The design and construction of a low cost visual display unit

Alan E. Adams

ABSTRACT

After discussing the techniques employed in alpha-numeric V.D.U.s, and the style of storage used in map displays, the thesis goes on to describe a screen addressing system. This addressing system allows the display of detailed and varied designs on a raster-scan display, without recourse to the large stores required to implement the map technique.

The addressing technique is used in a simple V.D.U. which, in order to minimise costs, uses a domestic television monitor as an output device. The display refresh store is made up of M.O.S. shift registers, and the store contents are organised under microprocessor control. Though the unit described is primarily designed for the display of circuit diagrams it is easily adapted to the display of a wide range of graphic formats.

THE DESIGN AND CONSTRUCTION OF
A LOW COST VISUAL DISPLAY UNIT

A Thesis submitted to the
University of Durham for
the Degree of Master of Science

Alan E. Adams, B.Sc., C.Eng., M.I.E.E.

Department of Applied Physics and
Electronics,
Science Laboratories,
South Road,
Durham.

February 1977

CONTENTS

Page

Page

LIST OF ILLUSTRATIONS

Figure Page

Plates

Flowcharts

ACKNOWLEDGEMENTS

I gratefully acknowledge the debt owed to Dr B. J. Stanier of Durham University who suggested and supervised this project and who was always ready to enter into stimulating and useful discussion.

Thanks are also due to Professor D. A. Wright, late Head of the Department of Applied Physics and Electronics at Durham University, who made available the facilities of that department. Equal thanks are due to the late Rector, Dr M. Hutton, the present Rector, Dr E. A. Freeman, and the Head of the Department of Electrical, Electronic and Control Engineering, Mr J. H. Jones, of Sunderland Polytechnic for allowing the facilities of that institution to be employed in the pursuit of this research.

Thanks must also be extended to my colleague, Mr M. Todd, who designed the cross-assembler program which was used to develop the microprocessor software and produced the listing of Appendix I.

To complete this acknowledgement I would like to thank Miss E. Moss for her patience and her excellent typing, and my wife for her tolerance and her help in the preparation of diagrams for the final presentation of this thesis.

A. E. Adams,
Department of Electrical, Electronic
and Control Engineering,
Sunderland Polytechnic.

INTRODUCTION

Visual display units have been available as computer peripherals for several years and almost from their introduction they have fallen into two distinct groupings.[1] There are simple, relatively cheap, machines with a purely alpha-numeric capability that have been used as a quieter, and perhaps more acceptable alternative to the teletype (though usually without the facility of a hard copy), and at the other end of the scale there are sophisticated interactive graphic systems where a wide range of facilities is available, and where every effort has been made to make the machine adaptable to many forms of problem.[2]

In discussions between Dr. Stanier of the Department of Applied Physics and Electronics at the University of Durham, and the author, in September 1974, it was decided that an attempt should be made to develop a unit that would go some way towards filling the gap between these two extremes. It was considered that a device which was capable of displaying a greater range of symbols with greater adaptability than the conventional alpha-numeric unit, and yet which did not attempt the extremely expensive sophistications of the larger graphics systems would find a useful place as a peripheral to, for example, a minicomputer, whose processing capabilities exceed the limitations of an alpha-numeric output but where the high cost of a commercially available graphics system could not be justified. An example of such a requirement might occur in a small college, or university department, where it would be very convenient to have a graphical display device linked to a small computer and capable of presenting the results of a circuit analysis program.

The unit described in the text is indeed biased towards the display of electrical circuits but it should be stressed that the principle of the design is quite general. A change in display format, and even in the method of display generation (e.g. using software techniques to generate information for the display of arcs, vectors, etc. rather than using pre-stored symbol information), is achieved simply by altering the software which, being held in Read Only Memory (R.O.M.), is a relatively inexpensive procedure. The hardware section is quite oblivious to the symbol format and the method of information generation and so requires no alteration whatsoever in order to display images other than those used in electric circuits.

For any V.D.U. the choice of the final electrical/visual converter is fairly straightforward, i.e. the use of some form of cathode ray tube (C.R.T.), however the development of Charge Coupled Devices (C.C.D.) and flat panel displays means that in the relatively near future this decision will no longer be so clear-cut. The use of a C.R.T. means that another primary decision must be made as to what form the C.R.T. should take. Should a storage tube be used, or a system utilising a raster-scan technique be employed? It is thus necessary to consider the relative advantages and disadvantages of the two types of tube.[3]

The major advantage of the storage tube, as its name implies, is that the display information is stored within the tube itself. This reduces the "external" storage requirement. The control of the writing beam is inherently an analogue problem and though this means that it is possible to display very intricate designs it does necessitate the conversion of information from a digital form, compatible with the driving processor, into an analogue form suitable for display. It is of course possible to use the storage tube as a digital display unit if the image is considered to be made up of a series of dots rather than continuous lines or curves, but this does eliminate one of its major advantages. The image produced by the storage tube is particularly acceptable under low ambient light conditions where the clarity of the display is most effective and there are no problems with flicker (as might be experienced with a raster-scan system). However the storage tube image is less acceptable under high ambient light conditions as the increase in the energy of the beams necessary to increase the brightness of the display leads to a dispersion of the image with a consequent loss in clarity. It should also be remembered that there is a limit to the display intensity which, if exceeded, will permanently damage the storage surface. Because the main storage medium is the tube surface selective erasure is not a practical proposition (though in some cases it is possible to erase one half of the display independently of the other half). Thus if only one section of the display is to be altered it is necessary to delete the whole display and then regenerate it. The regeneration time can be quite considerable for a complex display.[4] The need to regenerate the display after erasure necessitates either the provision of a considerable amount of local storage to hold the information as to the status of the display before erasure (thus offsetting, in part, one of the major advantages of the storage tube, its ability to hold data "internally"), or alternatively the dedication of the primary driving computer for long periods in order to reconstruct the display directly.

The raster-scan method consists of a continuous and regular sweeping of an electron beam across the face of a C.R.T. The deflection of the beam is controlled such that a series of horizontal (or vertical) lines progress across the screen in a vertical (or horizontal) direction, forming the raster. The image itself is formed by controlling the intensity of the electron beam and hence the brightness of the display at a particular position on the screen. The obvious disadvantage of using such a technique in a V.D.U. is that, unlike the storage tube, the C.R.T. across which the raster scans does not permanently retain the image and thus a memory external to the tube must be provided. However in the author's opinion the advantages of using a raster-scan technique, as opposed to a storage tube, in a low cost V.D.U. outweigh the above mentioned deficiency. Although the control of the raster-scan is essentially an analogue problem, the fact that the raster consists of a series of discrete lines, which in turn may be split up into sections, allows the display to be managed in a digital manner. The brilliance of the display may be varied over a much wider range than with the storage tube in order to suit the ambient light conditions. Indeed it is a simple matter to change the mode of the display, from a white image on a black background to a black image on a white background. The disadvantage of having to store the displayed information externally is balanced by the fact that this allows considerably greater control over this information: allowing, for example, selective erasure or modification of information, so that part of the image may be changed without the necessity to erase the whole display and then regenerate it (as with the storage tube). It is also the case that the C.R.T., together with its raster driving circuitry, is considerably cheaper than a storage tube of equivalent size.

The advantages of the raster scan method compared with the storage tube technique persuaded the author that this, the raster scan approach, was the most suitable for this project, particularly as, with the falling cost of solid-state memory devices, any benefit to be gained from having information storage within the tube was being rapidly eroded. In order to keep costs to a minimum it was decided to design the system so that the output signals used to drive the display would be of a form compatible with the video and synchronisation signals used in a domestic 625-line television monitor.[5] It was anticipated that such monitors would be readily available in the situations where it was intended that the V.D.U. would be used, and that they could easily be converted to perform both in their conventional role and as an output unit to the V.D.U.. This would effectively reduce the cost of the display device itself to a few pounds. A further advantage to be gained from using the raster scan technique, as opposed to the storage tube, and making the control signals consonant with those used in domestic televisions, is that it is a relatively cheap and simple matter to disseminate the display to several monitors for simultaneous presentation, a particularly useful arrangement in a teaching situation.

Having decided to use the raster scan method considerable thought was given to the major problem inherent to this system, i.e. the storage of the information which is used to refresh the display each time it is scanned. Chapter 1 discusses the storage techniques which were considered in an effort to develop an efficient display memory. The design of the unit follows from the choice of a particular memory format, the general principles of the design being set out in Chapter 2. At an early stage in the design procedure it was decided that the unit would be considerably simplified, and costs kept to a minimum, by the use of a microprocessor to control the presentation of display information to the refresh store. Such an approach would also make the unit easily adaptable to different forms of display, changes being made by means of alterations to the microprocessor software. The microprocessor used was the Intel 8008, in the form of the Intel SIM-6 microcomputer.[6] This proved to be quite adequate for use in the prototype unit which was designed, built and used to display pre-stored symbol information. However if the full potential of the V.D.U. is to be realised detailed information will need to be generated locally and it is then suggested that a more sophisticated microprocessor should be used. The hardware and software components of the design have been separated purely for descriptive convenience, and it should be borne in mind that many aspects of the software design will affect the hardware design, and vice versa.

The prototype system designed and built was able to display electrical components, though because of the limited store length (64 words), the amount of information that could be displayed simultaneously was restricted. The major problem encountered was the limitation imposed by the maximum shift rate of the store; this is explained more fully in the text. It is felt that the primary achievement of the project is described in Chapter 1, where a consideration of addressing techniques leads to the development of an efficient store organisation for the display of simple diagrams.

CHAPTER 1

STORE ORGANISATION

1.1 Introduction

This chapter considers various techniques for the organisation of a refresh store for use with a V.D.U. utilising a raster-scan technique. While raster-scan monitors are widely used for alpha-numeric-only displays, full graphic systems have tended to rely on storage tubes and random deflection units. This is basically because of the cost of providing the necessary memory for the display.

The alpha-numeric style store and the bit-per-element "map" system are considered first. The chapter outlines various addressing techniques, one of which was selected for use in the V.D.U. described in this thesis. It is believed that the choice of this particular storage format allows the freedom of display of the bit-per-element system to be implemented, without incurring the high cost of providing a "map" store.

1.2 Alpha-Numeric Style Storage

The general technique employed in an alpha-numeric V.D.U. is to split the screen up into sections. Each section is then referred to by quoting a horizontal code and a vertical code, as illustrated in figure 1.1. Each section coding also relates to a position in the refresh store which holds information as to which character is to be

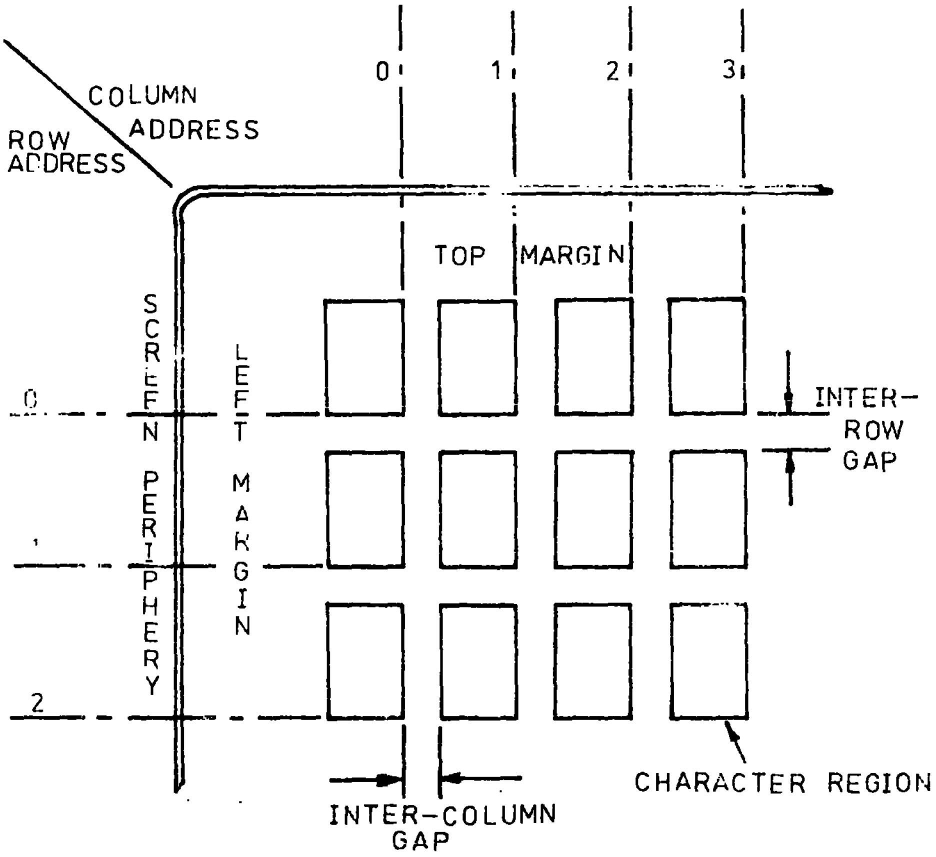


FIG. 1.1 CHARACTER POSITIONING

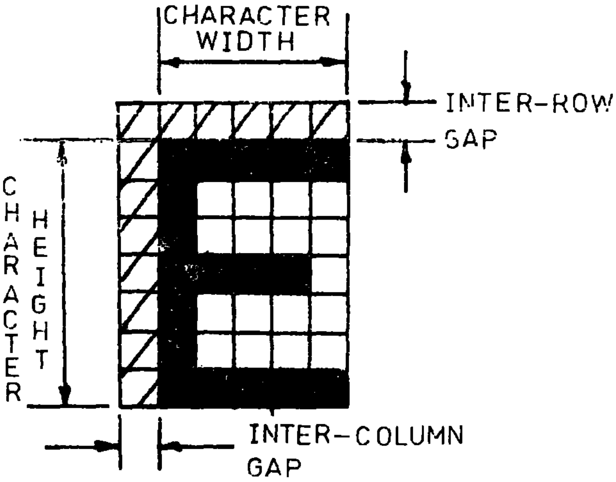


FIG. 1.2 STORED INFORMATION FOR CHARACTER "E"

displayed in this location. The individual character information is held in R.O.M. in a form directly equivalent to the style of display required, e.g. the contents of the character store for the letter "E" are as shown in figure 1.2. When the electron beam drawing the raster enters a particular section of the screen the part of the memory relating to that section is addressed, this in turn leads to the selection of the defined region of R.O.M. It is this information that determines the modulation of the electron beam as it crosses the addressed section. It will be appreciated that in describing a single character the beam passes through the relevant screen area several times, and each time the beam modulation must correspond to a different part of the character. This is achieved by the use of a line counter. This counter is set to zero when the beam starts to make the first sweep across a row of characters and is incremented for each complete sweep.

Many commercially available alpha-numeric displays have adopted a system whereby the character selection information is held in a recirculating shift register store and the selected R.O.M. data used to fill a subsidiary store, which holds the information equivalent to a single horizontal scan across the screen. This single line store is used to modulate the electron beam intensity during that scan. Falling costs have led to the use of Random Access Memory (R.A.M.) in place of the dynamic shift register as the main storage element.[7] Improvements in response time of both R.A.M. and R.O.M. have led in some cases to the elimination of the subsidiary store (holding information for a single horizontal line). The technique adopted is to latch the information relating to the selected character row from the R.O.M. into a 5-bit register (the character font for this form of display being almost universally a 5 x 7 dot matrix). The individual bits of this register are selected, via a multiplexor, to define the display intensity.

The technique described above is very efficient when the form of the display is derived from a limited range of symbols which may all be contained in blocks of similar size (as in the case of alphanumerics). The required storage is then merely a record of which character occupies a particular position and the information relating to the style of individual characters. It should not be thought, however, that this storage requirement is insignificant. A widely used V.D.U. which has a maximum display capability of 18 rows of 80 characters, has a memory capacity of 1,536 (3 x 512) 8-bit words, a total of 12,288 bits.[8] The applicability of this technique is reduced if the display requirement is for other than a standard range of characters, or if characters of varying overall size are to be displayed.

1.3 The Bit-Per-Element Store

The essence of this technique is to reduce the display to a matrix of small areas or elements, each element being represented by an equivalent bit position in the memory. The contents of the memory are examined in sequence as the raster is drawn on the screen, the display intensity at a particular position being modulated in accordance with the contents of the relevant bit position in the store; i.e. if the store content pertaining to a particular screen element is "1" the display is intensified, if the store content is "0" the beam intensity is kept at the black level. Such a memory is known as a display map, since the pattern of ones and zeros that form its contents is an imitation of the required display output.

The advantages of such a direct equivalence between display and refresh memory are obvious. It allows complete control over the pattern presented on the screen, quite free from any restrictions of format or position that are inherent in the alpha-numeric style store. Although during the read-out procedure examination of bit status would proceed sequentially through the store, there is considerable advantage in using R.A.M. as opposed to sequential memories such as shift registers, since this would allow the easy manipulation of information anywhere within the store without the delay necessary to search through large amounts of unchanged data.

The great disadvantage of the bit-per-element store is its large size. If we consider a 512 x 512 element display then the map memory required would be 262,144 bits. The cost of such a store, even with the falling price of solid-state memories, is prohibitive. In order to cope with this very large storage requirement, systems utilising the bit-per-element technique, such as the "PICASSO"[9] system, a 512 x 512 element display developed at Reading University, or the "Intergraphic"[10,11], a 1,024 x 1,024 element display designed at the University of New South Wales, have tended to employ storage devices where the cost-per-bit is relatively low, in the case of both the aforementioned designs a disk memory. Of course although the cost-per-bit of a disk system is low the overall cost of a disk memory is high. Both systems reduce the cost per terminal by driving several displays simultaneously, five in the case of "PICASSO", thirteen in the case of "Intergraphic".

An illustration of the continuing fall in the cost of solid-state memories was given in the announcement by the Intel Corporation of the production of a printed circuit board for use as a 512 x 512 map memory. The unit was first advertised in the early summer of 1976 when the project described by this thesis was in its final stages. The quoted one-off price in July of the same year was £1,200, still an excessive cost if considering the type of application for which the V.D.U. described here was designed. However it is an indication perhaps that hardware costs may, in the not too distant future, fall to such a level that the consideration of efficient storage techniques is of a lower priority than at present. It is also apparent that the development of reliable, low-cost flat-panel displays will have a dramatic effect on V.D.U.'s, being able to combine the flexibility and precision of displays using the bit-per-element store technique with the savings produced by having the storage built into the display medium.[12]

Some attempt[13,14] has been made to combine the relatively small storage requirements of the alpha-numeric display with the versatility of the bit-per-element technique. Here, as with the alpha-numeric system, the screen is split up into fairly gross sections and a listing kept of the format within each section. The system differs from that of the conventional alpha-numeric technique in that as well as pre-stored symbols it can generate its own vectors to be displayed within a defined section (long vectors being made up of the linkage of small vectors within adjacent sections). Within sections, the display is essentially of the bit-per-element type.

1.4 Image Addressing

When considering the type of refresh store to be employed by the V.D.U. described in this thesis it was decided that the alpha-numeric technique was too limiting, and that the bit-per-element store, while allowing great freedom of display format, could not be used because of the high cost of providing such a large storage capacity. The technique of providing low-cost storage by sharing a very high capacity store between several terminals, was inappropriate to the applications envisaged for this device, where the low cost of a single unit was of the highest importance. It was necessary therefore to devise a style of storage that would allow most of the facilities of the bit-per-element store to be employed, without incurring the excessive outlay inherent to such a system.

If we consider the type of image which may be displayed on a V.D.U. it will, in general, and more particularly in the case of the simple designs for which the present device is intended, consist of a skeleton of a relatively low number of lines, the great bulk of the screen being a backdrop against which the image is presented. It is only a very small fraction of the display area, and hence in the case of the bit-per-element store a very small fraction of the store, which actually contains the display information. Most of the storage area, in the bit-per-element system, is employed only to hold information to the effect that the display does not utilise that particular element. In this sense the great majority of the store capacity is wasted, particularly in the case of simple display forms. It is therefore very attractive to consider a storage technique in which only information relating directly to the image need be held, without having to store information concerning positions where the image does not appear. From this follows the idea of addressing the image, that is splitting the display down into very small elements, each of which may be referred to by an individual address, and then recording the addresses of the elements in which the image appears. This technique has the great advantage that a relatively simple image, requiring only a few addresses, needs only a small inexpensive store, a store whose capacity may easily be expanded at some later date when it is required to increase the display complexity.

