop whether the A-order or the B-order of a pair is the next to be obeyed.

Emergency Switches

11. At the ends of the programmer's panel assembly are mounted two emergency push buttons. These can be used to cut off the regulated AC supply to the computing cabinet and the power-supply cabinet, switching off the whole of the computing equipment and the fans.

ENGINEER'S PANEL

12. The engineer's panel is recessed below a flap in the centre portion of the desk. It carries the main power-supply switches and the controls required for marginal testing. It also carries a switch that enables either the drum clock track or the crystal oscillator to be used for time standardisation in the machine, a key for inhibiting writing on the drum, and five keys for inhibiting various types of stop. (The key for inhibiting the optional stop is on the programmer's panel). All the 'inhibit' keys on this panel, and the 'marginal test' switch, should be OFF, and the 'clock' switch should be set to DRUM, for normal computation.

MONITOR PANEL

Indicator Lamps

13. There is a row of nineteen indicator lamps at the top of the monitor panel. The first (left-hand) seven of these indicate which of the input and output devices are in use, according to a convention depending on the particular installation. In the standard installation, only lamp 1 is used; this indicates that the second tape reader is in circuit. When none of these lamps is on, the computer is operating with the first tape reader and the standard output punch.

14. The six lamps marked 'selected track' show which track on the drum was last selected by a drum-transfer order. The lamps are arranged to give an indication in binary notation, the track number being obtained in decimal form by adding together the numbers (power of two) engraved under those lamps that are on. There are sixteen blocks to a track, and the first track is numbered '0'; consequently track n will contain blocks $16n$ to $(16n + 15)$ inclusive, and words $128n$ to $(128n + 127)$ inclusive.

15. The last six indicator lamps show the condition of the power-supply switching circuits. These lamps should all be on for normal computation.

Programmer's Display

16. The right-hand oscilloscope is the programmer's display; it will show the content of the order register or the order-number register, according to the setting of the selector key on the programmer's panel. The order pair during A is shown on a double trace, the A-order (modified) appearing above the B-order (unmodified). The single (modified) order during B appears on the top trace; though it is duplicated (unmodified) on the lower trace during BD, and is replaced by the order number in BE. The order number appears only on the top trace. The display shows binary 'ones' as vertical lines, and binary 'noughts' as spots. Setting the 'scale' switch above the tube to ON spreads out the displayed digits into groups corresponding to the digit groups of a modified order. The timebase runs from right to left, and is triggered in a way that the operator can determine (see paragraph 20).

Engineer's Display

17. The left-hand oscilloscope, the engineer's display, has many more facilities associated with it. It may be made use of by the programmer, and those positions that are expected to be of use in programme control are engraved in white, the remainder of the engraving being in red. Only the white-engraved switch positions will be described here. For fuller details, the reader is referred to Chapter XVII of Volume 2 and Chapter VI of Volume 3.

18. The information to be displayed is selected by the 'Y-plate' switch on the left of the panel, which, in the three COMPUTING-STORE positions, works in conjunction with the two 'computing store' switches whose knobs are mounted below it. With the 'Y-plate' switch in the position marked NORMAL, and the coarse timebase control below the tube at HALF WORD, the form of the display is similar to that on the programmer's tube, and the 'scale' switch above the tube can be used to split up the digits into groups.

19. The two 'computing store' switches enable the content of any store line to be displayed, and several other locations in the computer to be investigated. The ordinary registers are selected by setting the block number on the upper switch and the position number under N-UNITS on the lower switch. The accumulators (positions under X on the lower switch) and the other white-engraved locations can be selected only when the upper switch is in one of the N-BLOCK positions 0 to 5.

Timebase Triggering

20. The two timebases are normally triggered together. A stationary display is possible only if conditions in the part of the machine to be studied recur at regular intervals. This will be true during stops, if the machine is obeying a loop, or if it is obeying a manual order repetitively. The 'trigger' switch at the right-hand side of the monitor panel can be used in conjunction with the seven 'drum trigger' handkeys on the programmer's panel to select a timebase trigger that will occur.

(i) repetitively (every two word-times),
or *(ii)* in synchronism with the drum rotation,
or *(iii)* in any preselected word-time in a bar.

21. When the 'trigger' switch is at DRUM ADDRESS, (the position marked '0' will normally be used) the timebase is triggered whenever the number picked up from the drum address track coincides with the setting of the seven 'drum trigger' handkeys. To make it possible for the timebase to be triggered several times per drum revolution (giving a brighter display) the effect of any of the first three handkeys can be inhibited by putting it in the 'up' position. Thus a key setting of

would cause a trigger coincident with drum addresses

BLOCK	POSITION	DECIMAL NUMBER	OCTAL NUMBER
0 0 0 1	0 1 1	11	1.3
0 1 0 1	0 1 1	43	5.3
1 0 0 1	0 1 1	75	9.3
1 1 0 1	0 1 1	107	13.3

Notice that the trigger occurs *after* coincidence, and that information will be displayed in the *next* word time.

22. The timebase can be triggered at the beginning of the first or last word-time of any order, or at the beginning of CD, by setting up the appropriate order number on the trigger handkeys and turning the trigger switch to CD, AD, AE, BD or BE as appropriate. The C-beat chosen is the one during which the order pair with the specified number is routed into the order register. With the switch in the position marked MULTI-BEAT, the timebase sweep occurs in word-time (AD + n) of a bar, where n is a binary number set up on the triggering handkeys. This facility will normally be used when a manual order is run repetitively.

POWER SWITCHING

23. The power supplies for the computer can be switched on from the control desk provided that the isolator on the power-supply cabinet and the main isolator on the star-delta starter (associated with the motor-alternator set) are both closed. An indication that this is so is given by the 'mains' lamp at the top of the monitor panel. The remaining switching operations can be carried out from the engineer's panel on the desk, the layout of which is shown in the figure at the beginning of this chapter.

Engineer's Panel

24. The switching procedure from the engineer's panel is as follows:

(i) Press the 'start M/A' button on the panel. After about ten seconds, the alternator windings will change from star connection to delta connection, and the motor-alternator indicator lamps on the engineer's panel and the monitor panel, and the 'start heaters' lamp on the engineer's panel, will come on.

(ii) Press the 'heaters on' button on the engineer's panel; this starts the 'run-up' of heater voltage and switches on the cooling fans. After about one minute a contactor (CR4) in the power-supply cabinet will close, showing that the heaters are on full power. There is then a further wait of about 1½ minutes due to a thermal delay unit in the power-supply cabinet. At the end of this time another contactor (RL1) will close, and the 'heaters ready' lamps on the engineer's panel and the monitor panel will come on. The bias supply will now be on, and the drum will start up.

(iii) Press the 'HT on' button on the engineer's panel. This energises a further contactor (CR2) and, as metal rectifiers are used, brings the HT supplies on almost immediately.

25. The power supplies are switched off by pressing one of the two buttons on the ends of the programmer's-panel assembly. If the computer is to be idle for some time, the isolator on the power-supply cabinet should then be opened to switch off the motor-alternator.

Other Switches

26. The motor-alternator can be switched on from the desk or from the star-delta starter; it can be switched off from the starter or with the isolator in the power-supply cabinet. The 'heaters on', 'HT on' and 'HT off' buttons on the engineer's panel have their counterparts in the power-supply cabinet, and there is a 'heaters off' button in the power-supply cabinet that has the same effect as the buttons on the ends of the programmer's-panel assembly.