The following sections describe the different techniques that were considered for the implementation of image addressing. In all cases the display considered consists of 512 x 512 elements. The choice of 512 vertical elements stems from the decision to make the display output compatible with a domestic 625-line monitor. In fact only about 605 lines are displayed on the screen, the other 18 to 22 lines being lost during the field blanking period.[5] If we utilise only 512 of the 605 lines each line may be selected using a nine bit address. It was decided that sufficient image definition could be achieved by splitting each horizontal line into 512 sections, each section being defined by a nine bit address. The choice of the number of horizontal sections is more arbitrary than that of the number of vertical sections, where the image is already quantised into a predetermined number of lines. The number of horizontal sections must not be too small or the image will become unacceptably coarse, however the number of sections must not be too large as, because the time to sweep one line is fixed, the larger the number of sections in a line the greater the frequency at which each section must be accessed (see section 1.6).

In the following discussion then each individual element may be defined by a 9-bit line count (L.C), and a 9-bit intra-line count (I.L.C.), the address of each element being quoted in the form L.C.: I.L.C. All images are considered, for the sake of simplicity, to consist of bright lines on a dark background, but the arguments apply equally well to a dark image on a light background.

1.4.1 Element-by-Element Addressing

With element-by-element addressing the address of each element that is intensified as part of the display is recorded in the store. It will be noted that in order to display horizontal lines using this technique it is necessary to store all the addresses of consecutive elements along the defined line. As each address (L.C.:I.L.C.) requires 18-bits it will be appreciated that this technique requires a relatively large store, even if only a few horizontal lines are to be presented.

1.4.2 Begin-End Addressing

The begin-end addressing technique consists of recording the address of the start of an intensified section of display and also the address of the end of th.. intensification. Thus any intensified part of the display, be it a single horizontal section, a complete line, or indeed the whole screen from first section to last, would be defined by quoting two addresses, the L.C.:I.L.C. of the start of the intensification plus the L.C.:I.L.C. of the end of the intensification, a total of 36 bits (two 18-bit words). It can be seen that this system is considerably more efficient than the element-by-element technique for the display of horizontal regions of more than two sections. However it is considerably less efficient for displaying vertical lines, which, being made up of individual intensified sections in several horizontal scans, must be referred to by a series of 36-bit, two-part addresses, as opposed to a series of 18-bit addresses in the case of the element-by-element addressing system.

1.4.3 Begin-End+Flag Addressing

The great disadvantage with the begin-end addressing technique is the necessity to quote two addresses when only a single section is to be intensified. This drawback may be eliminated by introducing a flag bit into the address thus increasing each address word to 19 bits, (a flag bit, a 9-bit line count and a 9-bit intra-line count, F:L.C.: I.L.C.). The status of the flag bit is used to define whether the change in the intensification of the display, at the position referred to by the L.C.:I.L.C., is to be "permanent", that is it is to last until the position defined by the next address in the store, or if it is to be "temporary", that is it is to last for one section only. Thus a single horizontal section may now be intensified by quoting only one 19-bit address. Any intensification of longer than a single section is defined by a 38-bit two-part address (begin F:L.C.:I.L.C., end F:L.C.: I.L.C., F = 0 in both cases). This combines the single section addressing advantage of the element-by-element technique (at the cost of one extra bit in each address) with the longer section addressing benefits of the begin-end addressing system (at the cost of two extra bits per intensified region).

1.4.4 Begin-Length Addressing

With begin-length addressing the start address of any intensified portion of the display is held in the store (as an L.C.:I.L.C.) together with the number of sections for which the intensification is to last. The number of bits allotted to the storing of length intensification information obviously defines the maximum amount of the display that may be intensified from a single address word. If we limit this maximum to a single line, of 512 sections, then 9 bits will be required to define the intensification length. Thus each intensified part of the display, from a single horizontal section to a whole line, will be defined by a 27-bit word (a 9-bit L.C., a 9-bit I.L.C. and 9 bits for the length of intensification). Intensifications of areas covering a larger fraction of the screen will be defined by several 27-bit words, each covering a maximum of a single horizontal line.

1.4.5 Begin-Length to Change-Length to Change-.....Addressing

The technique described in section 1.4.4 may be extended to include information not only as to how long the intensification is to last but also as to the gap before the next intensification is to occur, the length of the next intensification, the gap to the next intensification and so on. The word length to be employed with this technique will vary depending upon the number and size of intensified lengths we are to consider. If, for example, we allow for the definition of three length units, each a quarter of a horizontal line long, then a single address word will be 39 bits long, an 18-bit L.C.:I.L.C. plus three 7-bit words for length definition.

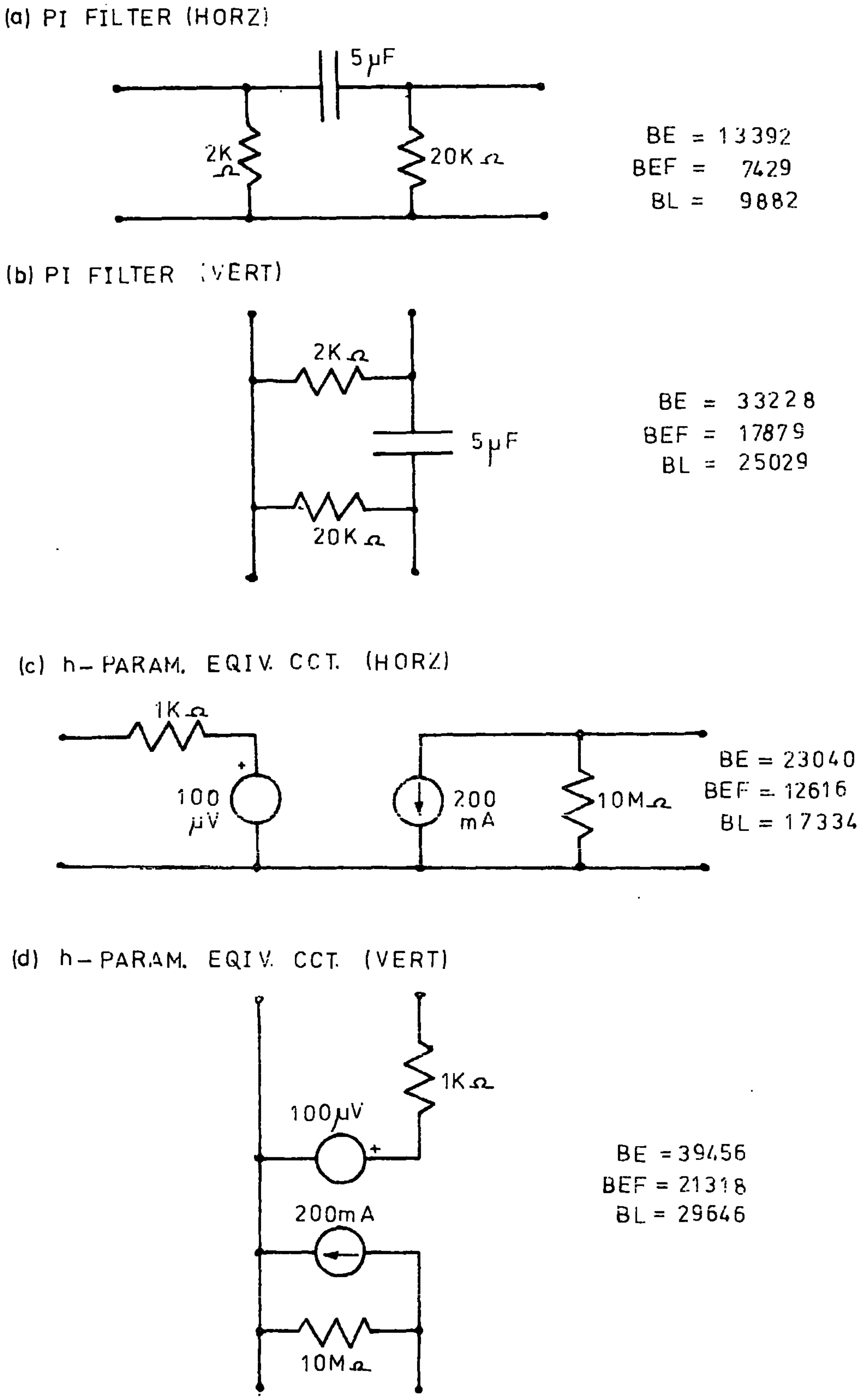
The use of this technique was not seriously considered since it would be very cumbersome to implement, especially if one thinks of the problems of rearranging the stored data when new information is to be inserted which refers to an image which overlaps with a design that is already being shown.

1.5 Comparison of Image Addressing Techniques

The element-by-element addressing technique described in section 1.4.1 was rejected because of the large amount of storage required to display horizontal lines. However it was thought that the begin-end addressing, the begin-end+flag addressing and the begin-length addressing techniques warranted further investigation.

The systems were compared on a bit-for-bit basis (the word length being different in each case). The number of bits required to represent characters A-Z, 0-9, μ, m, Ω, p, horizontal and vertical symbols for resistors, capacitors, inductors, voltage and current sources, and horizontal and vertical lines were determined for each of the selected addressing techniques. Characters were designed to be 10 lines high and 7 sections wide, horizontal symbols being 10 lines high and 93 sections wide, vertical symbols being 116 lines high and 10 sections wide. These dimensions were chosen as being suitable for use when displaying a 20-node circuit diagram on a domestic television monitor. Character height was designed to be 10 lines as the literature[15] suggests that for a high accuracy of identification (98-99% correct) the image should subtend an arc of 12 to 15 minutes at the point of observation, requiring a vertical resolution of between 8 and 12 lines. However it should be borne in mind that the greater the size of the image the larger the amount of stored information required to display it and thus the more expensive the overall unit. The characters finally displayed using the prototype unit were 5 sections wide and 7 lines high. This is the character format used in most alpha-numeric displays and was found to be quite acceptable in this instance.

The calculation of the number of bits required to display various symbols showed that the begin-end+flag system required the fewest number of store elements, the begin-length system requiring 36% more bits overall and the begin-end system requiring 81% more bits overall. The storage requirements for the different systems were then compared by determining the necessary store capacities in order to display the simple circuit diagrams illustrated in figure 1.3. Again



(a) PI FILTER (HORZ)

BE = 13392
BEF = 7429
BL = 9882

(b) PI FILTER (VERT)

BE = 33228
BEF = 17879
BL = 25029

(c) h-PARAM. EQIV. CCT. (HORZ)

BE = 23040
BEF = 12616
BL = 17334

(d) h-PARAM. EQIV. CCT. (VERT)

BE = 39456
BEF = 21318
BL = 29646

FIG 1.3 TEST CIRCUITS

the begin-end+flag system consistently required a smaller store capacity than either of the other two systems.

The reader's attention is particularly drawn to the comparison between the storage requirement to display the circuits shown in figure 1.3 using any of the image addressing techniques and the storage required if the bit-per-element approach were adopted, i.e. 262,144 bits.

Of course such tests are not exhaustive, there being a great number of different symbol combinations and display styles. However the results were sufficiently convincing to warrant the adoption of the begin-end+flag technique for use in the V.D.U. under discussion. As well as the smaller store required with this technique it provides a further very important advantage, as described in the next section.

1.6 Store Implementation

The adoption of a particular storage technique, in this case the begin-end+flag system, determines the basis for the rest of the design. The information defining the display will consist of a series of addresses which determine where on the screen the electron beam is to be intensified, either "permanently" or "temporarily" (see section 1.4.3). This means that the stored information must be accessed in the correct sequence (that is in the order of increasing L.C.:I.L.C.) if it is to be related to the beam position, which is being swept across the screen from the lowest order address to the highest.

The implementation of the begin-end+flag addressing technique (or indeed any of the other image addressing techniques) presents us with a major problem, that is the necessity to run the store at a speed which is fast enough to keep up with the sweep of the electron beam. Consider a single horizontal sweep of the beam, this is divided up intc 512 sections, each section having its own address. If the display is to be meaningful then the information defining changes in the intensity of the beam must be available, at the output of the store, before the beam reaches that section of the screen referred to by the next address. The minimum time to shift to the next address information will occur when a section intensification is followed by a single dark section, followed by a single intensified section, followed by a single dark section and so on. Under these conditions the information relating to the next address (which will be the address of the next section along the line) must be presented at the output of the store in less than the time it takes for the electron beam to cross the previous section. However with the begin-end+flag technique the flag bit describes whether the intensification relates to only the present (addressed) section, implying that the next section must be dark, or to more than one section, implying that the next section at least must be light too. Thus one address word in the begin-end+flag system defines the status of at least two adjacent sections. This in turn means that the store need only be accessed at half the rate necessary if any of the other addressing techniques had been used (see figure 1.4). However the employment of this longer access time does present problems when inserting new information which relates to an image which is to be immediately adjacent to one that is already held in the store (see section 5.1).

1.6.1 Store Hardware

As has already been explained, the use of the begin-end+flag addressing technique means that the addresses held in the store must be accessed in sequence and at a speed which is fast enough to keep

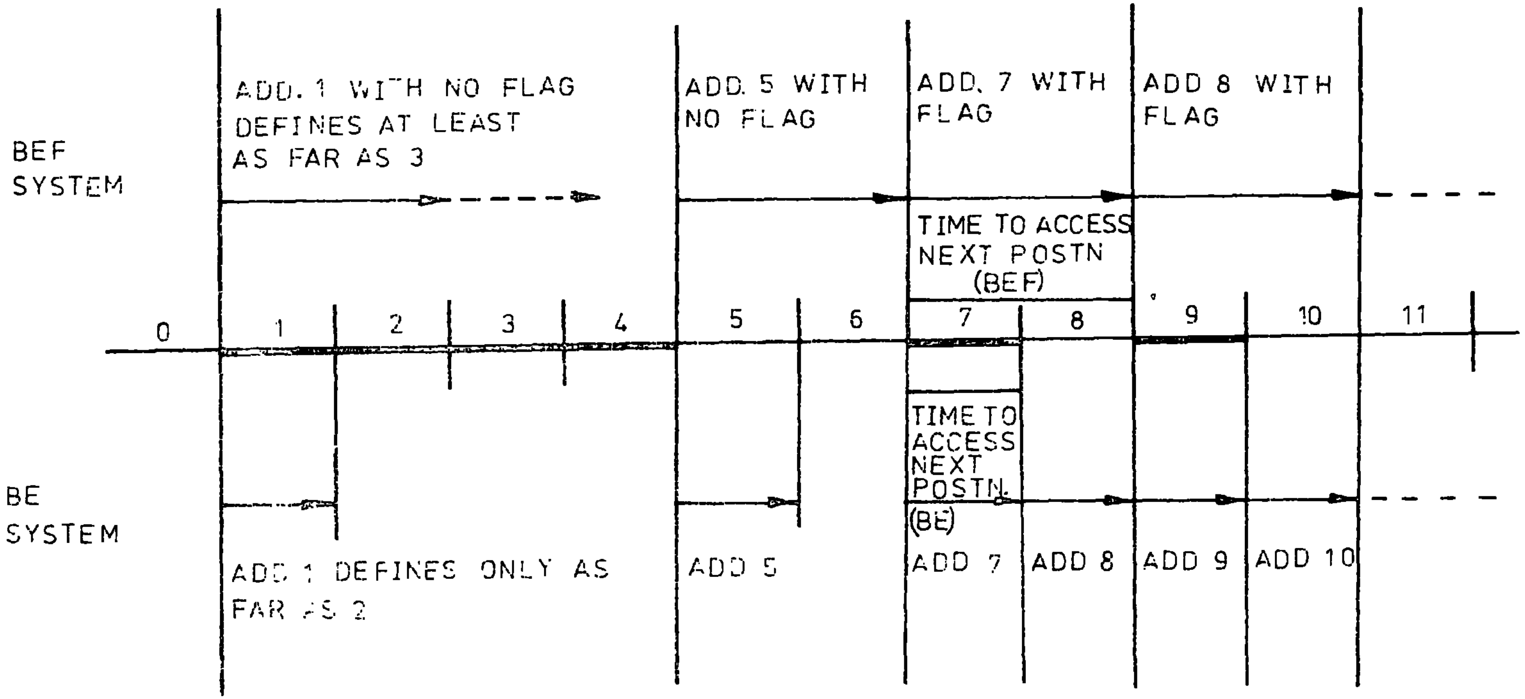

BEF SYSTEM
ADD. 1 WITH NO FLAG DEFINES AT LEAST AS FAR AS 3
ADD. 5 WITH NO FLAG
ADD. 7 WITH FLAG
ADD 8 WITH FLAG
TIME TO ACCESS NEXT POSTN (BEF)
0
1
2
3
4
5
6
7
8
9
10
11
TIME TO ACCESS NEXT POSTN (BE)
BE SYSTEM
ADD 1 DEFINES ONLY AS FAR AS 2
ADD 5
ADD 7
ADD 8
ADD 9
ADD 10

FIG 1.4 ACCESS TIME COMPARISON

pace with the sweep of the raster. The time allotted for a single horizontal sweep in the 625-line system is 64 μs[5]. However approximately 12 μs of this time are taken up by the line blanking period leaving 52 μs during which the beam may be intensified in order to present the display. As each line is split up into 512 sections it takes 102 ns for the beam to traverse a single section. With a conventional addressing system this would imply a store access frequency of 9.85 MHz. However with the begin-end+flag system under worst case conditions the maximum rate at which addresses must be retrieved from the store relates to alternate line sections, thus allowing a minimum of 204 ns between addresses, that is a maximum access frequency of 4.92MHz.

If the contents of the store are to be accessed in sequence then we have a choice between using shift register storage and R.A.M., accessed serially. At the time when the final decision had to be made as to which type of store to employ (summer 1975) the cycle time of suitable R.A.M. was of the order of 300 ns and the cost-per-bit greater than that of shift register storage, thus it was decided to construct the refresh store using shift registers. However since that time, and during the period when the prototype unit was being built and commissioned, the cost of R.A.M. storage has fallen considerably and the speed of operation increased, so that it is now possible to purchase a 1,024-bit R.A.M. with an access time of 70 ns for a little under £10.00.[16] The cost/speed relationship of shift registers however has not improved to anything like the same extent and if the project were beginning today it is highly likely that R.A.M. would be employed rather than shift-register storage.

The device chosen for use as the refresh memory in the prototype V.D.U. was the Motorola MC 14517 CP Dual 64-bit Static Shift Register. The quoted minimum value for the maximum shift rate for this device, operating from a power supply (V_{DD}) of 10 V, was 4 MHz, the typical value being 6.7 MHz.[17] At a V_{DD} of 15 V the quoted typical value is 8.3 MHz, no minimum value being stated. It was considered that by running the device at 15 V a shifting speed of greater than 4.92 MHz could be achieved. However it was found in practice that several of the devices purchased would not run at the quoted minimum value of 4 MHz, at 10 V, and that initially it was not possible to raise the value of V_{DD} above 13.5 V because of the problems caused by noise induced from other parts of the system. However after some design modifications a working store was produced.

It is possible to obtain a faster shift rate for the store as a whole than is possible for the individual units which make up the store. This may be accomplished by multiplexing the information between two parallel shift registers as shown in figure 1.5. The information is presented to the system at rate "A", alternate bits of information being shifted into SR1 and SR2 in turn by two out-of-phase clocks, each running at half the information frequency, i.e. the shift rate for both SR1 and SR2 is "A/2". At the system output the final stages of SR1 and SR2 are sampled in turn so that the information rate at the output is again "A." The disadvantage with this technique is the cost of providing the necessary gating which controls the flow of information through the paired shift registers. This would be insignificant compared with the cost of the store itself in a "practical" situation where the 19-bit wide store might be 1,024 or 2,048 words long and where the cost might be offset by the ability to

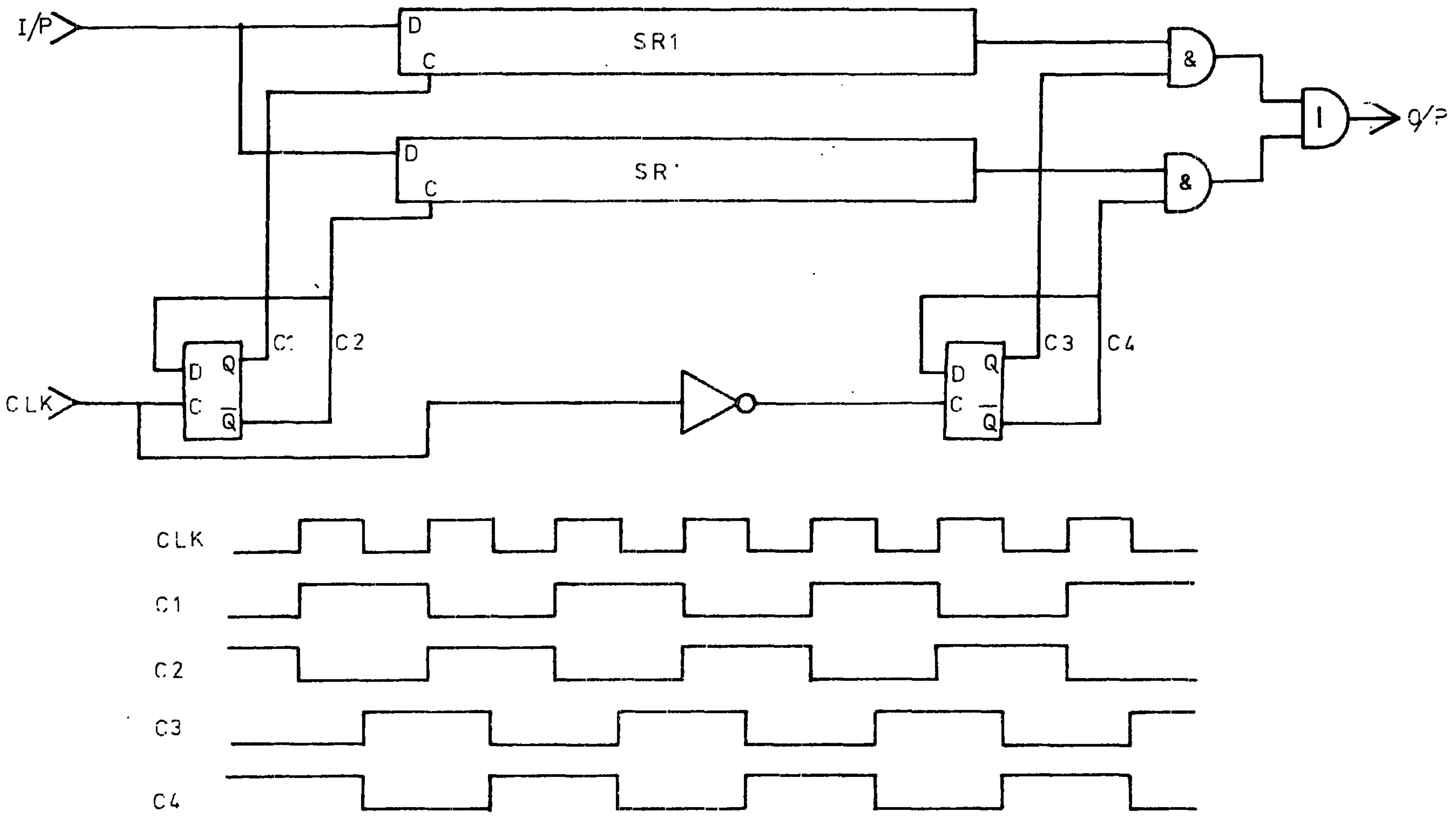


FIG1.5 INTERLACED STORE

use slower, and therefore cheaper, shift registers. It should be noted that the technique described above may be extended by having more shift registers in parallel with a consequent decrease in shift rate.[18]

Another form of storage which should be considered for use in any future unit utilising the techniques described in this thesis is the C.C.D. shift register. The possibility of achieving very high speed operation together with a very high packing density at a low cost makes this form of device a very attractive proposition. Such a device was in fact considered for use in the prototype unit. The Intel C.C.D. 2416, the only device commercially available at the time, consists of 64 recirculating registers each 256 bits long, giving 16,384 bits on a single chip (at a present cost of £21.98 per device). However this device had to be rejected, firstly because the maximum data rate of 2Mbits/sec was unacceptable - though the very high capacity of the store makes the idea of multiplexing between shift registers attractive - and secondly because of the necessity to refresh the stored information after short periods. In the case of the Intel C.C.D. 2416 the refresh cycle must take place at least every 9 μs. If we consider a display consisting of a single intensified dot then the information relating to the position of the dot will only be accessed once every frame period, that is every 40 ms. Thus the necessary refresh cycle for the C.C.D. store could not be guaranteed. This problem might have been overcome by introducing an otherwise unnecessary store shift every 9 μs but this was considered to be a needless complication. It is for this reason that the choice of shift register was limited to static devices, the dynamic shift registers available, which were capable of shift rates of 10 MHz and more, having to be rejected because their refresh rates were unacceptably high.

CHAPTER 2

SYSTEM OPERATION

2.1 Introduction

The design process required that the development of the hardware and the software should proceed, initially at least, in parallel, as the characteristics of one affect the characteristics of the other. Following the choice of a particular image addressing technique and the decision as to the form of hardware for the refresh store (see Chapter 1) it was possible to develop a basic design for the V.D.U. and decide how information might be fed into the store, or deleted from it, and how the information in the store would be used to control the image appearing on the screen. The next step was to consider the software required to control the microprocessor, in order that it should present the correct information to the store. This in turn led to restriction on the detailed hardware design. The software design was proved during the time the hardware design was being completed, and thus was ready to use during the hardware testing stage. When, after modification, the hardware design was shown to work it was again necessary to modify the software to take account of the problems which became apparent during the hardware testing.

2.2 General Organisation

Figure 2.1 shows the general organisation of the display system. Information from the controlling computer defines the screen position where a particular symbol (which may be a resistor, capacitor, or other circuit component in the case of the present design but could equally well, with software modification, be a vector, arc, circle, etc.) is to be displayed. The microprocessor then provides the detailed symbol information and the command signals which enable the refresh store to be updated in the correct manner. There are three basic operations under the control of the main computer, INSERT to introduce new information to the store, DELETE to remove selected information from the store and ERASE to clear the store. The main purpose of the hardware, as well as the correct organisation of the stored information, is the interpretation of the data held in store in order to control the intensity of the electron beam drawing the raster such that the required image appears on the screen. The hardware unit also provides the necessary line and field synchronising signals to control the raster. The flowchart shown in figure 2.2 describes the interaction between the controlling computer and the display unit.

In the case of the prototype unit, designed for the display of electric circuits, it is necessary for the main computer to present several sets of information to the microcomputer. These are the symbol status, that is whether the symbol aspect is vertical or horizontal, and the symbol direction, this applies to asymetric symbols such as current and voltage sources and relates to the conventional flow of current through the device assuming no other sources are present. For convenience where the current flow is from bottom to top, in the case of vertical symbols, and left to right

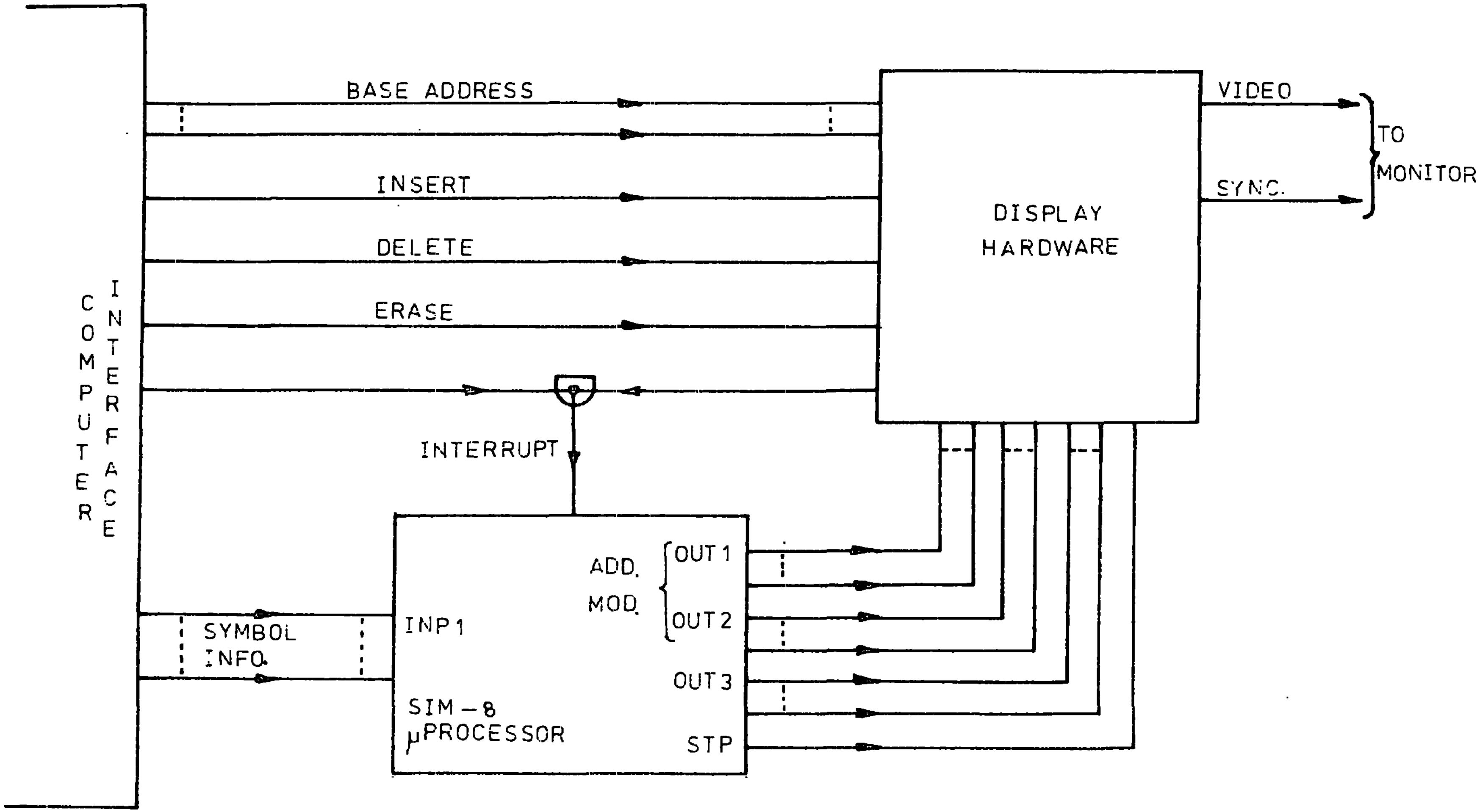


FIG 2.1 GENERAL ORGANISATION

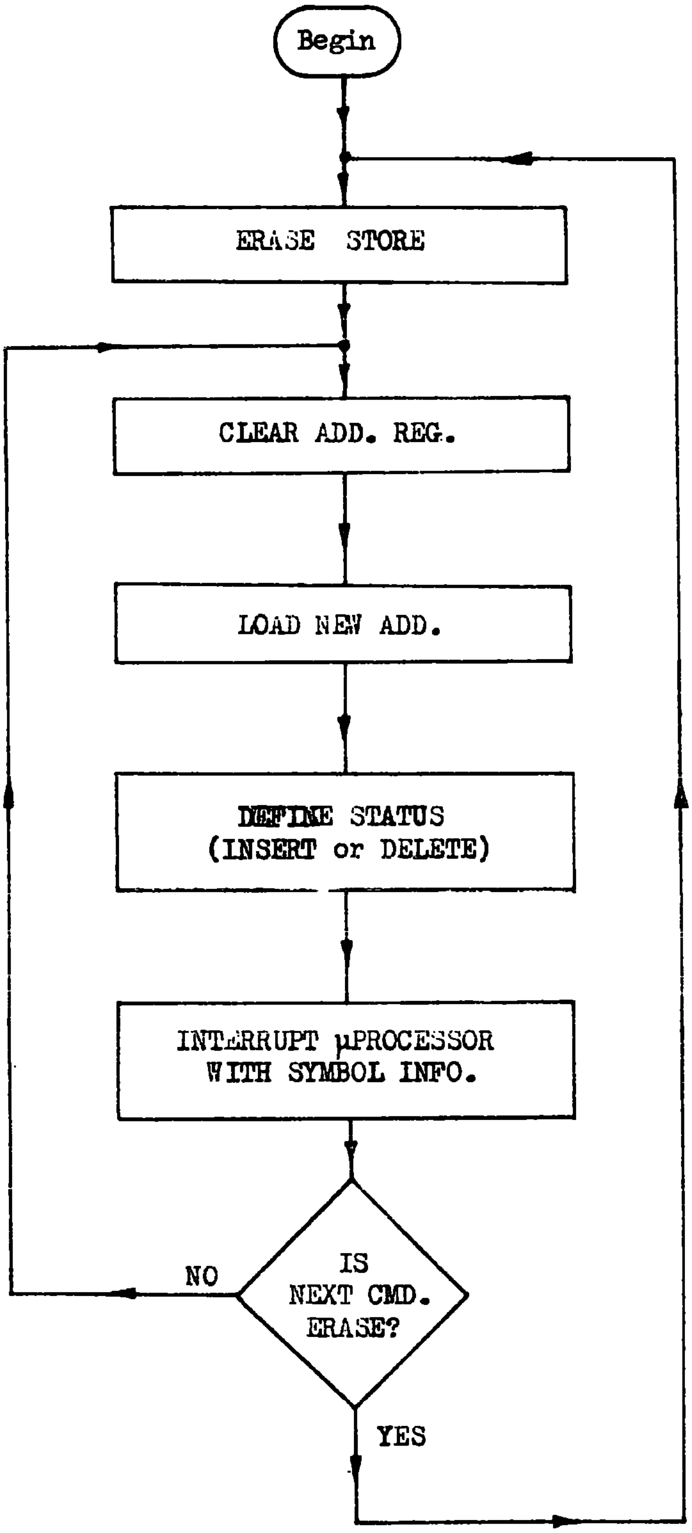


FIG. 2.2 The Operating Sequence

across the screen, in the case of horizontal characters, the symbols are referred to as UP symbols, when the current flow is in the opposite direction they are referred to as DOWN symbols. The next information that the main computer must present to the microcomputer is the symbol type (resistor, inductor, etc.). This is followed by the information relating to any characters which are to appear alongside the defined symbol. The program allows for the presentation of up to three numeric characters plus an exponent character alongside each circuit symbol. The exponent character must be either p, n, μ, m, k, M or a blank, this allows any number between 1 x 10^{-12} and 999 x 10^{6} to be displayed. In an effort to keep costs to a minimum no facility is allowed for the presentation of unit information (ohms, volts, etc.) as it is considered that this information is already contained within the display of a specific component symbol (i.e. if a resistor is being displayed it is implied that the units of the numerals beside it are ohms) and that to repeat this data with another character would mean taking up storage space unnecessarily. Indeed in an attempt to reduce symbol storage, image information which is common to all symbols, i.e. in the case of electric circuit components the leads, is stored separately. Thus when, for example, a resistor, is to be displayed the lead information is derived from a common source and the information which is special to the resistor symbol used to fill in the gaps between the leads as illustrated in figure 2.3.

The requirement then is for the master computer to provide the following information, plus an INSERT command, in order to feed the necessary data for the display of one character into the refresh store:-

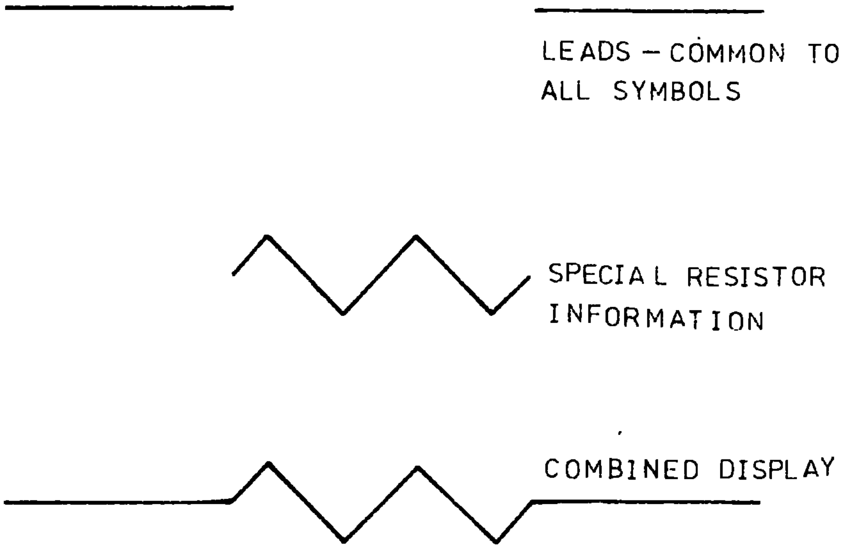


FIG 2.3 SYMBOL STRUCTURE

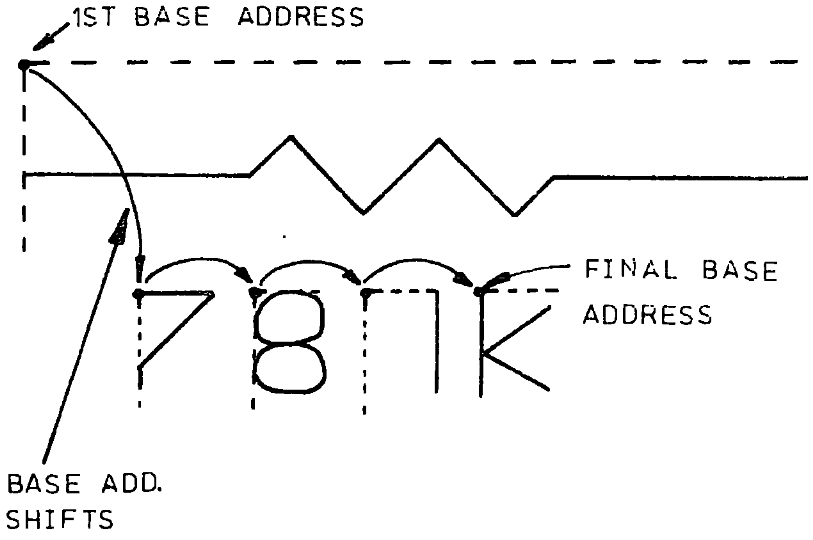


FIG 2.4 BASE ADDRESS MODIFICATION

(i) the base address for the symbol,

(ii) the symbol aspect, horizontal or vertical, UP or DOWN,

(iii) the symbol type, resistor, voltage source, etc.,

(iv) the first character, 0-9,

(v) the second character, 0-9,

(vi) the third character, 0-9,

(vii) the exponent, p, n, μ, m, k, M or blank.