27. The HT supplies should always be switched off when a package is changed. To this end, an 'HT off' and an 'HT on' button are fitted in each bay of the computing cabinet: these have the same effects as the corresponding buttons on the engineer's

panel and in the power-supply cabinet. Notice however that the HT supplies to the drum-servo packages (25C, 27H and 27J) and the HT and EHT supplies for the monitor come from separate power units and can be switched off only by switching off the heater supplies.

28. There is an emergency button in the back of each bay that has the same effect as the buttons at the ends of the programmer's-panel assembly, *viz.* to switch off all power supplies to the computing equipment. A junction plug is fitted in the top left-hand corner of each bay to enable the bias wiring to that bay to be isolated for fault-finding; the removal of this plug also prevents HT from being switched on. Otherwise the bias supplies (together with the HT supplies) can be switched off with a rotary switch in the power-supply cabinet. The drum can be switched off by itself by means of a toggle switch mounted behind the rear baffle in bay 1.

CHAPTER III

PUNCHING TAPES

CHAPTER III

PUNCHING TAPES

1. The system whereby a programme in manuscript is translated into punched tape has been arranged.

(i) to make matters as simple as possible for the perforator operator;

(ii) to present to the machine unambiguous information that it can compile, by means of the routine called Input, into the form most convenient to itself, and to enable it to stop if errors appear on the tape or occur in the reading process;

(iii) to enable the programme to be printed by an interpreter unit from the tape in an easily recognisable form.

2. The teleprinter code relating numbers and symbols to tape characters has been arranged so that all symbols used in orders and in information to be processed mathematically correspond to tape characters with odd parity, i.e. with an odd number of holes in the row. The exception to this rule is the teleprinter instruction 'carriage return' (CR). This is 111.10, where a 1 here represents a hole on tape and the stop represents the sprocket hole. It must be followed immediately by a second teleprinter instruction, 'line feed' (LF), otherwise it will be treated as an error, and the machine will stop. The symbols with odd parity on tape are:

0 1 2 3 4 5 6 7 8 9 + - . LF Sp Er

Sp is the teleprinter instruction 'space'. Er is the 'erase' character: it corresponds to a row of five holes on tape, and can therefore be used to obliterate any character punched in error. It is ignored by Input unless it comes after CR, when CR is treated as an error and the machine stops. It is printed out by the interpreter as a symbol like this  .

3. The teleprinter, like a typewriter, operates in either of two states, 'figure shift' and 'letter shift', so that letters or figures can be printed out as interpretations of the same tape characters. There are tape characters corresponding to the teleprinter instructions 'figure shift' and 'letter shift' - these are referred to here by the symbols φ and λ respectively. φ corresponds to blank tape (no holes); thus a blank leader on a tape fed into the interpreter unit will ensure that the teleprinter starts off in the 'figure shift' condition. λ corresponds to 110.11 on tape; consequently the two 'shift-instruction' characters have even parity. The character α, unless it comes within an order or number, causes the machine to enter a special section

of Initial Orders, called 'α-search', which will accept and ignore blank tape. The only way of leaving α-search and entering Input proper is to punch LF or CRLF, which must therefore be at the head of any tape. λ causes the machine to enter another section of Initial Orders called 'γ-search', which enables the machine to read, and act on, warning characters. The characters λ or φ coming within orders or numbers are treated as errors, and the machine stops.

NUMBERS

4. The machine 'recognises' the start of a number by the appearance of a character representing a sign (+ or -) on tape, and the end of a number by the terminating characters (CRLF or Sp). The sign and terminating character(s) must always be punched. Numbers may be decimal integers in the range.

$$-2^{38} \leqslant n < 2^{38}$$

or decimal fractions in the range

$$-1 \quad \leqslant x < 1$$

Mixed numbers (part integral, part fractional) are not permissible. The 'full stop' character is used as a point. Punching a zero between the sign and the point is optional. The integer -2^{38} and the 'fraction' -1 take the same form in the machine; they may be punched.

	-1.0 Sp	or	-1.0 CRLF
Zero is punched	+0CRLF	or	+0Sp

ORDERS

5. The machine 'recognises' the start of an order by the appearance of a numeral character *not* preceded by a sign, and the end of an order by the appearance of the terminating characters CRLF (*never* Sp). The punching rules for orders are as follows -

(i) N is punched in any of the ways in which it is written, *viz:*

decimal integer	e.g. 15
ordinary-register address	e.g. 1.5
jump address to B-order	e.g. 1.5+
relative block address	e.g. 15+

(ii) At least one Sp is punched after N; but this may be omitted if, and only if, N ends in +.

(iii) X and F are punched as they are written; but, if N is a main-store single-word address, a minus must be punched in place of X.

(iv) M is punched as written; but, except in optional-stop orders, it may be omitted, or replaced by Sp, if it is zero.

(v) Optional-stop order pairs are indicated by punching a full stop after M in either the A-order or the B-order, but not both. If M is zero in an optional-stop order, it must be punched as such or as Sp.

(vi) The order is terminated in CRLF. LF or CRLF may be punched at the beginning of an order to split the print-out into blocks.

(vii) As many Sp as desired may be punched before N, between N and X or after M; but Sp must not be punched between X and F or between F and M.

(viii) Er may appear anywhere except between CR and LF.

6. Orders are collated in pairs. If, therefore, a word is required to contain one order only, the other must be punched as a null order, i.e. as

OCRLF

Pseudo Orders

7. A modifier and/or a counter can be placed in store by punching a pseudo order, which will never be obeyed. Modifiers occupy digit places 1 to 13 in a word, i.e. those places normally occupied by the N-digits, the X-digits and the first three (binary) F-digits of the A-order. Pseudo-order pairs should always be punched as 'stop' order pairs to suppress the 'go' digit. The way in which a modifier is punched as a pseudo order depends on the way in which it is written.

(i) A block-and position address, e.g. 74.3 is punched with the block-number as N in an A-order and the position number as the first octal digit of F. X is punched as a minus, the second digit of F as 0 and M as 0 followed by a full stop, thus

74Sp-300.CRLF.

(ii) A relative-block address, e.g. 4+.3, us punched in the same way, except that Sp is replaced by +, thus

4+-300.CRLF

(iii) A decimal main store address, e.g. 2450, is punched simply as N. The first octal digit of F is then replaced by a minus, thus

2540Sp -- 0.CRLF

8. A counter is placed at the end of a word. It can therefore be punched as a pseudo B-order. It is simply punched as written, and terminated by CRLF. If the word is to contain a counter only, the A-order is punched as a null order (see paragraph 6).

WARNING CHARACTERS

9. Warning characters are characters in 'letter shift' that can be punched on the tape to make the machine enter some special subroutine. Warning characters must, of course, be preceded by the 'letter shift' character, λ, and must be followed by 'figure shift', φ. This will be followed by Sp or CRLF, according to the requirements of the print-out. Certain warning characters are followed by a main-store address or by two main-store addresses; CRLF is then punched after the address information. The punching sequences recommended for warning characters are detailed in Volume IV.