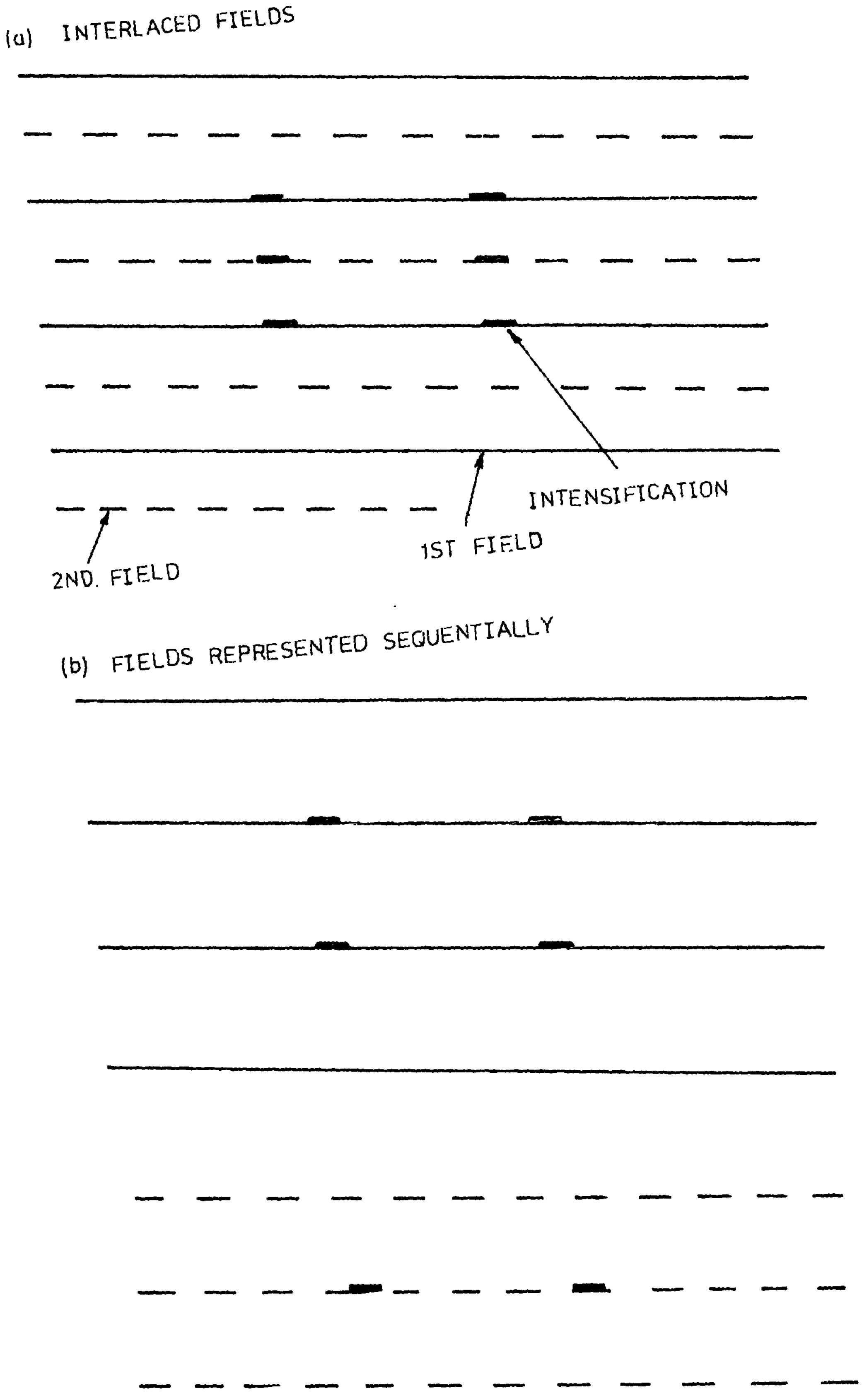
Parts (iv) to (vii) are not required if the symbol quoted in part (iii) is a short circuit (no magnitude information being appropriate in such a case).

In order to provide the refresh store with the necessary information to display a particular symbol at the screen position defined by the master computer, the base address is modified by the data which relates to the form of the required symbol. A particular component may be removed from the display by the same information sequence that is used for the insertion of information, the software being the same for both operations, it is merely the presence of a DELETE signal rather than an INSERT signal which distinguishes between the two actions.

2.3 Address Modification

The base address refers to the position on the screen which is in line with both the highest point on the screen and the furthest left point on the screen which the displayed symbol is to occupy. This allows all the required modifications of the base address to be of an additive nature, since all the elements used in the display of a particular symbol will have a greater line count and/or a greater intraline count than that of the base address. It is quite possible that the display element referred to by the base address will not be intensified as part of the symbol image, e.g. none of the electric circuit components use this element. The base address refers to that element with the lowest address that could be used in the display of a particular symbol.

The address word in the begin-end+flag technique is 19 bits long. Thus the base address is in fact an 18-bit word, since one of the 19 bits is the flag bit which takes no part in the definition of screen position. In order to store the display information then, the refresh store must be 19 bits wide. However the 8008 microprocessor uses an 8-bit word so, in order to allow for a larger degree of address modification, two of these 8-bit words were concatenated and used to alter the single 18-bit base address. This was achieved using two of the four output ports (referred to as OUT0, OUT1, OUT2 and OUT3) which are available from the SIM-8 microcomputer. The status of these output ports is held constant until new information is presented to them. Thus it was possible to present a 16-bit word, using ports OUT1 and OUT2, to modify the base address. Bits 0-6 of OUT1 were used to modify the intra-line count, being added to bits 0-6 of the base address. Bit 7 of OUT1 was used as the flag bit, which determines whether the display change, which is to take place at the element referred to by the address, will be "temporary" or "permanent" (see section 1.4.3). Bits 0-7 of OUT2 were used to modify the line count. Bits 0-6 of OUT2 were added to bits 9-15 of the base address (the line count using bits 9-17 of the base address, bit 18 being the flag bit). Bit 7 of OUT2 was added to bit 17 of the base address, this bit will be referred to as the field bit. The reason for this rather unusual distribution of the OUT2 data becomes apparent when one considers the manner in which the raster is drawn in a television monitor (see figure 2.5). One frame of the raster consists of two interlaced fields drawn one after the other, the lines of the second field falling between the lines of the first field. This allows a complete sweep across the screen (but not the complete display) fifty times a second. The complete image is presented only twenty-five times a second (as it requires two fields) and if interlacing were not used the time taken to produce the whole display in a single sweep would result in a flicker which would be perceptible to the observer. In our case where the number of addressable lines is 512, each field consists of 256 lines (in fact in order to produce the interlacing one field is reduced to 255½ lines). It is apparent then that the lines of different fields will be immediately adjacent to one another on the screen but will be widely separated in time. Any particular symbol will consist of a series of closely spaced elements and lines and is thus bound to use lines from both fields. This means that it must be possible for the base address modifiers to be able to produce addresses which refer to both fields. It will be noted that as half the lines appear in each field then half the number of elements will also appear in each field. Thus elements whose addresses are between 0 and 2^{17}-1 appear in field one (and have a zero in the bit 17 position of their addresses) and elements whose addresses are between 2^{17} and 2^{18}-1 appear in the second field (bit 17 of their addresses being a one). By assigning one bit position in our modifying word to the field bit position (bit 17) we are able to define addresses in either field, allowing symbols to utilise adjacent lines on the screen.


(a) INTERLACED FIELDS

(b) FIELDS REPRESENTED SEQUENTIALLY

FIG. 2.5 FIELD INTERLACING

It will be noted that allowance is being made for a seven bit modification of the intra-line count. Thus the maximum amount by which the intra-line count may be altered is 2^7-1, that is 127 sections or a quarter of the screen width. Seven bits are also allotted for line count modification in each field. This permits a line count change of 127, half the screen height, since there are only 256 lines in each field (equivalent to the full screen height). These maximum variations to the base address of a quarter of the screen width and half the screen height respectively, were considered to be quite acceptable for the display of circuit diagrams, the circuit components not requiring anything like this range of base address modification. However this range might not be acceptable for other applications. It does not, for example, allow the drawing of a diagonal line from one corner of the screen to another (though this might be achieved in four sections). The problem can easily be overcome by employing a third output port, i.e. OUTO. This would provide another 8 bits, of which only three would be required for address modification purposes. However if the system were to be built to utilise its own dedicated microprocessor it would be more attractive to consider the use of a device with a longer word length. The idea of using modular systems, which are usually based on 4-bit sections, that might be built up to 20 bits, is particularly attractive.

2.3.1 Base Address Shifting

As has already been mentioned, the prototype unit allows for the display of up to three numerals and an exponent character alongside each component symbol. If the numerals to be displayed were, for example, 999 then each 9 would occupy a different position on the screen and thus require different address information from its partners. This implies that three sets of modifying information are required for each character, for, although each of the 9 symbols is the same as its partners, its relationship to the base address is different. It is quite apparent that this duplication of character format information is unacceptable. In order to overcome this problem facilities were built into the system to enable it to change the base address to which the modifying information is applied. Thus, after the component modifications have been completed, new information is output from the microprocessor which changes the base address from the original value, quoted by the master computer, to a secondary datum point which is used as the base address for the first character. It is to this second base address that the individual character modifiers are applied. When all the information relating to the first character has been fed into the store another datum change is actioned to give a new base address for the second character (see figure 2.4). In practice this process is not as straightforward as this account may imply, as, if the initial characters are zeros these are not displayed on the screen in order to save on storage space. As the special relationship between the base address for the first character and the modifying information for the first character is the same as that for the second, and subsequent characters, base addresses and their modifiers, then the same character information can be used for all character positions, i.e. only one set of address modifying information need be stored for each character.

It is apparent then that the hardware is being presented with two different types of information, address modifying information, which is to be added to the base address and the result fed into the refresh store, and address changing information, where the base address itself is to be altered to form a new datum for the modifying process. Both types of information are output by the microprocessor on output ports OUT1 and OUT2, as described in section 2.1. It is essential then that the hardware is able to distinguish between the two types of data. This distinction is achieved by using the OUT3 output port of the microcomputer, the status of OUT3 being coded in a different manner in order to discriminate between the different forms of data. The OUT3 coding is set up after the OUT1/OUT2 coding in order that the modifying information is ready before the signal to action it appears. After setting up the OUT3 coding the microcomputer comes to a halt and presents a STP (stop) signal, it is this signal that is used to process the OUT1/OUT2 codings, the type of action being defined by the status of OUT3. The microcomputer remains halted until it receives an interrupt from the hardware showing that the last set of information has been processed. The processor also enters the halt state (thus presenting a STP signal) when awaiting an interrupt from the master computer, so a separate OUT3 coding is introduced to define this status.

The full range of OUT3 codings are defined below:-

(Note: All the output ports present an 8-bit word, coded in the form XX XXX XXX, which is represented here as a three digit octal word).

Microcomputer Status	Action
$\overline{STP}$	None (Run display)
STP.(OUT3 = 001)	Run display. Microcomputer awaits interrupt from master computer to read next symbol information from input port INP1.

Microcomputer Status	Action
STP.(OUT3 = 002)	Permanently modify datum (base address) with OUT1/OUT2 info. Interrupt when action completed.
STP.(OUT3 = 004)	Add OUT1/OUT2 contents to datum (without altering base address), feed result into store. Interrupt when completed.

2.4 Store Modification

Having considered the manner in which the information to display a particular symbol is derived from a base address and a series of modifiers, it is now appropriate to describe the way in which this information is applied to the refresh memory.

It has already been mentioned that the refresh store consists of 19 parallel recirculating shift registers. In order to facilitate the description of the three store modification processes three particular 19-bit registers .vill be referred to, these are register B, register C and register D. Another register is mentioned which does not form part of the main store but which is used during store modification, this is register A. Figure 2.6 shows the basic relationship between these registers, the main bulk of the refresh memory lying between registers C and D. The normal signal path, that is the path used by the data when the image is being displayed, is B, C (main memory), D, B, etc. However, during store modification, this path is varied, as described in the following sections.

2.4.1 The INSERT Operation

The most important point to bear in mind, when considering the manipulation of data within the store, is that the information must be arranged in strict order of ascending address. This is

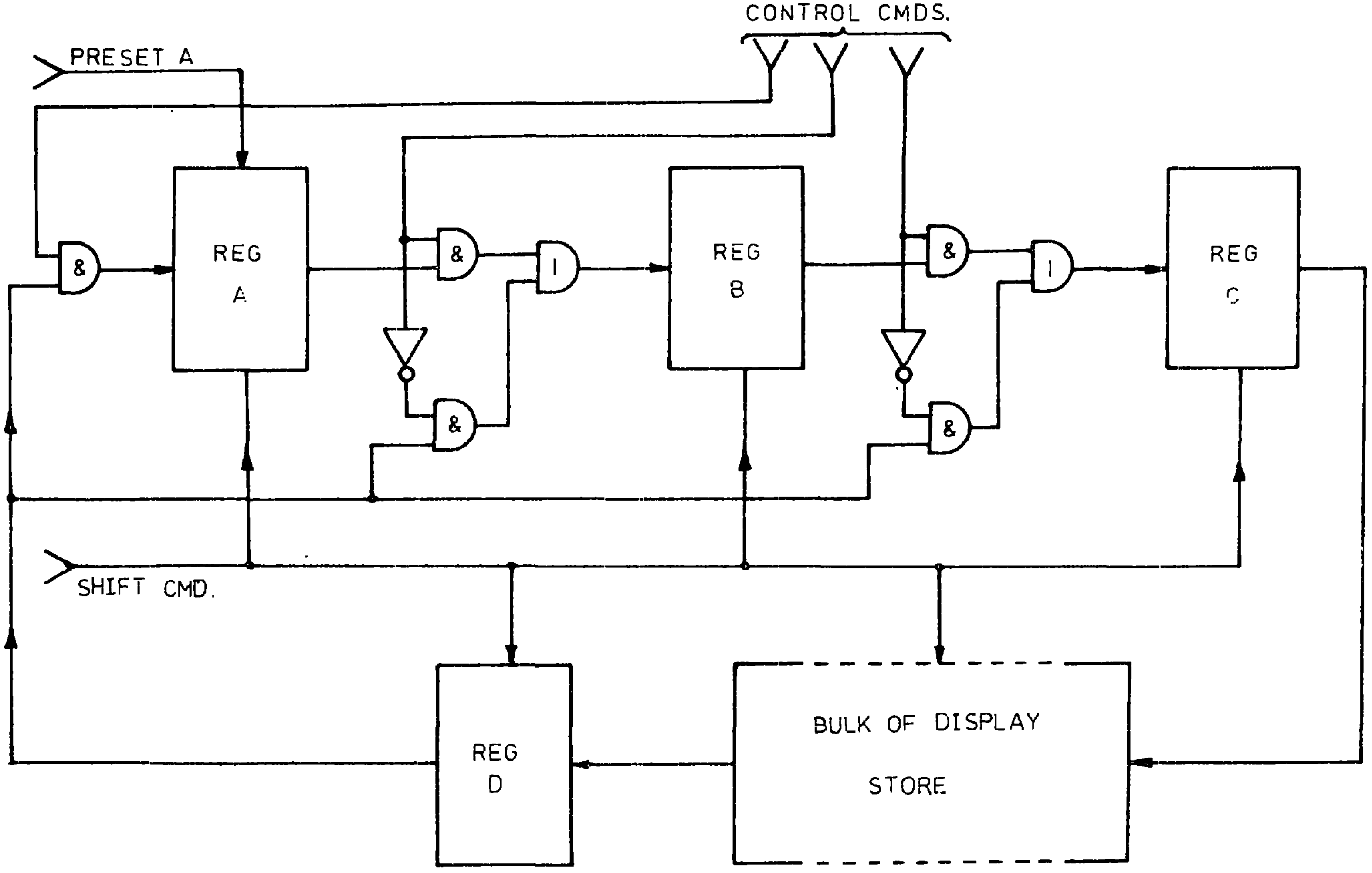


FIG. 2.6 REGISTER ORGANISATION

because, as the raster sweeps across the screen, it starts at element 0, in the top left-hand corner of the screen, and progresses, in order, through every element until it reaches the last element, element 2^{18}-1, in the bottom right-hand corner. Thus any information which is held in the refresh store must be stored in order, if it is to be accessed in order during the raster scan.

The problem of information insertion then is mainly concerned with keeping the stored addresses in the correct sequence. Let us consider a situation where a particular image is being displayed on the screen and that it is required to introduce a new symbol onto the screen. The information already held in the refresh memory will consist of a block of addresses, which refer to the displayed image, organised in ascending order, and continually circulating through the store, following the path, B, C, D, B, C, D, etc. If the store is M words long and there are N : .ored words of information ($N < M$), then there will be M-N empty word positions, which are all set to zero. The M-N empty word positions will follow the highest order address held in the memory. Thus in the memory we have an information block of ascending addresses and a block of all zero words, as shown in figure 2.7.

When new symbol information is to be fed into the refresh memory the INSERT command stops the normal read sequence of the system (described later) and uses the clock pulse to shift the information block through the store until the first address word is held in register D. The shift command is then inhibited so that no further shifting takes place. The new information, which is formed from the addition of an address modifier to the base address, is fed into register A. The fact that register A is other than zero, and that

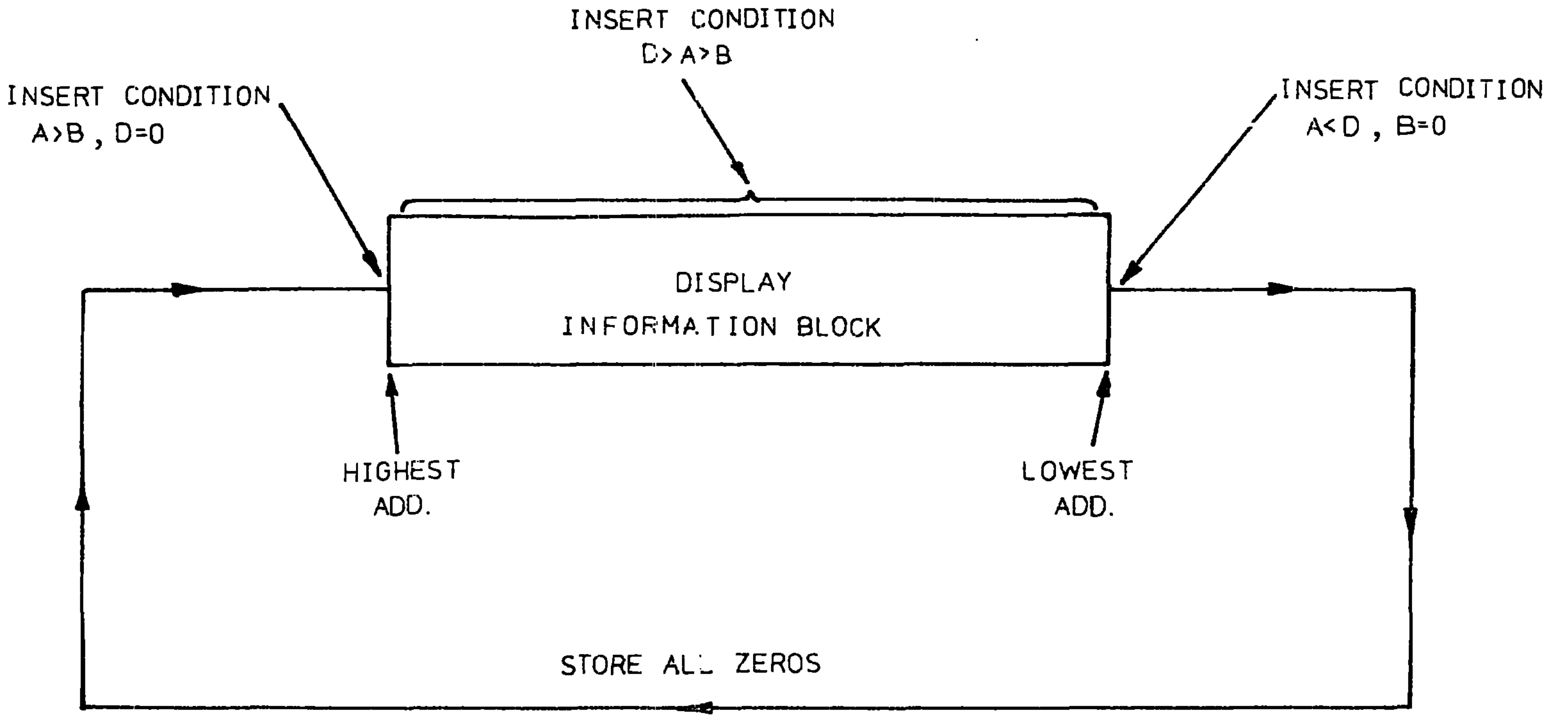


FIG. 2.7 THE DATA BLOCK

the store modification process is under way, defined by the condition STP.(OUT3 = 004), causes the store shift command to be actioned again. The normal information path is still E, C, D, B, C, D. However, in order to maintain the correct address sequence, the contents of register A are compared with the contents of registers B and D before each shift operation. If the address stored in A is less than the address stored in D, but greater than the address stored in B, then obviously the correct position for the information held in A is between the addresses held in B and D. If this condition arises then the signal path is changed to D, A, B, C ... (thus increasing the length of the memory by one word). After the next shift pulse the contents of D will be held in A, the contents of A held in B and the contents of B held in C. As the information held in the memory was already in order of ascending address, the condition D > A > B will still be maintained and the information path will remain D, A, B, C. This will continue until the first of the all zero words enters register D, when the condition will be D = 0, A > B. This status must also maintain the information path D, A, B, C, for two reasons; (i) the reduction of the information path to D, B, C at this stage would leave the last address of the sequence "stranded" in register A, (ii) the new information being fed into the store may have an address which is greater than any address already held in the memory, in which case the condition D = 0, A > B, exactly defines the correct insertion position for this address (the other possibility, that the new address is smaller than any of the other addresses held in the store, is covered by a special case, B = 0, of the D > A > B condition). On the next shift command the last address in the information block will be fed into register B, and register A is set to all zeros.