CHAPTER IV

THE BOOTSTRAP, OCTAL INPUT AND MANUAL WRITING

BOOTSTRAP MANUAL OPERATIONS

1. Insert tape on 'zero'
2. Opt. stop key normal
3. NORMAL to START to NORMAL
4. Handkeys 0.0 0 60
5. NORMAL to MANUAL
6. Handkeys 34 2 02
7. S/S (Single shot)
8. Handkeys ⊙ 15 1 00
9. S/S S/S S/S
10. Handkeys 0.0 1 10 2
11. S/S S/S
12. MANUAL to NORMAL
13. S/S S/S
14. Handkeys 0.1 1 10
15. NORMAL to MANUAL
16. Handkeys 0.0 2 66
17. S/S S/S
18. MANUAL to NORMAL
19. RUN RUN
20. Handkeys 5 2 50
21. RUN
22. Handkeys 16 2 01
23. RUN
24. Handkeys 0.3 2 64
25. RUN
26. Handkeys 0.6 2 10 2
27. RUN RUN

Reads tape

BOOTSTRAP TAPE

Leader of zeros

8	4	1	0	8	.	.	0								
=	5	0	0	8	≠	2	0								
9	1	5	.	8	≠	2	0								
.	LF	1	4	8	1	2	0								
8	×	→	4	0	υ	×	0								
8	6	(	0	0	8	n	×								
8	4	(	.	2	1	0	8								
	2	6	φ	8	>	2	0								

binary punching

CR	0	5	LF	=	2	0	2	0	2	0	0	0			
4	3	0	5	0	6	1	0	0	2	6	3	0			

unconventional octal

2	0	4	2	6	1	0	0	0	0	0	7	7	0	CR	LF
3	2	0	0	6	0	0	1	1	2	1	0	0	0	CR	LF
3	0	3	1	1	0	0	1	0	0	4	4	0	0	CR	LF
2	0	6	4	5	0	0	0	0	4	0	6	0	0	CR	LF
3	0	1	0	6	4	0	1	2	0	2	1	0	0	CR	LF
0	1	7	1	0	0	0	0	0	1	1	5	2	0	CR	LF
2	2	4	1	5	3	0	0	0	2	1	0	1	0	CR	LF
2	0	0	0	0	0	0	0	0	0	0	0	0	0	CR	LF
0	1	7	1	0	0	0	1	1	5	1	1	0	0	CR	LF
0	1	7	2	0	0	0	1	1	3	0	6	0	0	CR	LF

conventional octal

CHAPTER IV

THE BOOTSTRAP, OCTAL INPUT AND MANUAL WRITING

This chapter is included primarily as an exercise in following the detailed operation of the machine. An explanation of the order code will be found in Chapter IV of Volume 2.

1. The bootstrap is a programme designed to put into the computing store enough orders to enable the machine to read simple tapes when for any reason the permanent record on the drum is not available. The routine read in by this method occupies two blocks of registers and will accept order pairs punched octally, ignoring all characters with checked (parity-adjusted) values greater that 12 and stopping (77 stop) on reading the character of checked value 12, *viz.* full stop. Subsequent programmes read in by this routine are stored sequentially starting with register 2.0.

2. The bootstrap consists partly of a series of manual operations and partly of a series of orders punched on tape. The process can be outlined as follows:

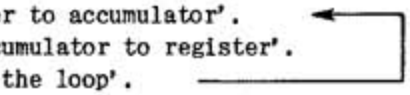
(i) A three-order loop is put into the computing store from the handkeys by 'single-shot' operations.
(ii) This simple loop is used to enable the machine to read still from handkeys, but more rapidly by means of 'run' operations, a simple input routine for binary-punched tape.
(iii) The binary-input routine is used to compile a simple octal-input routine punched in binary form on tape.
(iv) The remainder of the octal-input programme, which is punched octally, is then read in from the same tape.

HANDKEY OPERATIONS

Primary Loop

3. To enable handkey orders to be read in by 'run' operations, a three-order loop is necessary. The essential orders are,

(i) 'Read handkey order to accumulator'.
(ii) 'Transfer from accumulator to register'.
(iii) 'Jump to complete the loop'.



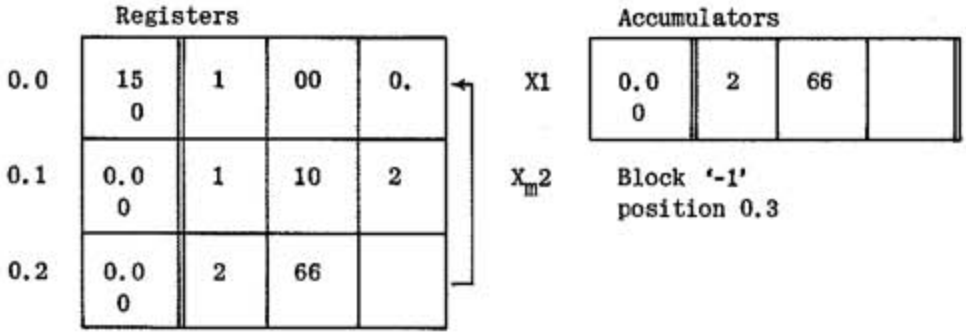
The orders read in this way must be stored serially, which is most conveniently done by making *(ii)* a modified order and *(iii)* a 'unit modify' order. *(i)* must contain an optional stop to enable the handkey setting to be changed each time the loop is cycled.

4. Assembling these three orders takes up operations 1 to 19 of the bootstrap as follows:-

1.	Insert tape on 'zero'	Tape will be read in on 'checked' input so that blank tape would appear as the number 31; thus it must be preceded by a 'leader' of zeros.
2.	'Inhibit opt. stop' key up	
3.	Switch NORMAL → START → NORMAL	This clears the overflow staticiser and any internal stops that may be 'up'.
4.	Set on handkeys 0.0 0 60	This is identical with the B-order of the 'start' pair, and is inserted in case there is a fault in its generation.
5.	Switch NORMAL → MANUAL	Handkey order (4) enters order register.
6.	Set on handkeys 34 2 02	(All orders are 'go' orders unless otherwise indicated).
7.	SINGLE SHOT	Machine obeys order (4) clearing order-number register and returning to C-state. Order (6) taken into order register.
8.	Set on handkeys 0 15 1 00	First order of loop.
9.	SINGLE SHOT	Machine obeys order (6) putting -2^{-10} into X2. This is equivalent to a modifier referring to block '-1' and ensures that X2 will be empty when the loop has been cycled 8 times. An order clearing X2 would have been necessary in any case.
	SINGLE SHOT	Machine obeys null order (B-portion of order 6) and returns to C-state taking order (8) into order register.
	SINGLE SHOT	Machine obeys order (8) reading order (8), which is still on the handkeys, into X1. Beat counter now in A-state (next order 'B')
10.	Set on handkeys 0.0 1 10 2	Second order of loop.
11.	SINGLE SHOT	Machine obeys null order (B-portion of 8) and returns to C-state taking order (10) into order register.

	SINGLE SHOT	Machine obeys order (10) transferring order (8) from X1 into 0.0, (no modification for manual orders). Beat counter now in A-state (next order 'B')
12.	MANUAL → NORMAL	
13.	SINGLE SHOT	Machine obeys null part of (10) and returns to C-state. Order number is still 0.0 so order (8) now enters order register from N-bus.
	SINGLE SHOT	Machine obeys order (8) i.e. reads order (10) from handkeys into X1. Beat counter in A-state (next order 'F'). Order number is now 0.1.
14.	Set on handkeys 0.1 1 10	A temporary order needed to give the correct destination for order (10), as the modifier cannot yet be stepped on.
15.	NORMAL → MANUAL	This injects order (14) into the order register *in the B-position.*
16.	Set on handkeys 0.0 2 66	Third order of loop.
17.	SINGLE SHOT	Machine obeys order (14), i.e. transfers order (10) from X1 into 0.1. Moves on into C-state and takes order (16) into order register.
	SINGLE SHOT	Machine obeys order (16). Modifier in X2 becomes 0.1 (minus 8), which is non-zero (mod. 8); but switch is still at MANUAL so order 16 enters order register again with order number at 0.0 and beat counter at C.
18.	MANUAL → NORMAL	
19.	STOP → RUN	Machine obeys order (16) again, i.e. it increases modifier in X2 to 0.2 (minus 8) and jumps, transferring order (8) to order register from 0.0, which causes an optional stop.
	RUN → STOP → RUN	Clears optional stop and causes loop to be obeyed once. Order (16), still on handkeys, is transferred via X1 to 0.2 and obeyed. Modifier is increased to 0.3, and jump to 0.0 causes an optional stop.

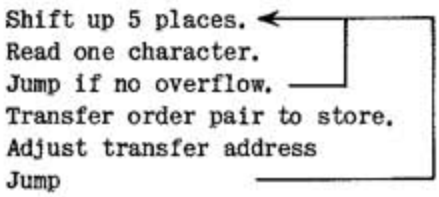
5. At the end of bootstrap operation 19, the contents of the main store are as follows:



The order number is 0.0, the order from 0.0 is in the order register, and the beat counter is held in the C-state by the optional stop.

Binary-Input Orders

6. The rest of the manual operations are concerned with feeding the first part of the binary-input routine into the machine. The process of compiling orders from binary-punched tape consists in reading the five-digit characters one at a time into an accumulator, the content of the accumulator being shifted up five places before every 'read' operation. As all the order pairs compiled in this way will be 'go' order pairs, the end of the process of compiling one order pair will be indicated by the appearance of a 'one' in the 'stop-go' position, i.e. by overflow after a shift. The process is therefore:



Notice that the 'read' order must come *after* the 'shift' order, otherwise the word would be transferred to store with its last five digits all zeros, and its last character would appear instead as the first character of the next word.

7. The final manual operations of the bootstrap are as follows:-

20.	Set handkeys 5 5 50	Shift order. Arithmetical so that overflow is detected.
21.	RUN → STOP → RUN	Machine obeys 3-order loop. Order (20) transferred to 0.3. Modifier in X2 becomes 0.4 (minus 8). Opt. Stop.

22.	Set handkeys 16 2 01	'Read tape character into X2'.
23.	RUN → STOP → RUN	3-order loop. Order (22) transferred to 0.4. X_m2 becomes 0.5 (-8). Opt. stop.
24.	Set handkeys 0.3 2 64	'Jump if no overflow or clear overflow'.
25.	RUN → STOP → RUN	3-order loop. Order (24) transferred to 0.5. X_m2 becomes 0.6 (-8). Opt. stop.
26.	Set handkeys 0.6 2 10 2	Transfer order for compiled order pair.
27.	RUN → STOP → RUN	3-order loop. Order (26) transferred to 0.6. X_m2 becomes 0.7 (-8) Opt. stop.
	RUN → STOP → RUN	3-order loop. Order (26) duplicated in 0.7. Content of X2 becomes zero after 'unit modify' so no jump. Control passes to order in 0.3 ('shift') and thence to order in 0.4, which causes tape-reader to operate.

8. At this stage the contents of the main store are:

Registers

0.0	15 0	1	00	0.
0.1	0.0 0	1	10	2
0.2	0.0 0	2	66	
0.3	5 0	2	50	
0.4	16 0	2	01	
0.5	0.3 0	2	64	
0.6	0.6 0	2	10	2
0.7	0.6 0	2	10	2

Accumulators

X1
same as 0.6 and 0.7

X2
zero

The three-order loop necessary for collating binary-punched orders is in 0.3, 0.4 and 0.5; but there is no transfer order except for the peculiar one in 0.6 and 0.7. The transfer orders to be used eventually are taken in from the binary tape.

BINARY TAPE

Remainder of Binary Input

9. The first binary order pair on tape is:

2 0.3	0 0	10 60	

This is compiled in X2 by cycling the orders in 0.3, 0.4 and 0.5, after which control passes to the order in 0.6, the N-address of which is modified by the 11th, 12th and 13th digits of the order pair in X2, i.e. by the first three F-digits of the A-order. The order in 0.6 is therefore obeyed as:

0.7	2	10	

so that the new order pair is written into 0.7 and is obeyed next. It causes X2 to be cleared and transfers control back to 0.3, the start of the compilation loop.

10. The second binary order pair is

34 0.6	4 2	00 10	

which is compiled in X2 as before, and used to modify the order in 0.6. As the first three F-digits in the new A-order in X2 are all zero, the putative order in 0.6 is unaffected by modification. The new order pair is therefore transferred to 0.6, the null B-order originally in 0.6 - and now in the order register - is 'obeyed', and control passes to the order pair in 0.7, which clears X2 and transfers control to 0.3.

11. The third binary order pair on tape is:

16 0.6	4 2	53 10	

When this has been compiled in X2, control passes to the order pair just transferred to 0.6 (see paragraph 10), which first sets the number 2^{-10} in X4 and then erases

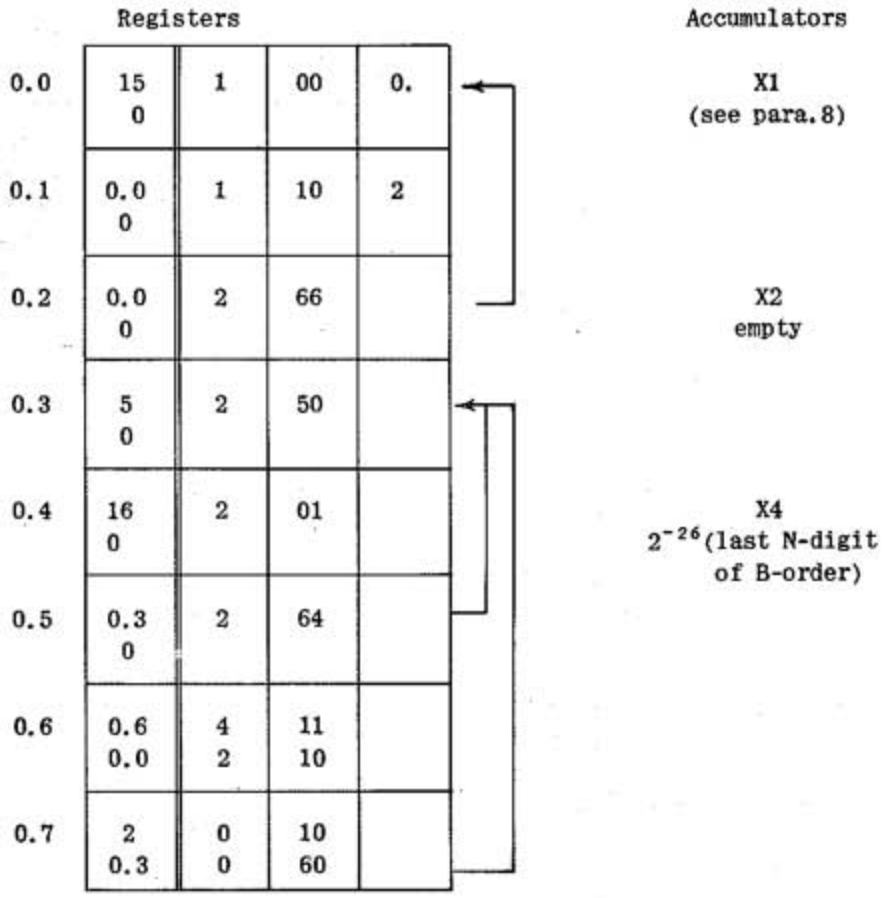
itself in 0.6 by transferring the new order pair from X2 to that address. The order pair in 0.7 (paragraph 9) then clears X2 and transfers control back to 0.3. The number in X4 is to be used eventually for stepping on the transfer address.