Neither the condition D > A > B or D = 0, A > B is now true and the information path reverts to D, B, C, D (shortening the store length by one word). Register D being all zeros causes the clock to remain connected to the store shift command until the information block has been shifted through the store and the first address appears in register D. Register A going to all zeros shows that the insertion process is at an end, and causes an interrupt to be sent to the microcomputer, which will eventually cause register A to be refilled with new information and the insertion process to be repeated. This cycling procedure continues until the store modifying process is complete (shown by a change in the OUT3 coding and the removal of the INSERT command).

The insertion of the very first address into a previously empty store is a special case, presenting its own problems. It is not possible to introduce the address word into the information path simply by testing for the condition D = B = C = 0, as this occurs naturally, even when address information is already present within the store. In order to overcome this problem the ERASE command is used to set a bistable. This bistable holds the information path, for an empty memory, at D, A, B, C... Thus the first address word fed into A is automatically passed into B. The change in status of B, from an all zero condition, is used to reset the bistable, so that after the insertion of this first address word the subsequent information is introduced to the refresh memory as described in the earlier part of this section.

2.4.2 The DELETE Operation

Having considered how an individual address is fed into the memory, it is appropriate to describe the technique employed to remove selective information. This is achieved using the DELETE command. As described in the previous section, the usual information path through the store is B, C, (main memory), D, B... when the DELETE command appears the normal read operation is halted and the clock is used to shift the information block through the store until the first item (the lowest address) is held in register D, any further shifting is inhibited so long as register A remains at all zeros. The item of data to be removed is then fed into register A. This activates the shift command and the information block is shifted through the store, still following the path B, C, D, B. However before each shift the contents of registers A and B are compared. When A = B this shows that the address held in B at that time is the address to be deleted from the information block. Register A being equal to B causes the information path to change from D, B, C, D to D, C, D (thus reducing the length of the store by one word), and the contents of D are also fed into A, register B is set to zero and the address that was held there is thus erased from the store. The information path D, C, D (D into A), continues until the first of the all zero words enters register A (it will simultaneously enter register C). Register A going to an all zero condition, together with a DELETE command and the status STP.(OUT3 = 004), causes an interrupt to be sent to the microprocessor, to show that the defined address has been deleted from the store. Register A equal to register B equal to all zeros also causes the information path to revert to the sequence B, C, D, B. The next address to be deleted is then fed into register A and the cycle is repeated until the OUT3 coding is changed and the DELETE command is removed. This shows that a particular modification sequence is at an end, and that all the addresses relating to a particular symbol have been deleted, while the correct order has been maintained for the addresses which are left to form the information block.

It should be noted that the action of the microcomputer is the same for both the INSERT operation and the DELETE operation. This action being the presentation of an address modifier (to be added to the base address) which is changed in response to an interrupt signal. Thus the same software may be used for both the INSERT and the DELETE operations, it is merely the presence of either an INSERT or a DELETE command which distinguishes between them.

2.4.3. The ERASE Operation

If it is required to cancel the whole display then it is unnecessary to carry out a separate DELETE operation for each symbol, the whole store may be cleared in one pass using the ERASE command. The presence of the ERASE signal causes the information path to change to A, B, C, D, A, B,....., and for the clock to be directly connected to the shift command of the store. In addition registers B and C are forced into an all zero condition, which is then shifted through the store until all registers have been cleared. No direct indication is given when the memory has been cleared and thus the ERASE command must be held for a time longer than the time taken to make one pass through the store. In the prototype unit, where the store length was sixty-eight words (during ERASE) and the clock shift rate was 2 MHz (see section 2.5), this time was 32 μs. For a larger store of, for example, 1,024 words, this time would increase to 0.512 ms.

The ERASE signal also sets a bistable which is used to enable the first word to be fed into a blank store (see section 2.4.1).

Thus an ERASE command should always be actioned immediately after the system is powered-up.

2.5 The Read Operation

Once the correct information has been fed into the refresh memory in the correct order, it is necessary to use the information to control the intensity of the raster. The information, as has already been explained, is stored in order of increasing address, and the raster sweeps across the screen from the lowest address to the highest. As the raster sweeps across the screen an 18-bit counter is synchronised with it, thus the status of this counter gives the address of the line section through which the electron beam of the monitor is passing at any particular instant (this counter is also used to derive the line and field synchronising pulses which control the raster scan).

Consider then the situation where the lowest order address is stored in register D, and the raster is commencing a sweep, starting at address zero. As the electron beam crosses each line section, the address of that section, derived from the 18-bit counter, is compared with the address held in register D. If the addresses are not the same no action is taken and the display remains unchanged. The raster continues its scan across the screen, the counter being incremented for each new section crossed, and each count being compared with the contents of register D. At some time the count will be equal to the address held in D. This equivalence causes two actions to be implemented simultaneously. First the status of the display is changed, to bright if it was dark, to dark if it was bright, whether this change is to be "temporary" or "permanent" is determined by the status of the flag bit held in D. Secondly the refresh memory is shifted once. Thus the contents of D are now in B, those of B are now in C, and so on. This means that the next highest order address held in the information block is now positioned in the D register. The comparison between the counter and the D register continues until equivalence between the two is again signalled. This again causes a change in display status and the next highest address word to be shifted into register D. This process continues until the first of the all zero words enters register D. The fact that the D register goes to an all zero state causes the clock pulse to be connected to the shift command of the refresh memory and the information block is shifted through the store until the lowest order address, that is the first of the block, occupies register D, at this point the shift pulses are disabled, leaving the store in the correct state for the commencement of the next frame.

In practice it is not the main clock signal, which runs at 4 MHz, and which is used to drive the 18-bit line/intra-line counter, which is applied to the shift command but a pulse train of half this frequency, derived from the counter unit. This is because the system was found to be more reliable if the lower clock rate was used. The use of a lower shift rate only becomes critical if a very long store is employed. Consider the worst case conditions under which the shifting of the information block, in order to be ready for the next frame, takes place. If only two sections on the screen are addressed, and these sections are section 1, the lowest addressable section, and section $2^{18}-1$, the highest addressable section, then these two addresses would be immediately adjacent to one another in the refresh store, all the rest of the memory being zeros. Thus after section

2^{18}-1 has been addressed, in order to be ready for the next frame, the store must be shifted through all its stages except one (since two positions are occupied) in the time taken up by the unused line scans (625 less 512) and the field blanking period. This allows approximately 3.4 ms for the shift operation. At a shift rate of 2 MHz this means that a maximum store length of 6,800 words may be used, quite sufficient for the applications envisaged for this type of V.D.U.

CHAPTER 3

SOFTWARE

3.1 The 8008 Microprocessor

The interpretation of the master computer signals and the presentation of the correct data to the display store is controlled by an Intel 8008 microprocessor. This device is described in detail in reference 6 but it is considered appropriate to outline its major characteristics at the start of this chapter to facilitate the understanding of the program desc~ibed later.

The 8008 uses an 8-bit word and has seven working registers, A, B, C, D, E, H and L. The memory is organised in the form of 377_8 word pages and the store location used in any memory reference instruction is defined by the contents of registers H and L; the contents of register H define the high order (page) address, the contents of register L define the low order (word) address.

In the prototype V.D.U. the 8008 is used in the form of the SIM-8 microcomputer (also described in reference 6). The extra circuitry of the SIM-8 allows the input of information from two 8-bit input ports (INPO and INP1) and the output of information at four output ports (OUTO, OUT1, OUT2, OUT3). All input and output operations take place via register A. Register A is also distinguished from the other working registers in that it has associated with it four flags

(Carrv, Sign, Parity and Zero), which mav be set in accordance with the status of register A. This allows for conditional jumps depending upon the value of a particular flag.

Interfacing between the microprocessor and the rest of the machine is achieved by the use of the STP (Stop) signal, which becomes true when the program enters a HLT (Halt) state, and by the use of the INT (Interrupt) line, which, when activated, causes the computer to leave the HLT state and execute the next instruction. Thus when the program is waiting for data from the master processor it enters the HLT state and sends out a STP signal. When the input data is ready a pulse on the INT line causes the next instruction, in this case an input instruction, to be obeyed. Similarly when the microprocessor has information which is to be used to modify the display store data it enters a HLT state and sends out a STP signal (it is this signal that shows that the data present on the output ports is to be processed). The machine then awaits an INT signal to show that the data has been processed before continuing with the next instruction.

The program consists of a master segment and five subroutines. A complete listing of the program is given in Appendix I. This listing was prepared using a cross-assembler program on a Digital Corp. PDP-8 machine. The program consists of $1,253_8$ words and thus utilises three 377_8 word P.R.O.M.s (type 1702A P.R.O.M.s were used). The flowchart of the program is contained within this chapter, and the rest of the text is concerned with program description.

3.2 The Master Segment

The program begins by setting output port OUT3 to the correct status. It will be remembered from section 2.3.1 that it is the status of OUT3 which determines the action taken by the hardware. The program then enters a HLT state, awaiting the input of the first word of the display sequence from the master processor. This first word contains information as to whether the symbol is vertical ($A_0 = 0$) or horizontal ($A_0 = 1$), and whether the symbol, if unidirectional, is (DOWN $A_1 = 0$), or UP ($A_1 = 1$). Bit 0 is shifted into the carry flag (A_1 being shifted into A_0) and a branch made depending upon the flag status. As the DOWN/UP test is made at a later stage this information is preserved by feeding it into register B (it may not be retained in register A as this is used for input and output operations).

Once the decision has been made as to whether the symbol is to be vertical or horizontal the program proceeds along one of two parallel paths. One path deals with the sorting of vertical symbols, the other with the sorting of horizontal symbols. The action of these two paths is the same, it is simply that the parts of the memory addressed by one path refer to display data for vertical components, those addressed by the second path refer to display data for horizontal components, thus only one path will be referred to in the text, the same explanations serving for both.

Let us suppose that the test on the first word shows that a vertical character is to be displayed. The next action is to load address registers H and L with the start address of the address modifiers required for the display of vertical leads. Subroutine SRT1 (described in detail in section 3.3) is then used to feed the modifiers to the display store, via output ports OUT1 and OUT2.

Following the processing of the lead data the program halts and awaits the second word of the display sequence, which defines the symbol type. The present program allows for six different symbol types, identified by codes 001 to 006, but this is easily extended to the limit of the 8 bit word, i.e. 377_8. A search is made for the individual symbol by comparing the contents of register A with that of register D. The D register is incremented by one and the comparison continued until its value is the same as that of the A register. As this search technique is used several times throughout the program, D is incremented by calling upon subroutine SRT2. When equivalence is determined between registers A and D the program loads the H and L registers with the memory location which is the start of the data modifications required for the display of the defined symbol. Subroutine SRT1 is again called upon to feed this information to the display store. When the display information has been processed the program is then ready to receive the first character word, defining the symbol magnitude. However in the case of short circuit symbols no magnitude data is required, and the program returns to the start position to await the first data word of the next symbol. Voltage and current sources present special problems because they are unidirectional. If the symbol coding is discovered to be either a current or voltage source, a test is made as to whether the symbol is to be an UP or a DOWN symbol. This is achieved by retrieving the UP/DOWN information from register B, feeding it into the carry flag position and testing the status of this flag. Depending upon the result of this test the H and L registers are loaded with the start address of the UP or DOWN component of the source symbol as appropriate. Much of the source symbol is common to all UP and DOWN, current and voltage sources, so, in order to reduce the symbol data storage requirement, only the parts of the symbol which distinguish it from the other source symbols are fed to the display store at this stage.

When these distinguishing characteristics have been absorbed into the display store the program then loads the start address of the data which is common to all voltage and current sources, and this is then presented to the display store (see figure 3.1).

The program then awaits the third word of the display sequence, which defines the first character of the display. It is necessary to permanently modify the base address at this stage (as was described in section 2.3.1). The magnitude of this first base address shift differs depending upon whether a horizontal or a vertical symbol is being displayed (see figure 3.2). A test is made to determine whether the first character is zero. If this is the case no action is taken, since to display this character would take up space in the display store without increasing the information presented. However a record of the first character being zero is kept in register B. If the first character is other than zero a search is made to determine which character is called for by incrementing register D and comparing it with the information held in register A until an equivalence is found. At this point the start address of the base address modifiers is loaded into the H and L registers and the modifications processed using SRT1. As a similar procedure is used for the second and third characters this action is implemented using a subroutine, SRT4. Following the processing of the first character modifiers the base address is further shifted in preparation for the second character. As the base address shift is the same for the third and fourth characters, independent of whether they refer to horizontal or vertical symbols, the shifting is incorporated in subroutine SRT5. If the first character is zero no shift of the base address takes place.