12. The fourth binary order pair, which completes the binary-input routine is:

0.6 0.0	4 2	11 10	

When this has been compiled in X2, the order pair last transferred to 0.6 (paragraph 11) is obeyed, which shifts the single digit in X4 down 16 places and then erases itself in 0.6 by transferring the new order pair from X2 to that address. X4 now contains the number 2^{-26}, i.e. a single digit in the least significant N-position of the B-order. The order in 0.7 then clears X2 and transfers control to 0.3.

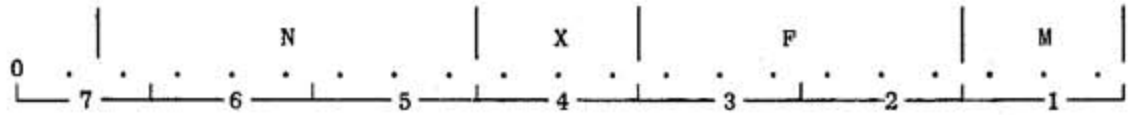
13. The contents of the computing store at this stage are:



The handkey-input loop still remains in 0.0 to 0.2; but 0.3 to 0.7 contain a seven-order routine for reading binary tape and transferring compiled orders to successive registers. Notice that the first transfer will be to 0.0: although the first order in 0.6 alters the N-address of the B-order *in store*, the B-order in the order register will still specify the *original* N-address.

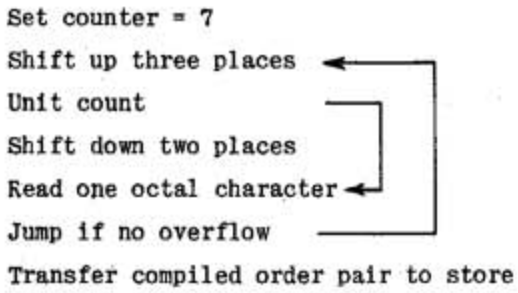
Octal-Input Orders

14. The basic routine for compiling *octal*-punched orders is similar to that used for binary-punched orders, in that it consists in successive 'shift' and 'read' orders, with an 'overflow' jump order to indicate when the compilation of a 'go' order pair is complete, followed by a transfer order putting the compiled order pair into an ordinary register. The octal digits, each of which corresponds to three binary digits, are allocated in a single order in the following way:

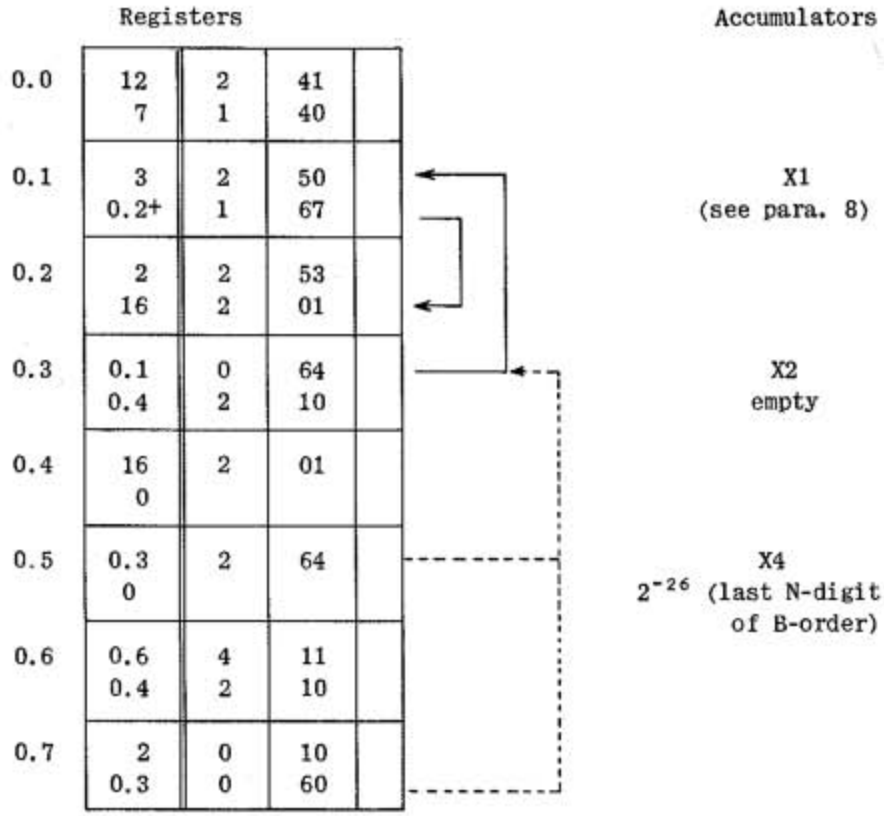


There are therefore seven octal digits to an order, the most significant of which contains the ordinary-register marker and (in an A-order only) the stop-go digit.

15. The A-order is read in first. To ensure that the B-order is placed immediately after the A-order, the most significant two binary digits in the first octal character of the B-order (which will both be zeros) must be made to 'overlap' the least significant two digits of the A-order. This is done by shifting the A-order *down* two places before the first octal digit of the B-order is read in. As the A-order will at this stage be confined to the less significant part of the word, a counter must be used to indicate when it is complete. The basic octal-input routine for one order pair is therefore



16. The binary input routine in 0.3 to 0.7 (see paragraph 13) is used to read four more binary-punched order pairs into registers 0.0 to 0.3; these will form the basis of the octal-input routine. The order pair placed in 0.3 replaces the first order in the binary-input routine, and, as the B-order in 0.7 prescribes a jump to 0.3, provides a convenient entry point for the octal-input routine. At this stage, the contents of the computing store are:

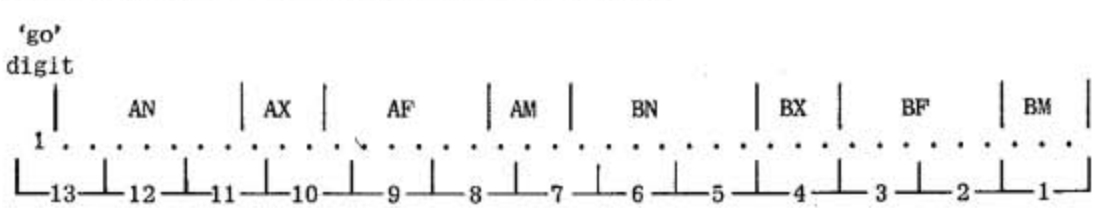


Notice that the N-address in 0.6+ is now 0.4 - after four additions from X4 as prescribed by the A-order in 0.6.

OCTAL TAPE

Special Punching

17. The portion of octal input so far assembled is deficient in that, although there is in 0.0+ an order for setting a counter of value 7, this order is as yet inaccessible. The only 'process completed' indication is therefore the 'overflow' jump order in 0.3; consequently the first piece of octal-punched tape must be compiled not in separate orders but in complete order pairs. The B-orders in these pairs can be punched in the normal (octal) manner; but the three-digit groups in the A-orders will be 'out of step' with the functional divisions in the orders. An order pair punched octally for assembling as a whole must be divided up as follows:



18. Two order pairs are punched in this special way. The first is:

0.3 16	4 2	11 00	

This is placed in 0.4 by the order in 0.3+ (see paragraph 16) and is obeyed next. It causes the N-address in 0.3+ to be increased by unity (by addition from X4) and reads the next tape character into X2, after first clearing X2 (function 00). Control then passes to the order in 0.5, which transfers control to the order in 0.3, which in turn transfers control to the shift order in 0.1, the overflow staticiser having been cleared when the first order in 0.3 was last obeyed.

19. The next order pair read in is:

12 0.0	2 2	43 63	

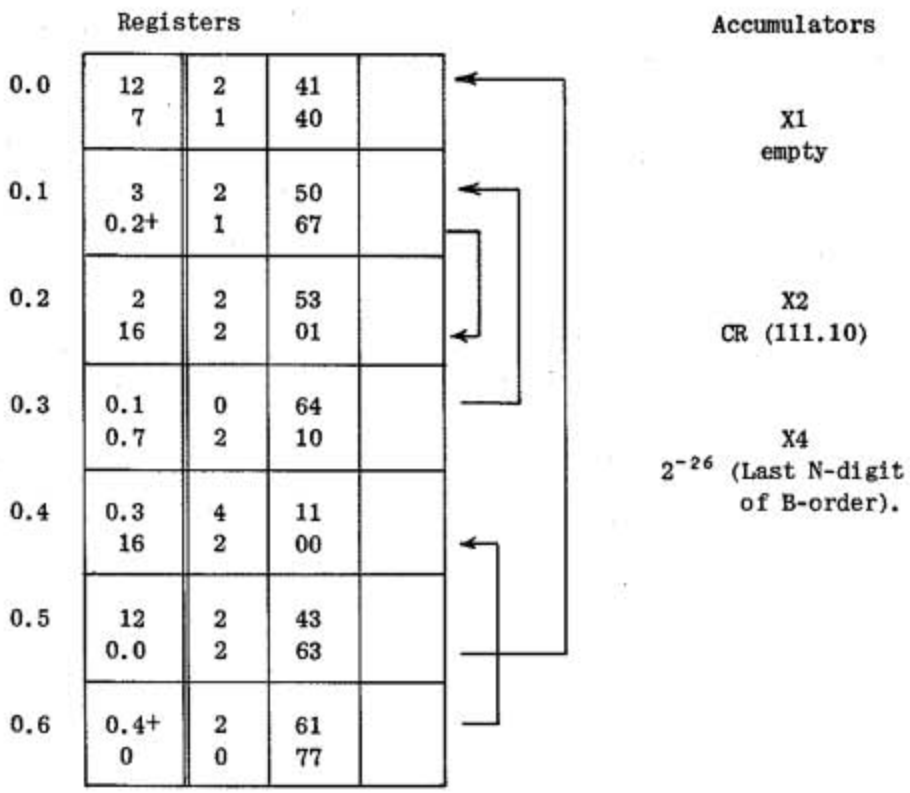
This is transferred to 0.5, which is the N-address in 0.3+ at this stage. The machine then obeys the order pair in 0.4 (paragraph 18), which increases the transfer address to 0.6 and reads the next character into X2. The new order pair in 0.5 contains a jump order transferring control to 0.0, and hence giving access to the counter order. Subsequent order pairs on tape can therefore be punched octally in the conventional way. The subtraction of 12 by the A-order in 0.5, and its replacement by the A-order in 0.0 give a means whereby the routine can detect the full stop at the end of the tape and can ignore teleprinter instructions - CR, LF and Sp - and 'erase' and blank tape. The way in which this is done will become apparent when the next order pair is considered.

REMAINDER OF OCTAL INPUT

20. The first order pair punched according to the conventional octal system is:

0.4+ 0	2 0	61 77	

This is transferred to 0.6; the order pair in 0.4 (paragraph 18) causes the transfer address to be increased to 0.7 and the next character, which is CR, to be read. The machine is now storing the full octal input routine.

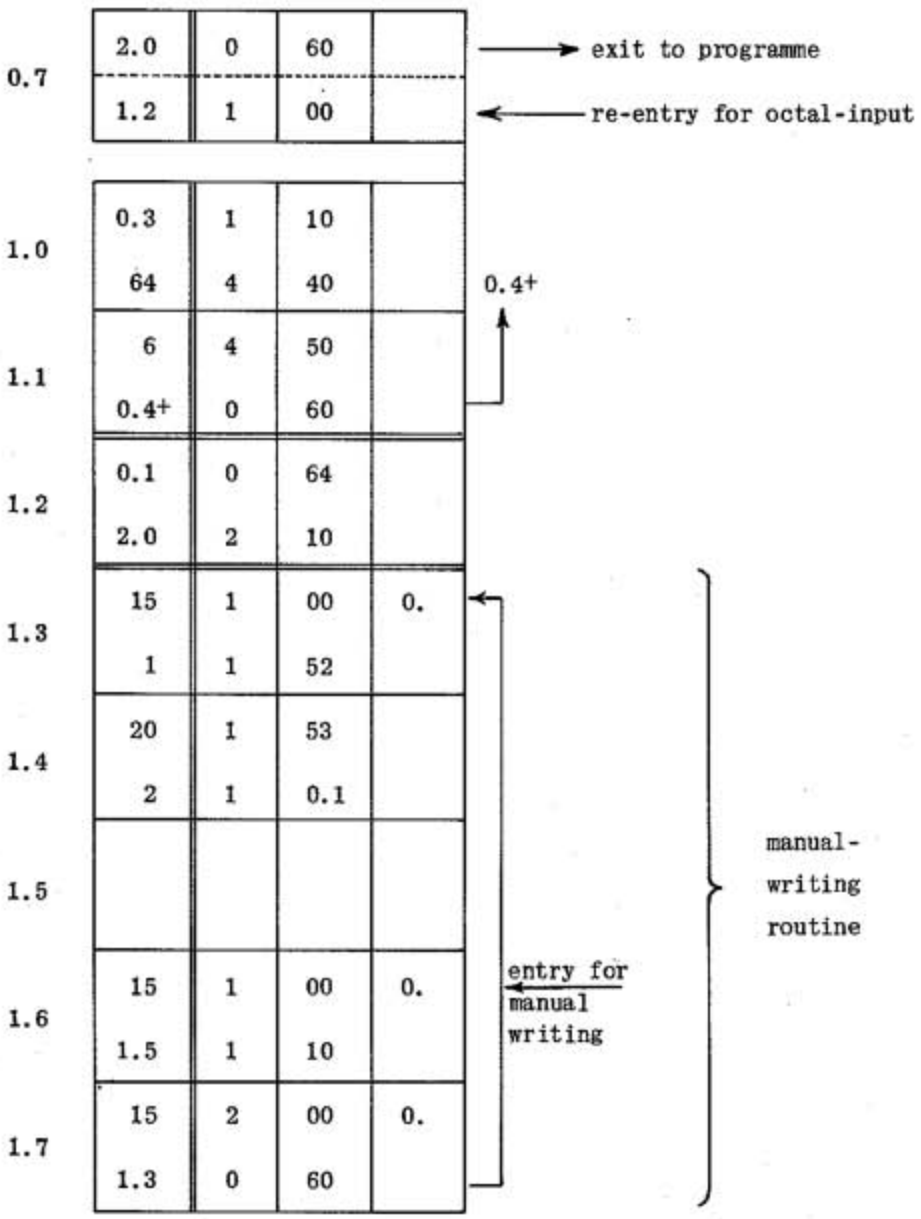


Order number 0.5

21- The CR character has a numerical value of 30. The first order in 0.5 reduces this to 18; so the order in 0.5+ ('jump if negative') is unsatisfied. The order in 0.6 ('jump if non-zero'), however, transfers control back to 0.4+, which clears X2 and reads in the next tape character. This is LF - numerical value 13 - so the same loop is cycled once again and the next character is transferred to X2. This is 3, the first octal character of the next order pair. Order 0.5 reduces it to -9, and the satisfied jump from 0.5+ transfers control to 0.0. The octal number 3 is reconstituted, a counter equal to 7 is set in X4, and the order pair is read into X2 by means of the loop in 0.1, 0.2 and 0.3. The order pair is then transferred to 0.7, the transfer address is increased to 1.0, and control enters the loop in 0.4+, 0.5 and 0.6 to ignore the terminating characters CR and LF.