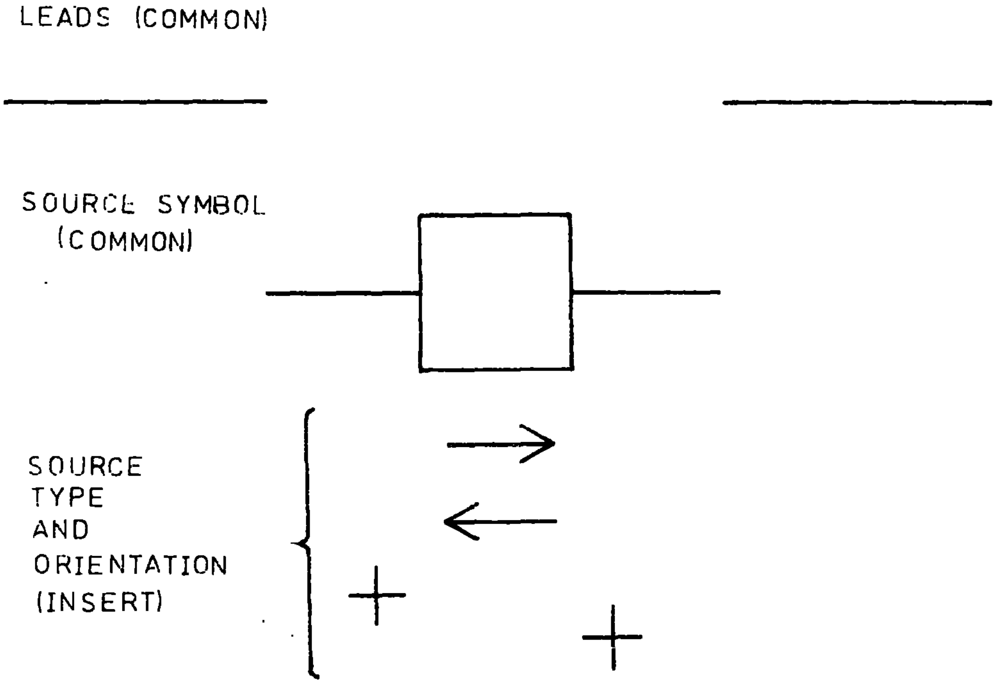


FIG 3.1 SOURCE ˉYMBOL STRUCTURE

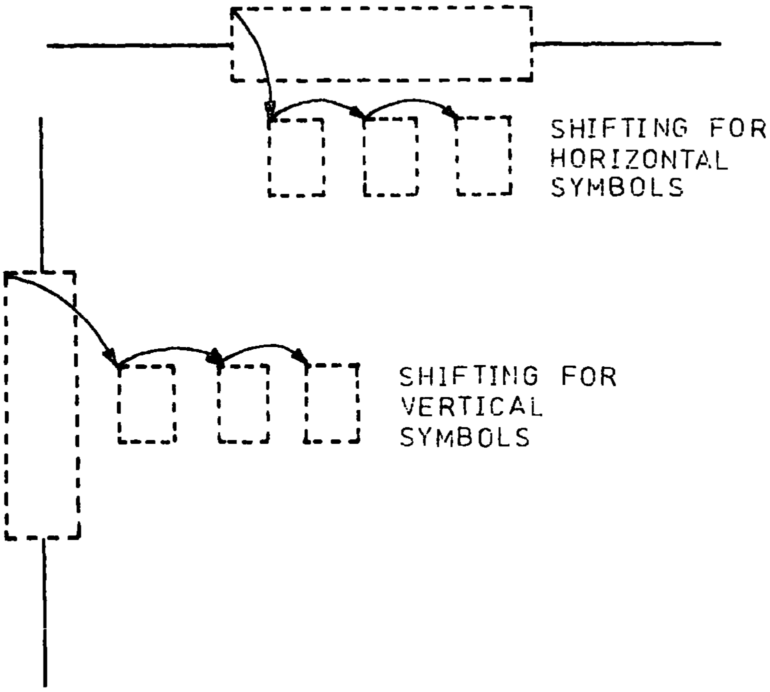

SHIFTING FOR HORIZONTAL SYMBOLS
SHIFTING FOR VERTICAL SYMBOLS

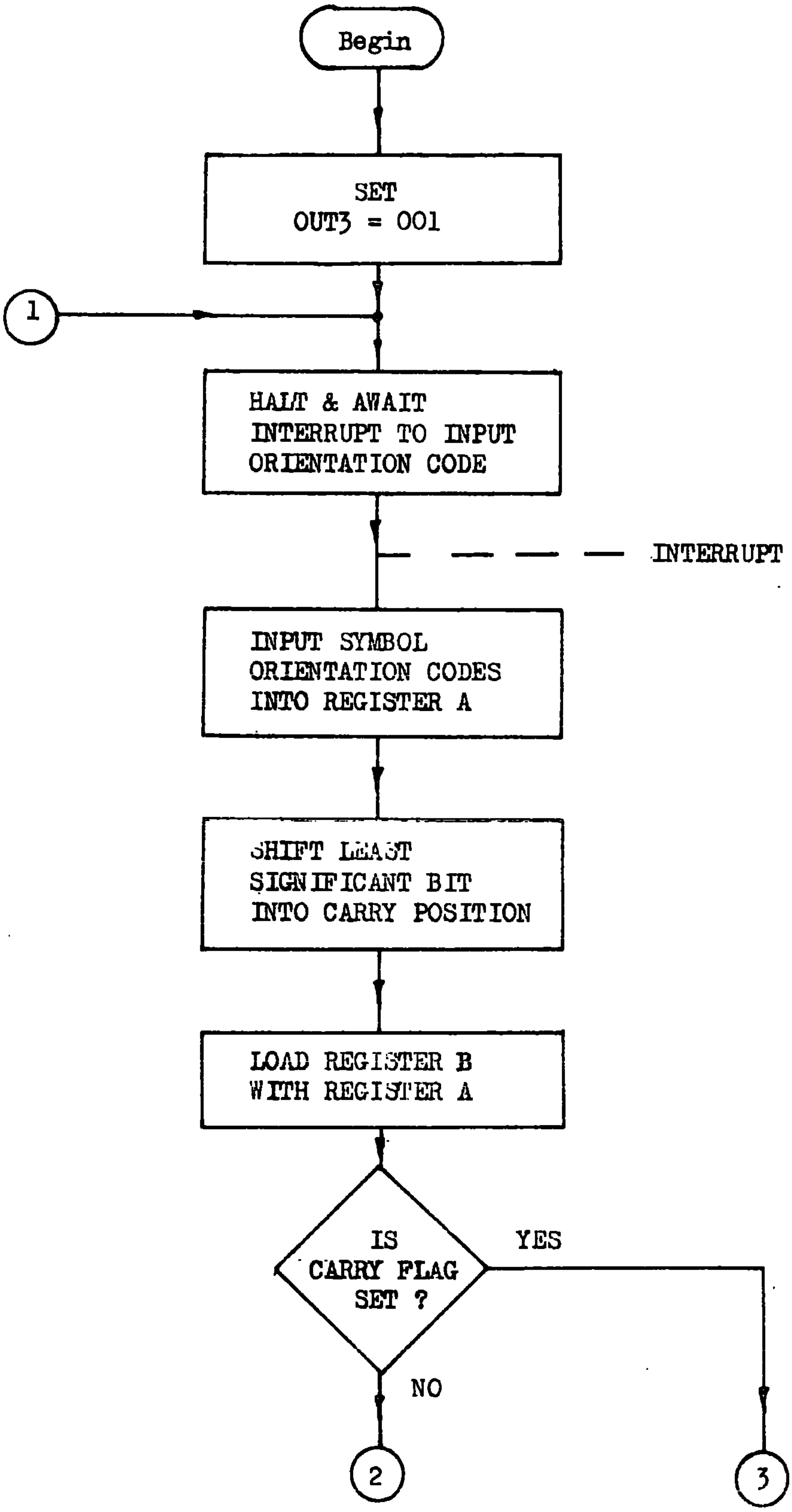
FIG 3.2 BASE ADDRESS COMPARISON

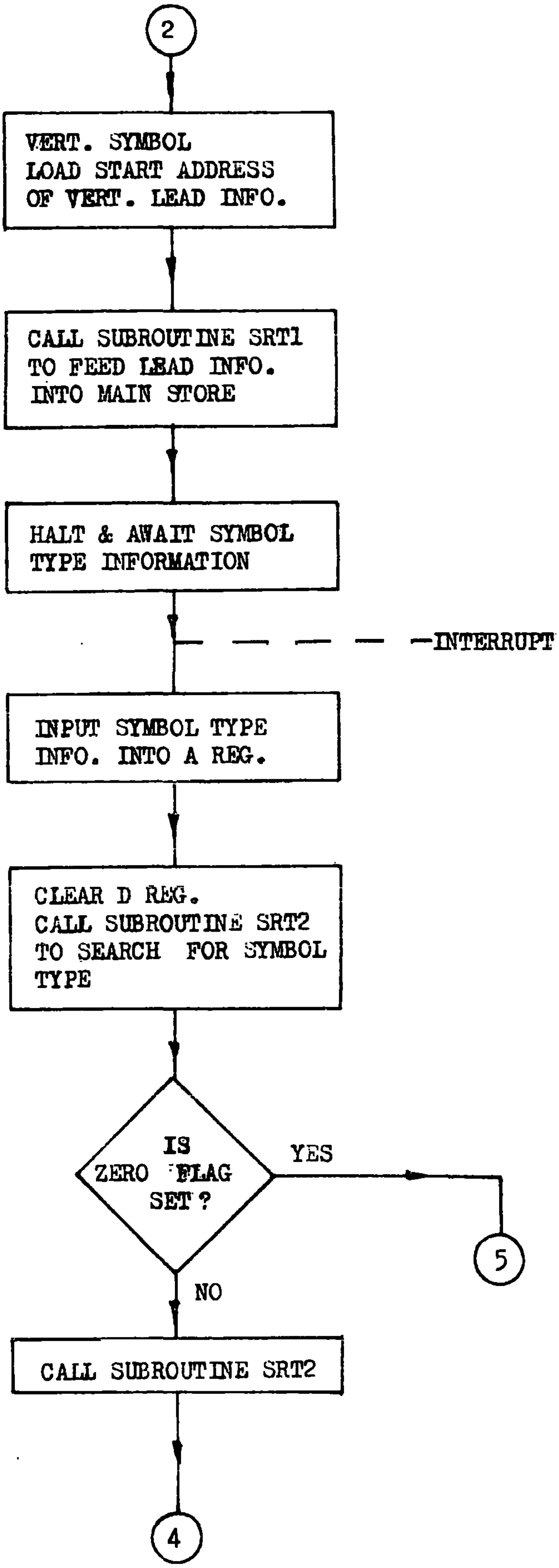
Again the microprocessor enters the HLT state and remains there until an INT signal causes the fourth word of the display sequence to be read into register A. This defines the second magnitude character and a test is made to determine whether this character is zero. If it is not zero the correct character is selected and the necessary data modifications made to the display store using SRT4 and SRT1. The base address is then shifted using SRT5. However if the second character is found to be a zero a second test is made to determine if the first character was also a zero. If the first character was other than zero then the second character must be displayed as a zero. The start address of the data modifiers for the zero character is loaded into registers H and L and the modifiers presented to the display store. SRT5 is then used to shift the base address in preparation for the next character. If however the first character was a zero then th. second zero is not displayed and there is no further modification to the base address.

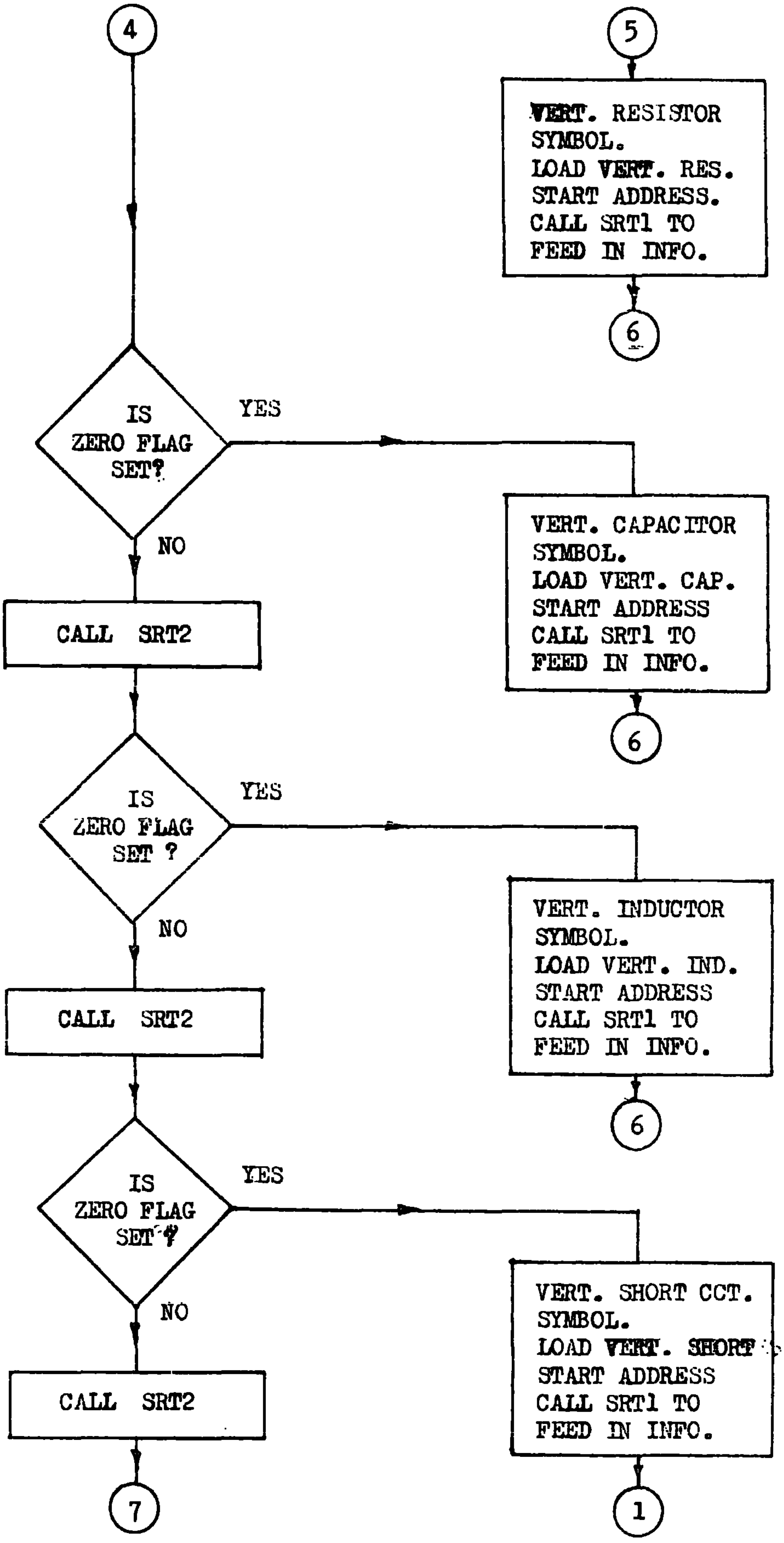
After reading in the fifth word of the display sequence, which refers to the third character, the program goes through the same sequence as described for the second character, with one exception. If the third character is a zero then this zero is always displayed on the screen, even if both previous characters were zeros. As the third character is always displayed, it is always necessary to shift the base address (SRT5) in preparation for the exponent character.

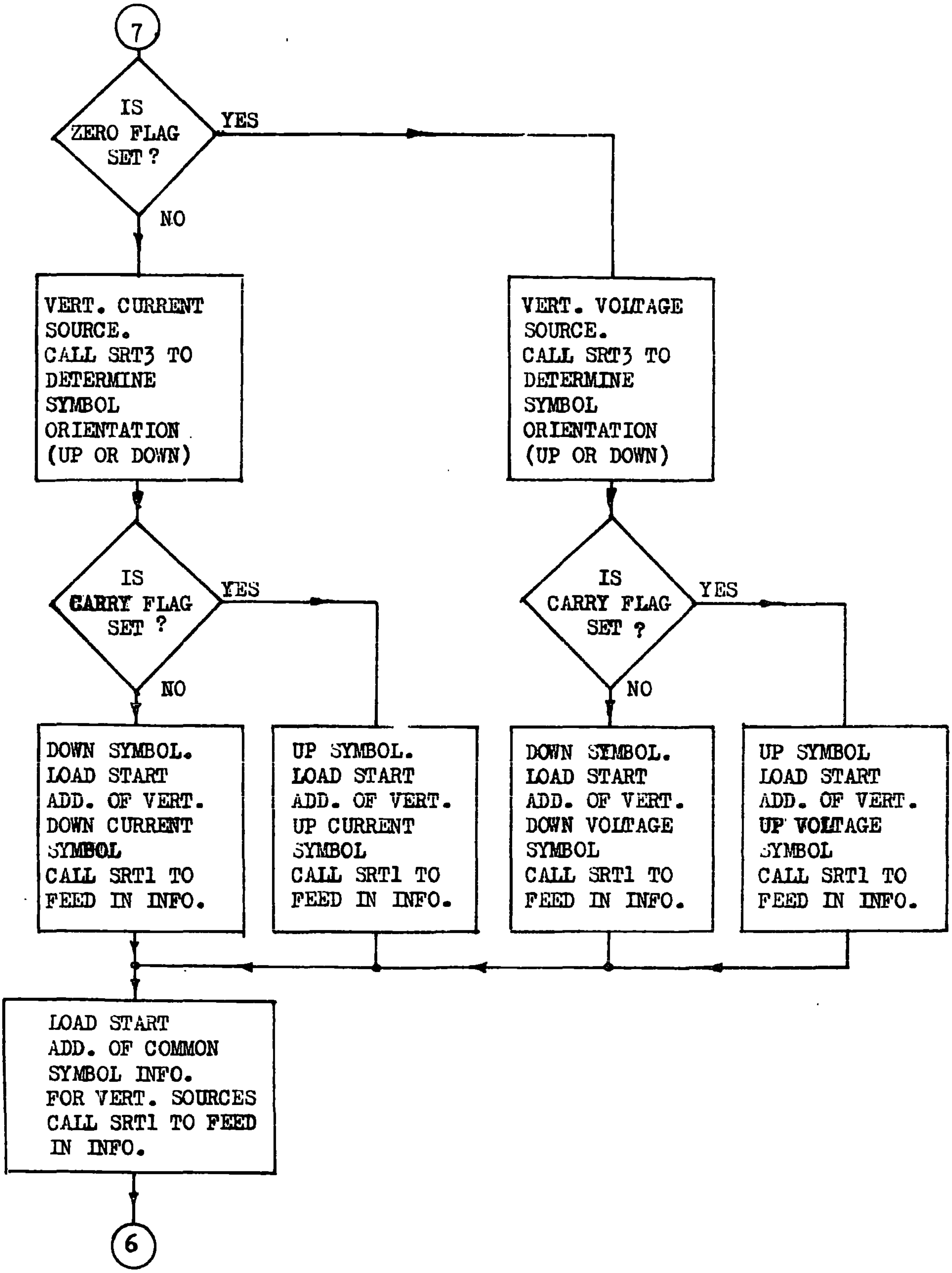
When the sixth word in the display sequence is read into register A a search for the correct exponent character is made in the same manner as was used to determine the correct numerals for the first, second and third characters. However as this search is carried out only once it is included in the main program, rather than as a subroutine. If the exponent character is a zero then the display remains unaltered and the program returns to the start to await the input of the first word of the next sequence. If the exponent character is other than zero the start address of the relevant data modifiers is loaded into registers H and L, and the modifiers fed to the output ports using subroutine SRT1. The program then returns to the start position.