22. The rest of the octal tape is read the same way, characters with checked (parity-adjusted) values greater than 12 being ignored, provided that they appear between complete order pairs. The tape is terminated in a full stop, which has a checked numerical value of 12. Control then passes to the 'stop' order in 0.6+, and the machine stops in the C-state after the order from 0.7 has been transferred to the order register. The full stop must either follow immediately after the last order pair or be separated from it only by characters with checked values greater than 12. Notice that the first order in 0.5 clears X2 after a full stop has been read in.

23. When the whole octal-input tape has been read in, registers 0.0 to 0.6 are still as shown in paragraph 20, except that the transfer address in 0.3+ has been increased to 2.0. The other contents of the computing store are as follows:



At this stage, X4 still contains 2^{-26}.

24. If it is now desired to read in another programme (punched octally, of course), a manual jump order transferring control to 0.7+ must first be inserted. The procedure

is as follows:

1.	Insert Tape on blank leader	
2.	Switch RUN → STOP	Clears 77 stop.
3.	Set handkeys 0.7+ 0 60	Unconditional jump.
4.	Switch NORMAL → MANUAL → NORMAL	Puts handkey order into order register.
5.	Switch STOP → RUN	Handkey order obeyed. Control passes to 0.7+ and machine reads tape.

25. The orders in 0.7+ and 1.0 transfer the order pair in 1.2 (which is never obeyed from this address) via X1 into 0.3. The only effect of this is to change the transfer address in 0.3+ to 2.0 so that the programme on tape will be read into block 2 of the computing store *(et seq.)*. The orders in 1.0+ and 1.1 set 2^{-26} in X4 - this is a general requirement - and the order in 1.1+ transfers control to 0.4+, which is the start of the input loop that will accept blank tape. The tape is then read as before; it must terminate, like the octal-input tape, in a full stop, so that, when the whole programme has been assembled, the machine will obey the '77' stop and take the order pair in 0.7 into the order register. The new programme is then entered by operating the 'stop' switch (RUN→STOP→RUN), which clears the '77' stop and causes the machine to obey the jump order in 0.7, transferring control to the first order of the new programme in 2.0

MANUAL WRITING

26. Registers 1.3 to 1.7 contain a short routine for writing full-length numbers or order pairs from the handkeys into the computing store. The three read orders in this routine embody optional stops to enable the handkey setting to be changed; the 'inhibit optional stop' switch must therefore be 'up'. The procedure starting from a '77' stop in C is as follows:

1.	Set handkeys 'go' 1.6 0 00	Jump order to entry of routine.
2.	Switch NORMAL → MANUAL → NORMAL	Jump order enters order register.
3.	Switch RUN → STOP → RUN	Stop cleared. Jump obeyed. Order pair in 1.6 enters order register. Opt. stop.

4.	Set of handkeys 'go' N 1 10	Where N is the address for transfer from handkeys.
5.	Switch RUN → STOP → RUN	Machine obeys order pair in 1.6, transferring order from handkeys via X1 to 1.5. Order pair in 1.7 enters order register. Opt. stop.
6.	Set handkeys as required	More significant half of number, or A-order (with *s-g* digit), to be written into store.
7.	Switch RUN → STOP → RUN	Handkey setting read into X2. Jump to 1.3. Opt. stop.
8.	Set handkeys as required	Less significant half of number, or B-order, to be written into store.
9.	Switch RUN → STOP → RUN	Orders in 1.3 to 1.5 obeyed. Handkey setting read into X1, shifted logically up one place (to clear any unintentional E-digits) and then down 20 places into B-position. More significant half is brought into X1 from X2 and whole word is transferred to address specified in operation (4). Control passes to 1.6. Opt. stop.

10. Continue as from operation 4 with new address. Exit by means of a manual jump order.

BOOTSTRAP TAPE

27. The characters on the bootstrap tape are given at the beginning of this chapter. As shown, they are arranged in order pairs; but the actual print-out of the tape will depend on the appearance of the characters CR and LF, and the operation of the automatic carriage return and line feed on the teleprinter. CR and LF are not printed of course. CR and LF have been arranged to appear in the first unconventional octal order pair to improve the recognisability of the print-out. In fact, CR places two extra 'ones' at the more significant end of the order pair; these are not transferred to store, of course, and their only effect is to set the overflow staticiser during the first, instead of the third, of the three left shifts.

GLOSSARY

AND

LIST OF SYMBOLS

GLOSSARY

A

Accumulator — Generally a special storage location with an associated arithmetical circuit enabling it to be used in accumulative processes. In Pegasus, one of the eight storage locations of which at least one is used in any arithmetical operation.

Address — Number designating a part of a computer store.

AND Operation — The selection of a class whose members possess two or more specified characterising features; especially the selection of those digit places in which a 'one' appears in two or more binary numbers.

Arabic Numerals — The figures 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Argument — The significant digits of a number taken in sequence, but without regard to their place values. Usually expressed with a standard placing.

Assembly — The process of compiling within the machine a complete programme made up from a master programme and a number of subroutines. A permanently stored programme designed to do this.

B

Bar — A sequence of beats; in Pegasus, the period during which an order pair is obeyed.

Beat — A period one or more word-times long. A component of the rhythm of a computer.

Binary Scale — Numeration scale with radix two.

'Bit' — A binary digit. Either 1 or 0.

C

Clock — A waveform controlling the timing of digit pulses.

Collation — The selection of part of a word by an AND operation with another word containing 'ones' in the appropriate places.

C *(cont.)*

Commutator	Circuit for generating p-pulses *(q.v.)*
Complementation	The replacement of a number by its deficiency from a numerically greater quantity.

D

Decimal	Proceeding by tens.
Decimal Scale	A scale of counting using ten as radix.
Decoding	The generation of gating waveforms in accordance with the digits in an order.
Digital Computer	Computer in which numbers are represented digit by digit by different states of the computing elements.
Double-Length Number	A number two word-lengths long.

E

Exponent	The power to which a number is raised. In floating-point working, the power of the radix by which the argument is assumed to be multiplied.

F

Fixed Point	Method of computation in which the position of the point in a number is determined by the operands and the type of operation producing it.
Floating Point	Method of computation in which a number is treated as two numbers, one giving the digits and the other implying the place values to be ascribed to them. (see Argument and Exponent).
Full Adder	A device used to add binary numbers together (*c.f.* Half Adder)
Function	The type of operation specified in an order.

G

Gate	An electronic circuit performing a logical operation on two or more sequences of electrical pulses.

H

Half Adder — Logical circuit that can perform the operation of adding a single digit to a number.

I

Initial Orders — A permanently stored programme containing Input *(q.v.)* and subroutines associated with the warning characters.

Input (Routine) — A permanently stored programme designed to compile orders and operands from characters punched on tape.

Inversion — The selection of the class of all the elements that are not members of a given class. In binary computers, the substitution of a 1 for a 0 and a 0 for a 1.

J

Jump — In a computer programme, the transfer of control from one order to another involving a break in the normal sequence; usually conditional on some form of test.

Justification — 'Carry' operation between two halves of a double-length number. Also the formation of a double-length number from a single-length number outside the standard range.

L

Logical Operation — The selection of one class out of a larger set according to some stated rule.

Loop — A series of orders obeyed cyclically and constituting one step in a repetitive process.

M

M-bus — Output line fed from the X/M bus *(q.v.)* and used for routing modifiers into the order register.

Mill — The main arithmetical unit of the machine.

Modifier — A series of digits added to the digits of an order to alter the addresses specified.

N

N-Address — Any address in the computing store. The part of an order specifying such.

N-Bus — Output line common to all parts of the computing store.

N-Line — One of the forty-eight ordinary registers.

Normalise — To put the argument of a floating-point number into standard placing. In Pegasus, with the two digits immediately following the point complementary.

NOT-EQUIVALENT Operation — The selection of a class whose members possess one only of two specified characterising features; especially the selection of one of those digit places in which 'one' appears in only one of two binary numbers.

NOT Operation — See Inversion.

Numeration — The expression of numbers in symbolic form.

O

Octal Scale — Numeration scale with radix eight.

Order Register — Register providing temporary storage for an order while it is being obeyed.

Operand — Any quantity entering into a logical or arithmetical operation.

Order — Instruction to a computer to perform a single operation.

OR Operation — The selection of a class whose members possess one, some, or all of certain specified characterising features; especially the selection of those digit places in which a 'one' appears in at least one of several binary numbers.

Overflow — The formation of a number outside the permitted range of values.

P

Parity — Attribute of the number of 'ones' in a binary number, i.e. whether odd or even.

Partial Product	The product of a number (multiplicand) by one of the digits of another number (multiplier).
Place	The position of a figure in a series, as indicating its value in decimal or other notation.
Place Value	The value to be ascribed to a figure by virtue of its place.
ϕ-Pulse	A pulse occurring at word frequency. There are as many different ϕ-pulses as digit-times in a word-time.
Programme	A sequence of instructions for computation.

R

Radix	(*pl.* radices) Number or symbol used as the basis of a numeration scale.
Register	A rapid-access storage unit.
Rhythm	A sequence of beats governing the operation of a computer.
Routine	See 'Subroutine'.

S

Shift (arithmetical)	The multiplication or division of a number by some power of the radix.
Shift (logical)	A positional (timing) shift of a series of digits.
Sign Digit	A digit from the value of which the sign, + or -, of a number can be inferred. In Pegasus, the digit next before the point.
Significant (least)	Used of the places with the lowest values in a series of figures, or of the figures occupying those places.
Significant (most)	Used of the places with the highest values in a series of figures, or of the figures occupying those places.

Standard Timing — A certain relationship between the digit pulses of a word and the timing pulses. In Pegasus, a word is in S.T. when its least significant digit coincides with $\phi 0$.

Staticiser — A circuit capable of being set in either of two conditions and of maintaining that condition until reset.

Subroutine — Any part of a programme that is required frequently and can be incorporated in other programmes (also 'routine')

T

Timing Pulse — See ϕ-pulse.

Transfer — The copying into one part of a computer of information stored in another part. Unless special provision is made for erasure, the contents of the originating store remain unaffected.

Transfer of Control — See Jump.

Truth Table — Table in which membership of a certain class, or logical combination of classes is denoted by a 1 in the appropriate position, and non-membership by a 0.

W

Word — The longest series of binary digits that can be stored or used in one operation. In Pegasus, equivalent to 40 binary digits.

Word-Time — The time occupied by one word and its associated gap digits. In Pegasus, 42 digit-times or approximately 127μ sec.

X

X-Bus — Output line fed from the X/M-bus and used for routing operands to the arithmetical circuits.

X-Line — See Accumulator

X/M-Bus — Output line common to all accumulators.

LIST OF SYMBOLS

LOGICAL-DESIGN SYMBOLS

Waveforms

Capital letters are used to denote types of waveforms as follows:

A, B, C, D, E	Basic rhythmic waveforms.
F (0 to 5)	Outputs of function-decoding staticisers.
G (00 to 07 and 10 to 17)	Combined F-waveforms.
H (1 to 9)	Outputs of drum 'read' amplifiers.
I (0 to 17)	Order-register outputs.
J	'Jump' action waveform.
K, L	Basic rhythmic waveforms in multiplier-divider.
M (1 to 10)	Action waveforms for multiplication.
N	Combinations of S1 to S7.
N00	Gating for accumulators.
N10 to N15	Gating for ordinary-register blocks.
N01, N02, N04	Gating for special-register blocks.
N20 to N27	Gating within blocks.
Q (1 to 5)	Action waveforms for division.
R (00 to 04)	Track-selection decoding (columns).
R (10 to 17)	Track-selection decoding (within columns)
S (1 to 7)	Outputs of N-decoding staticisers.
S (8 to 10)	Outputs of X/M-decoding staticisers.
S (21 to 26)	Outputs of track-selection staticisers.
S (27 to 31)	Outputs of punch-selector staticisers.

T	Combinations of ϕ-pulses.
U	Combinations of rhythmic waveforms.
V	'mix' of X-waveforms.
W	'mix' of Y-waveforms.
X	Action waveforms.
Y	Operands.
Z	Leads for handkeys and monitor switches.

ϕ-pulses are denoted by a number alone.

A subscript number indicates the output of a cathode-follower.

Levels

The standard signal level (output of line, delay or inverter) is denoted by the Greek letter α. The level drops to β when the signal passes through a cathode-follower (which may be preceded by a gate), and to γ after a subsequent cathode-follower operation. A minus is used to denote the drop in level produced by a 'mix' type OR operation.

Loads

The fixed cathode load on the output stage of a line, delay or inverter is denoted by the letter a. This may be 'paralleled' by resistors, or a parallel combination of resistors, denoted by the letters b, c and d, to give adequate current for 'holding-down' gates.

PROGRAMME SYMBOLS

Addresses

The capital letters N, X, F and M are used to denote the four sections into which an order is normally divided, or the numbers given in binary form by the sections of an order. The letter N may be used to represent the first seven address digits of a putative order or the ten address digits produced after modification. The three modifier-extension digits are referred to by the letter E.

The letters N and X are also used to represent respectively any computing-store address and any accumulator. The letter may be followed by a number to give a precise address; thus -

X3 = accumulator 3
N15 = handkeys.

P and Q are sometimes used to denote accumulators 6 and 7 respectively.

Blocks in the main store and computing store are denoted by the letters B and U respectively. (Note that U7 represents the accumulators).

S represents a single-word address in the main store.

Operands

Operands are in general represented by the lower-case equivalent of the address symbol, thus

n	=	content of register
x	=	content of accumulator
1 or $x1$	=	content of accumulator 1
p	=	content of accumulator 6
q	=	content of accumulator 7
s	=	one word contained in main store
b	=	content of main-store block
u	=	content of computing store block.

A prime ($'$) is used to denote the content of an address at the end of an operation.

The letters m and c are used as subscripts to denote the 'modifier' and 'counter' parts of a word in store.

INDEX TO VOLUMES