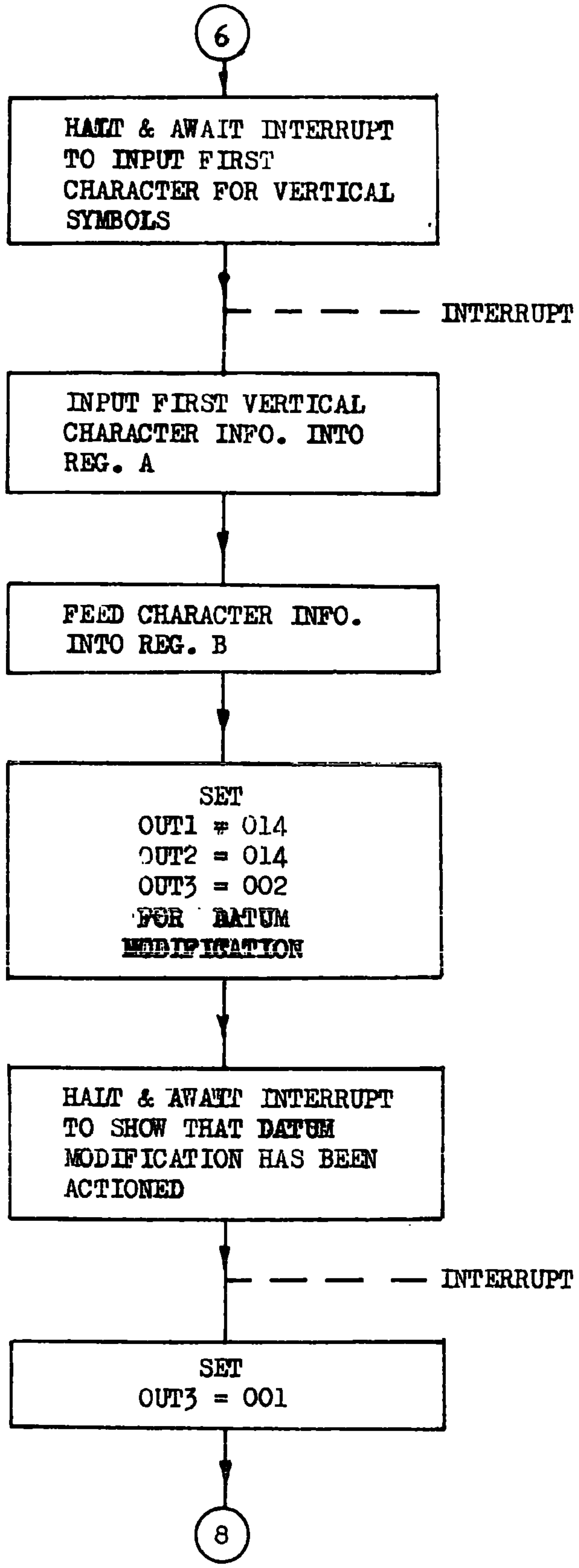
Master Segment

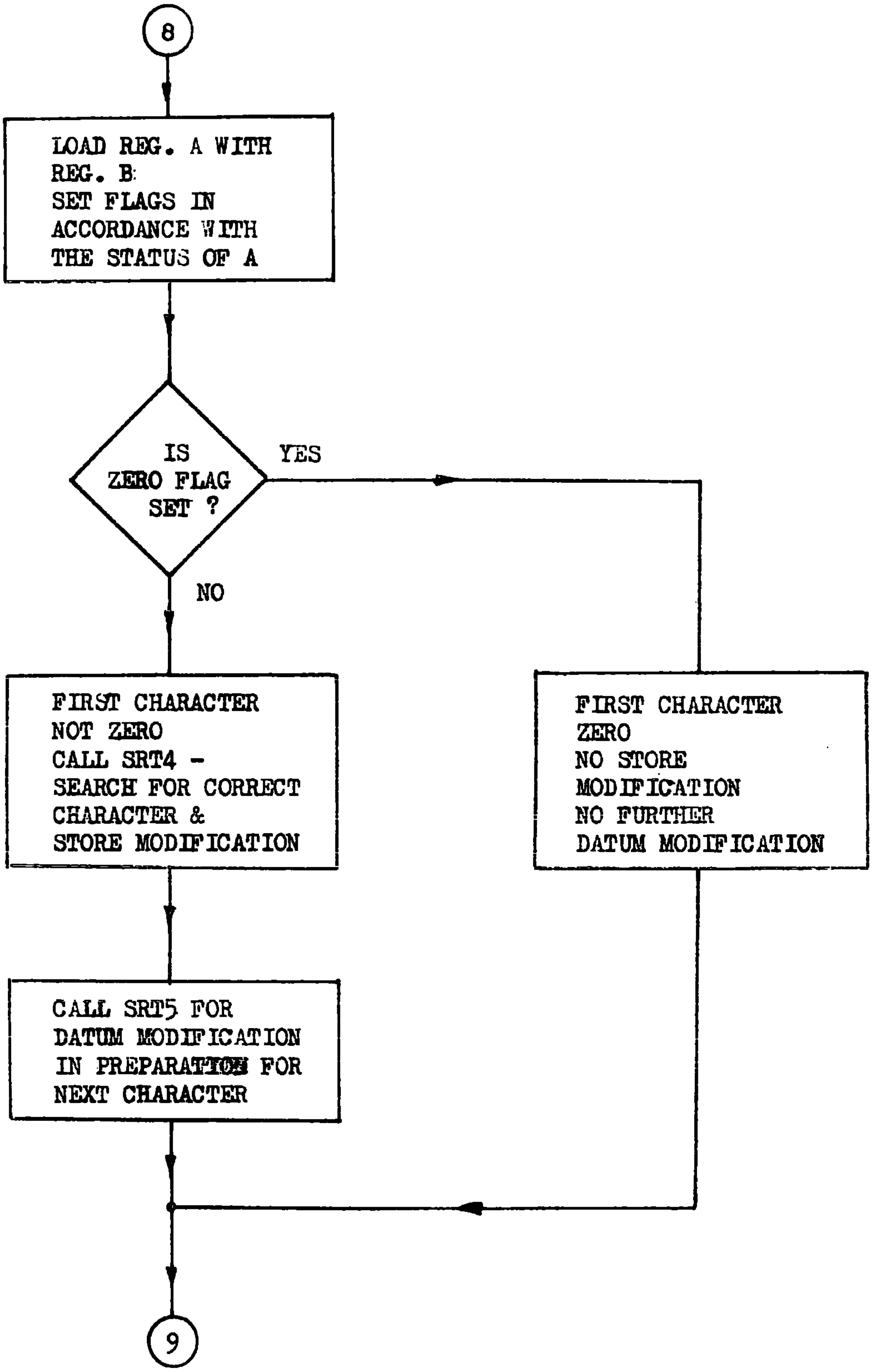


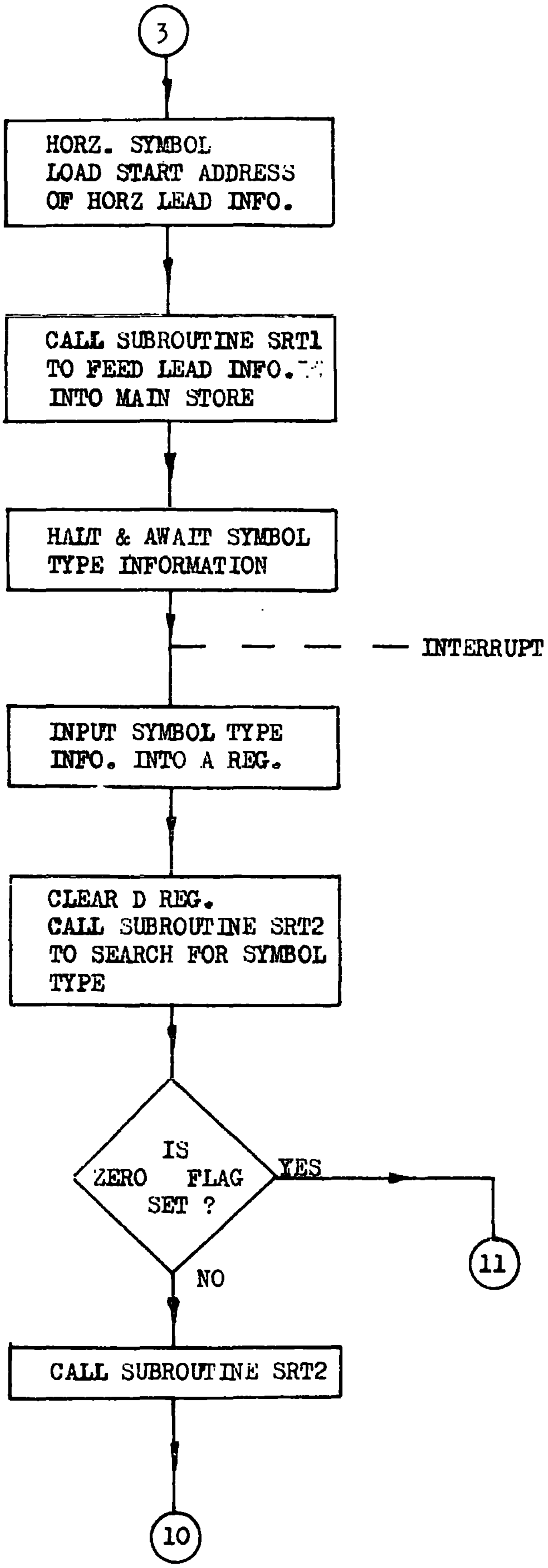

2
VERT. SYMBOL
LOAD START ADDRESS
OF VERT. LEAD INFO.
CALL SUBROUTINE SRT1
TO FEED LEAD INFO.
INTO MAIN STORE
HALT & AWAIT SYMBOL
TYPE INFORMATION
INTERRUPT
INPUT SYMBOL TYPE
INFO. INTO A REG.
CLEAR D REG.
CALL SUBROUTINE SRT2
TO SEARCH FOR SYMBOL
TYPE
IS
ZERO FLAG
SET?
YES
5
NO
CALL SUBROUTINE SRT2
4

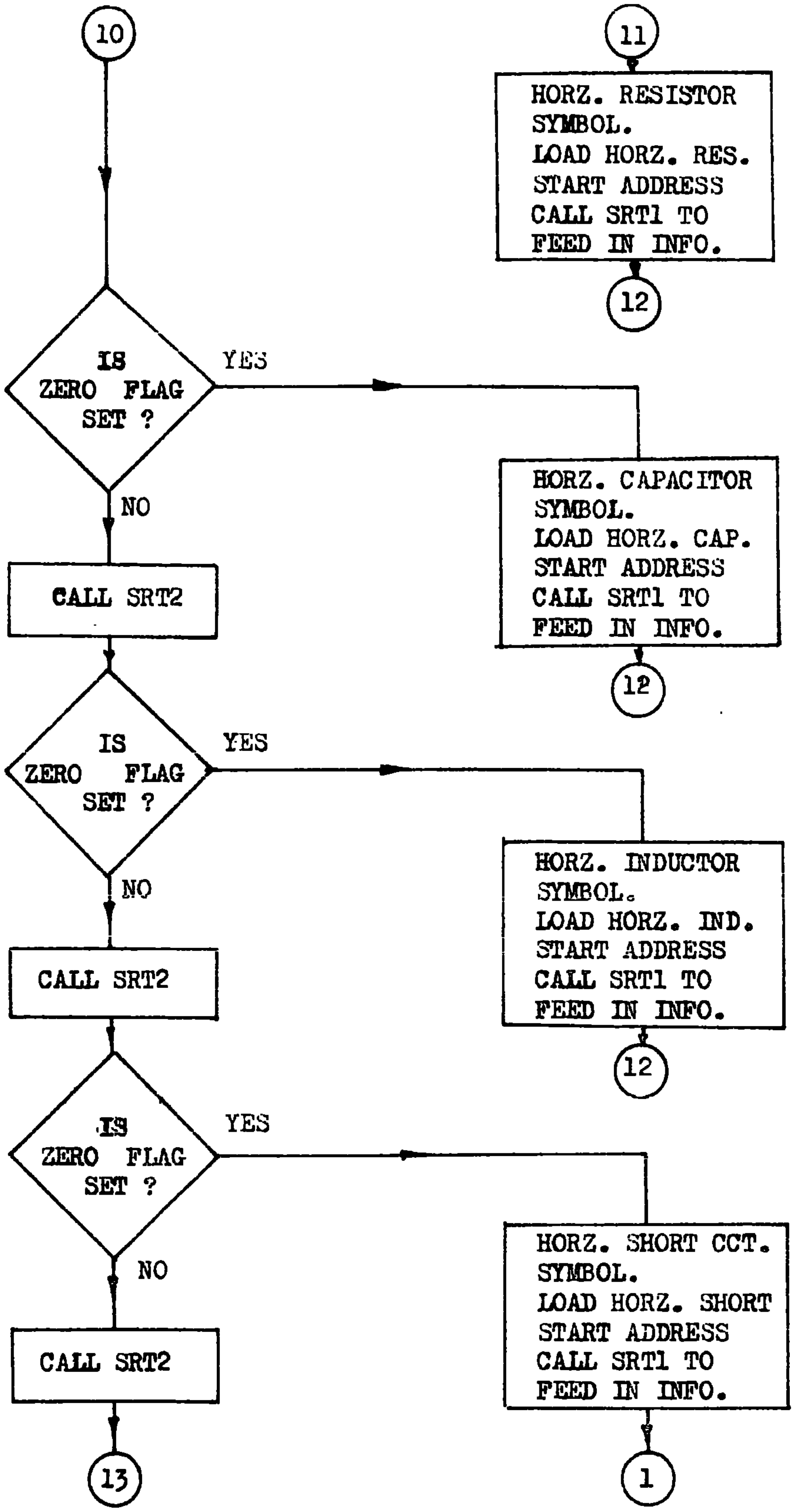

4
5
VERT. RESISTOR SYMBOL.
LOAD VERT. RES. START ADDRESS.
CALL SRT1 TO FEED IN INFO.
6
IS ZERO FLAG SET?
YES
NO
VERT. CAPACITOR SYMBOL.
LOAD VERT. CAP. START ADDRESS
CALL SRT1 TO FEED IN INFO.
CALL SRT2
6
IS ZERO FLAG SET ?
YES
NO
VERT. INDUCTOR SYMBOL.
LOAD VERT. IND. START ADDRESS
CALL SRT1 TO FEED IN INFO.
CALL SRT2
6
IS ZERO FLAG SET ?
YES
NO
VERT. SHORT CCT. SYMBOL.
LOAD VERT. SHORT START ADDRESS
CALL SRT1 TO FEED IN INFO.
CALL SRT2
7
1

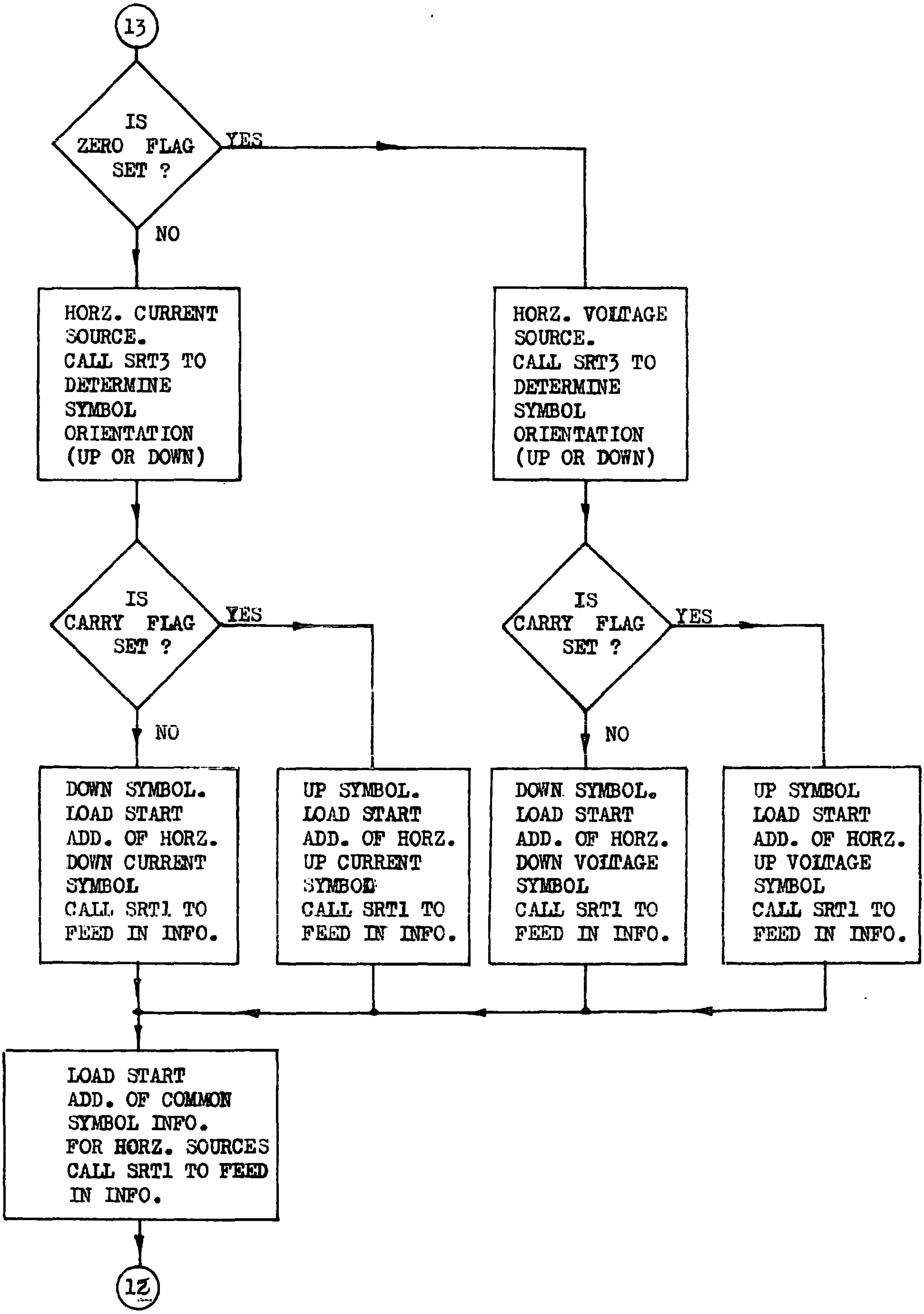

7
IS ZERO FLAG SET ?
YES
NO
VERT. CURRENT SOURCE. CALL SRT3 TO DETERMINE SYMBOL ORIENTATION (UP OR DOWN)
VERT. VOLTAGE SOURCE. CALL SRT3 TO DETERMINE SYMBOL ORIENTATION (UP OR DOWN)
IS CARRY FLAG SET ?
YES
NO
IS CARRY FLAG SET ?
YES
NO
DOWN SYMBOL. LOAD START ADD. OF VERT. DOWN CURRENT SYMBOL CALL SRT1 TO FEED IN INFO.
UP SYMBOL. LOAD START ADD. OF VERT. UP CURRENT SYMBOL CALL SRT1 TO FEED IN INFO.
DOWN SYMBOL. LOAD START ADD. OF VERT. DOWN VOLTAGE SYMBOL CALL SRT1 TO FEED IN INFO.
UP SYMBOL LOAD START ADD. OF VERT. UP VOLTAGE SYMBOL CALL SRT1 TO FEED IN INFO.
LOAD START ADD. OF COMMON SYMBOL INFO. FOR VERT. SOURCES CALL SRT1 TO FEED IN INFO.
6

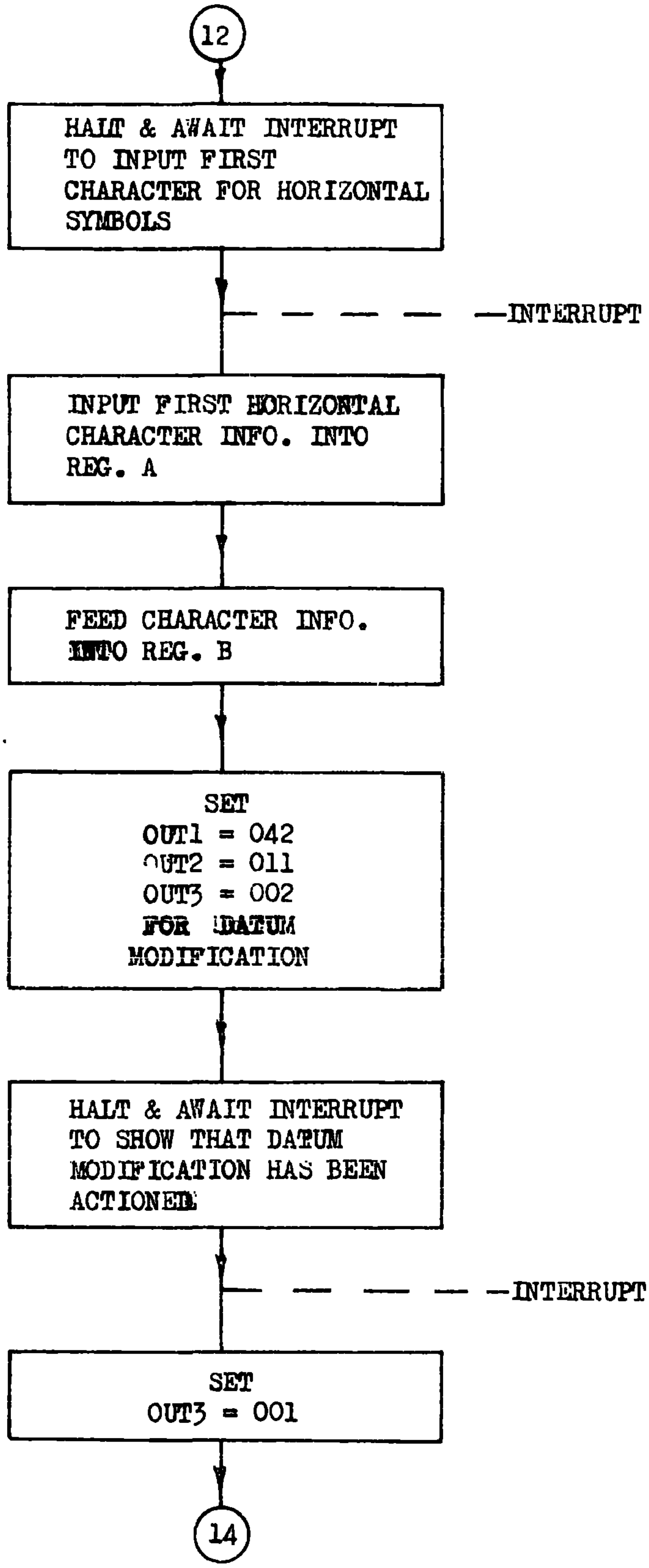

6
HALT & AWAIT INTERRUPT TO INPUT FIRST CHARACTER FOR VERTICAL SYMBOLS
INTERRUPT
INPUT FIRST VERTICAL CHARACTER INFO. INTO REG. A
FEED CHARACTER INFO. INTO REG. B
SET
OUT1 = 014
OUT2 = 014
OUT3 = 002
FOR DATUM MODIFICATION
HALT & AWAIT INTERRUPT TO SHOW THAT DATUM MODIFICATION HAS BEEN ACTIONED
INTERRUPT
SET
OUT3 = 001
8

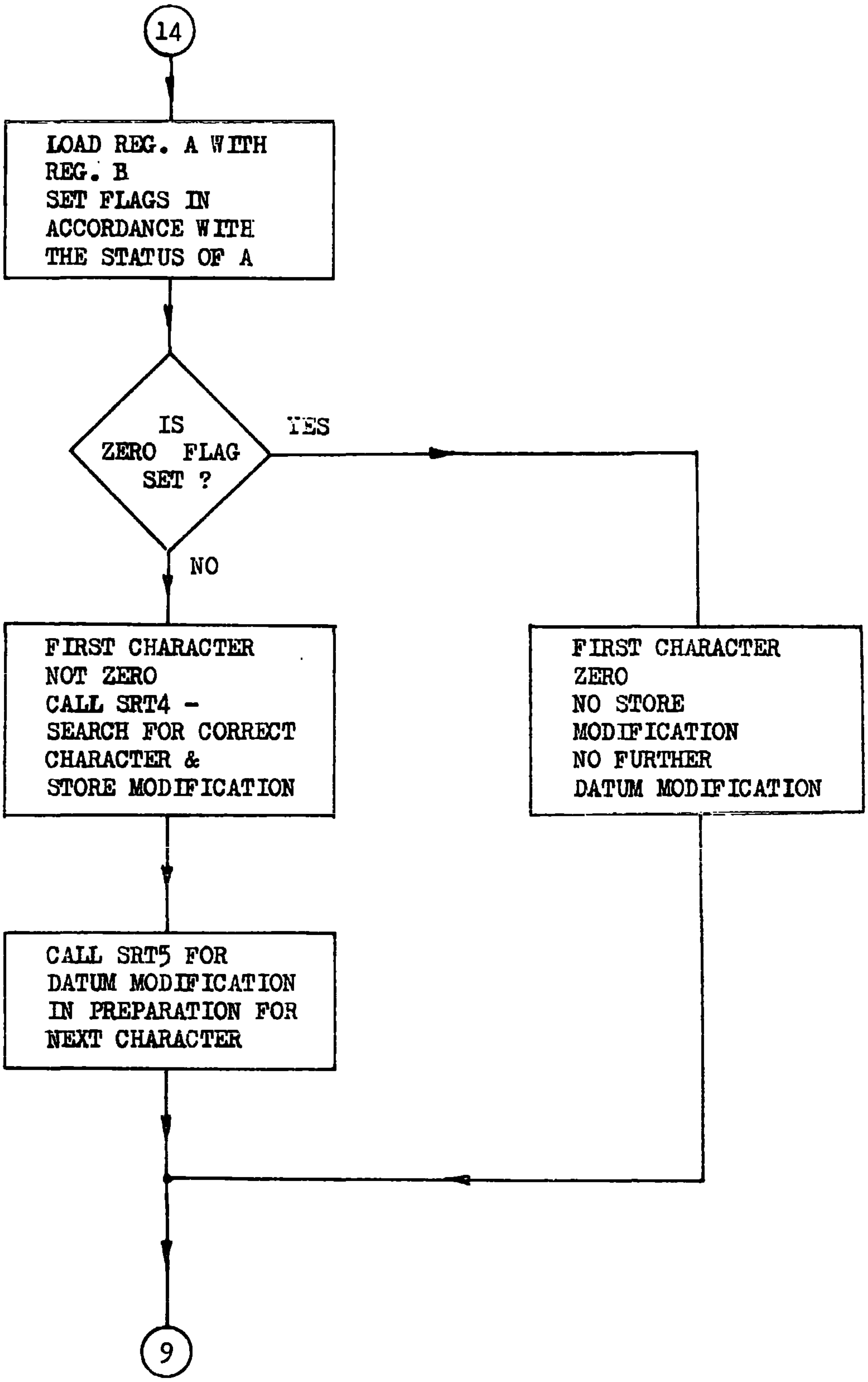

8
LOAD REG. A WITH REG. B
SET FLAGS IN ACCORDANCE WITH THE STATUS OF A
IS ZERO FLAG SET ?
YES
NO
FIRST CHARACTER NOT ZERO
CALL SRT4 - SEARCH FOR CORRECT CHARACTER & STORE MODIFICATION
FIRST CHARACTER ZERO
NO STORE MODIFICATION
NO FURTHER DATUM MODIFICATION
CALL SRT5 FOR DATUM MODIFICATION IN PREPARATION FOR NEXT CHARACTER
9

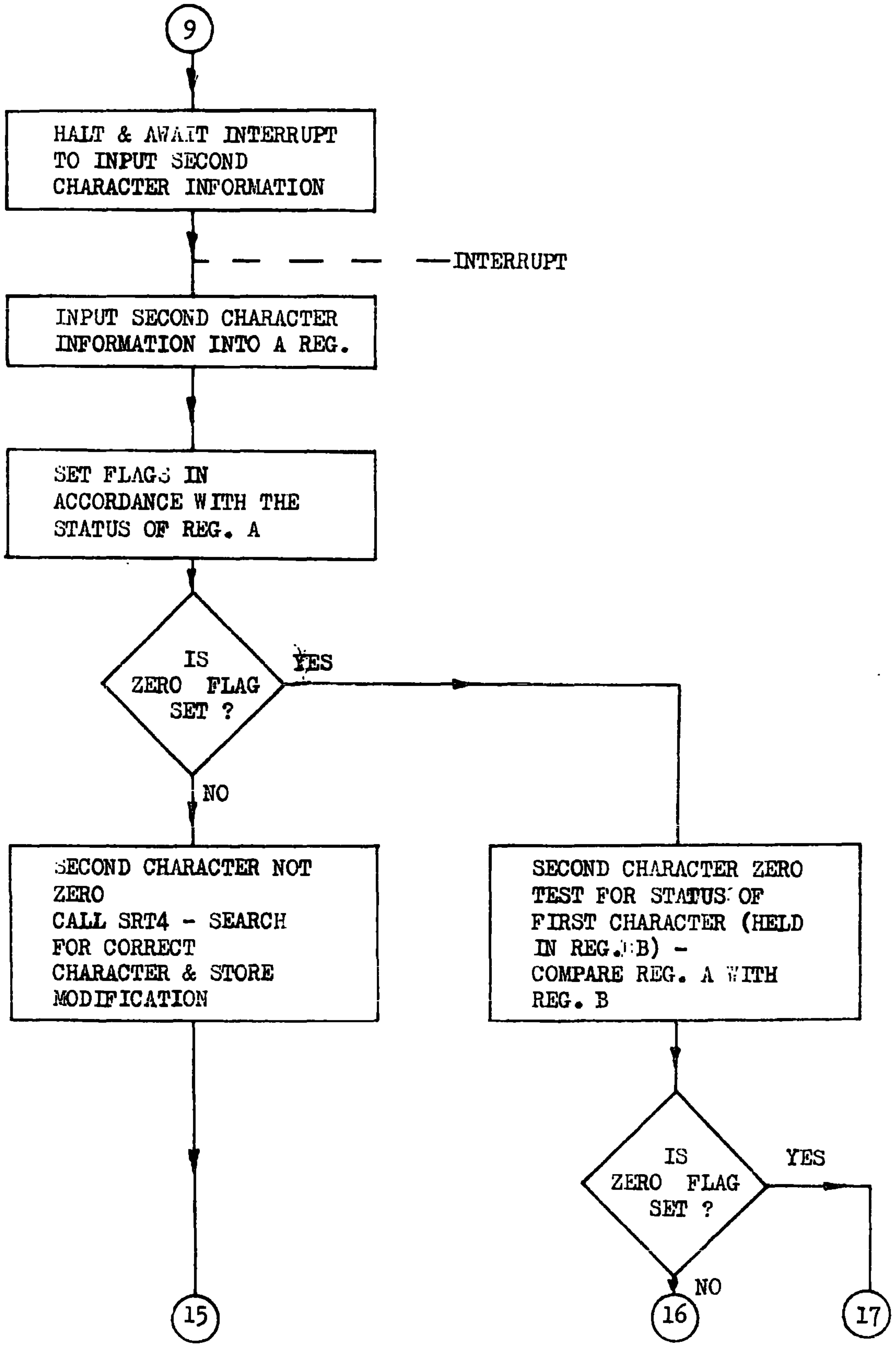

3
HORZ. SYMBOL LOAD START ADDRESS OF HORZ LEAD INFO.
CALL SUBROUTINE SRT1 TO FEED LEAD INFO. INTO MAIN STORE
HALT & AWAIT SYMBOL TYPE INFORMATION
INTERRUPT
INPUT SYMBOL TYPE INFO. INTO A REG.
CLEAR D REG. CALL SUBROUTINE SRT2 TO SEARCH FOR SYMBOL TYPE
IS ZERO FLAG SET ?
YES
11
NO
CALL SUBROUTINE SRT2
10

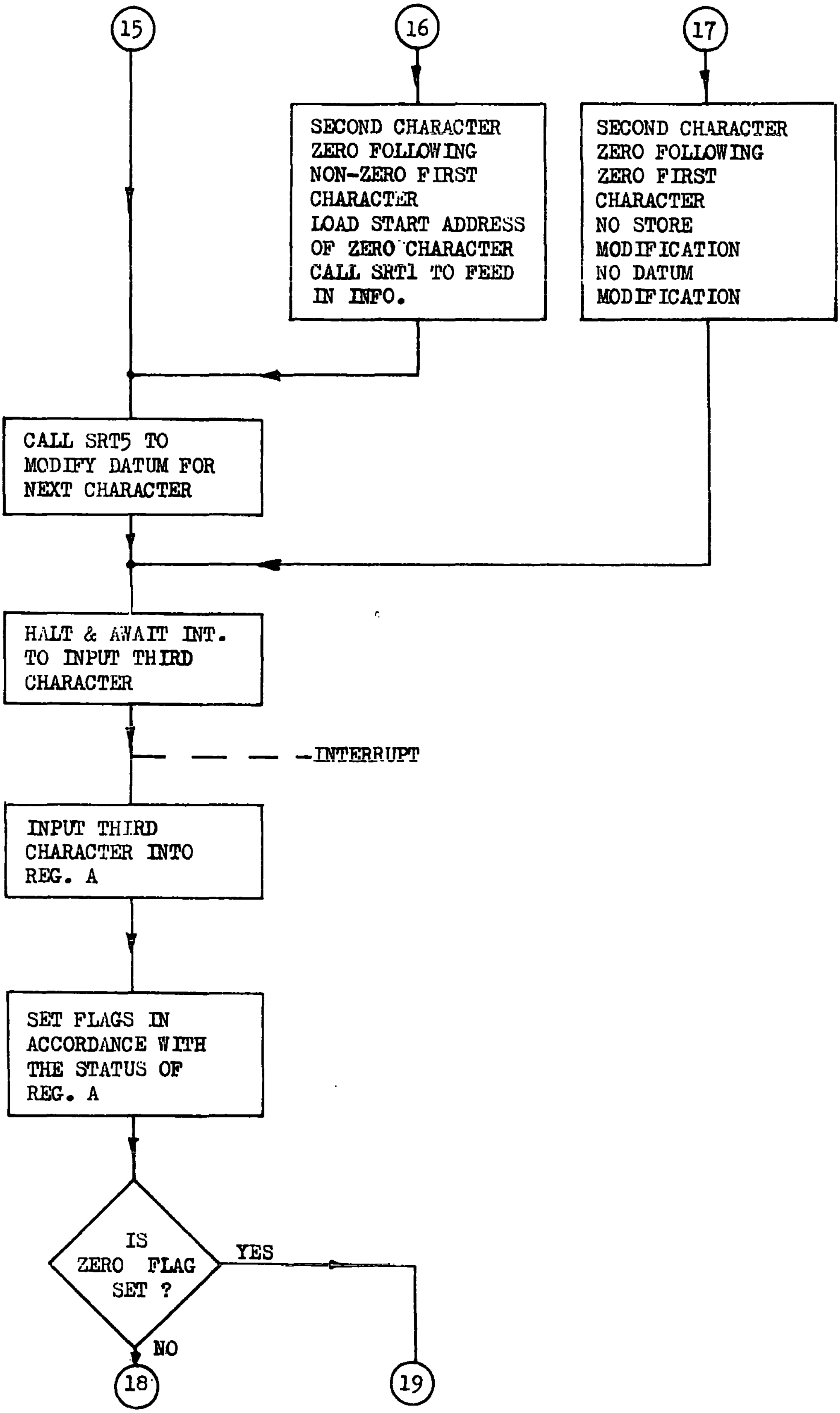

10
11
HORZ. RESISTOR SYMBOL. LOAD HORZ. RES. START ADDRESS CALL SRT1 TO FEED IN INFO.
12
IS ZERO FLAG SET ?
YES
NO
HORZ. CAPACITOR SYMBOL. LOAD HORZ. CAP. START ADDRESS CALL SRT1 TO FEED IN INFO.
CALL SRT2
12
IS ZERO FLAG SET ?
YES
NO
HORZ. INDUCTOR SYMBOL. LOAD HORZ. IND. START ADDRESS CALL SRT1 TO FEED IN INFO.
CALL SRT2
12
IS ZERO FLAG SET ?
YES
NO
HORZ. SHORT CCT. SYMBOL. LOAD HORZ. SHORT START ADDRESS CALL SRT1 TO FEED IN INFO.
CALL SRT2
13
1

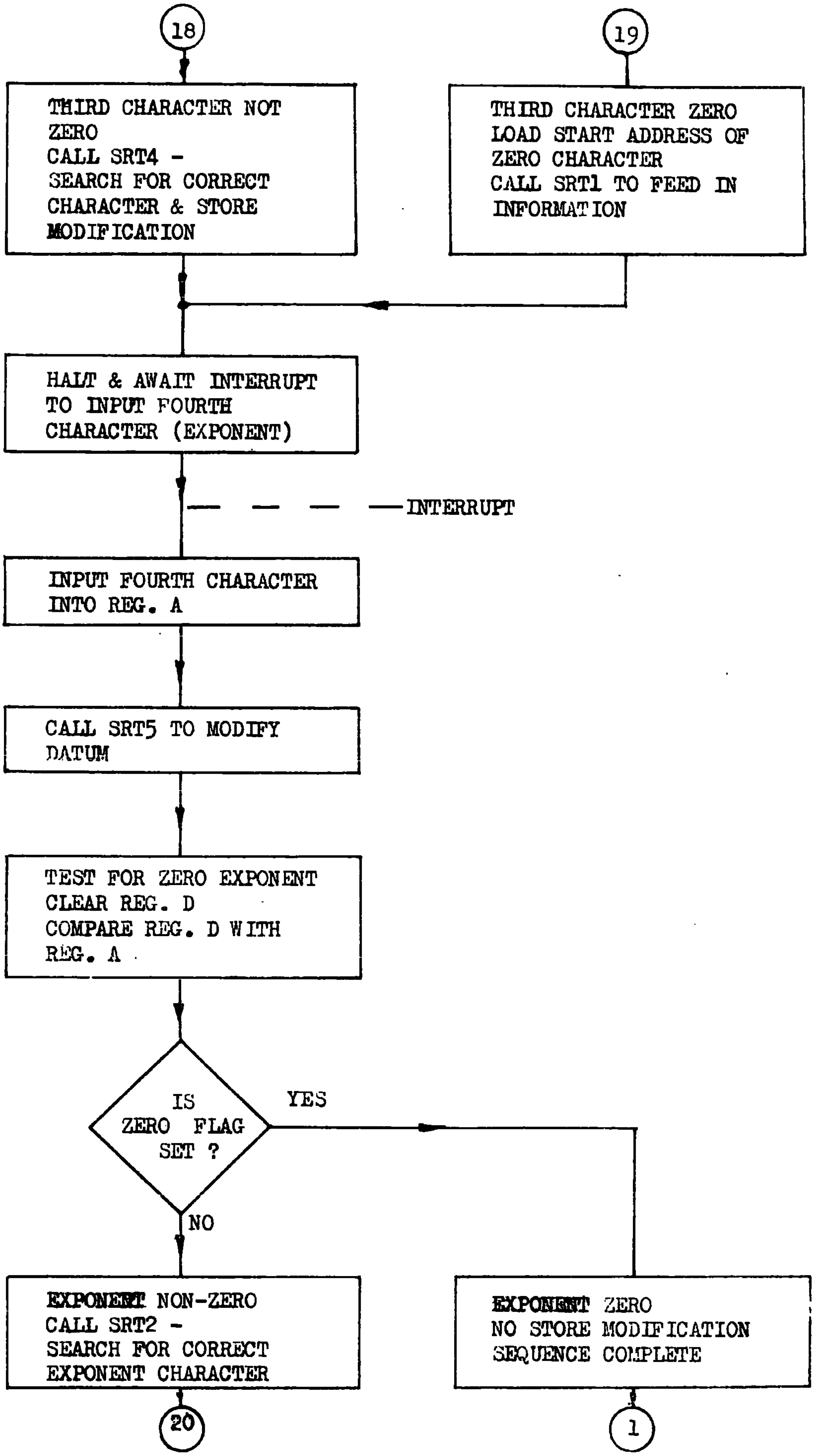

13
IS ZERO FLAG SET ?
YES
NO
HORZ. CURRENT SOURCE. CALL SRT3 TO DETERMINE SYMBOL ORIENTATION (UP OR DOWN)
HORZ. VOLTAGE SOURCE. CALL SRT3 TO DETERMINE SYMBOL ORIENTATION (UP OR DOWN)
IS CARRY FLAG SET ?
YES
NO
IS CARRY FLAG SET ?
YES
NO
DOWN SYMBOL. LOAD START ADD. OF HORZ. DOWN CURRENT SYMBOL CALL SRT1 TO FEED IN INFO.
UP SYMBOL. LOAD START ADD. OF HORZ. UP CURRENT SYMBOL CALL SRT1 TO FEED IN INFO.
DOWN SYMBOL. LOAD START ADD. OF HORZ. DOWN VOLTAGE SYMBOL CALL SRT1 TO FEED IN INFO.
UP SYMBOL LOAD START ADD. OF HORZ. UP VOLTAGE SYMBOL CALL SRT1 TO FEED IN INFO.
LOAD START ADD. OF COMMON SYMBOL INFO. FOR HORZ. SOURCES CALL SRT1 TO FEED IN INFO.
12

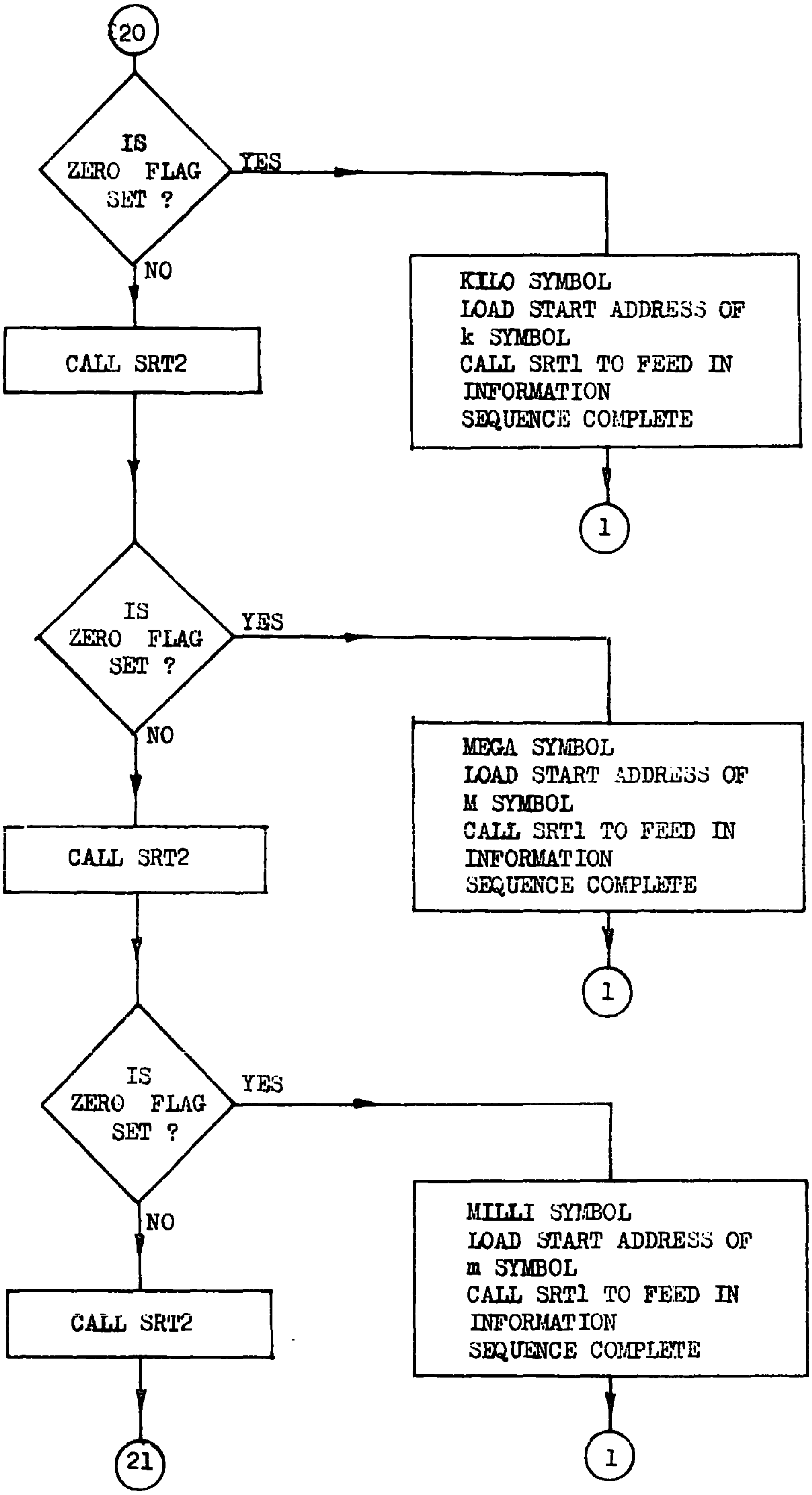

12
HALT & AWAIT INTERRUPT TO INPUT FIRST CHARACTER FOR HORIZONTAL SYMBOLS
INTERRUPT
INPUT FIRST HORIZONTAL CHARACTER INFO. INTO REG. A
FEED CHARACTER INFO. INTO REG. B
SET
OUT1 = 042
OUT2 = 011
OUT3 = 002
FOR DATUM MODIFICATION
HALT & AWAIT INTERRUPT TO SHOW THAT DATUM MODIFICATION HAS BEEN ACTIONED
INTERRUPT
SET
OUT3 = 001
14

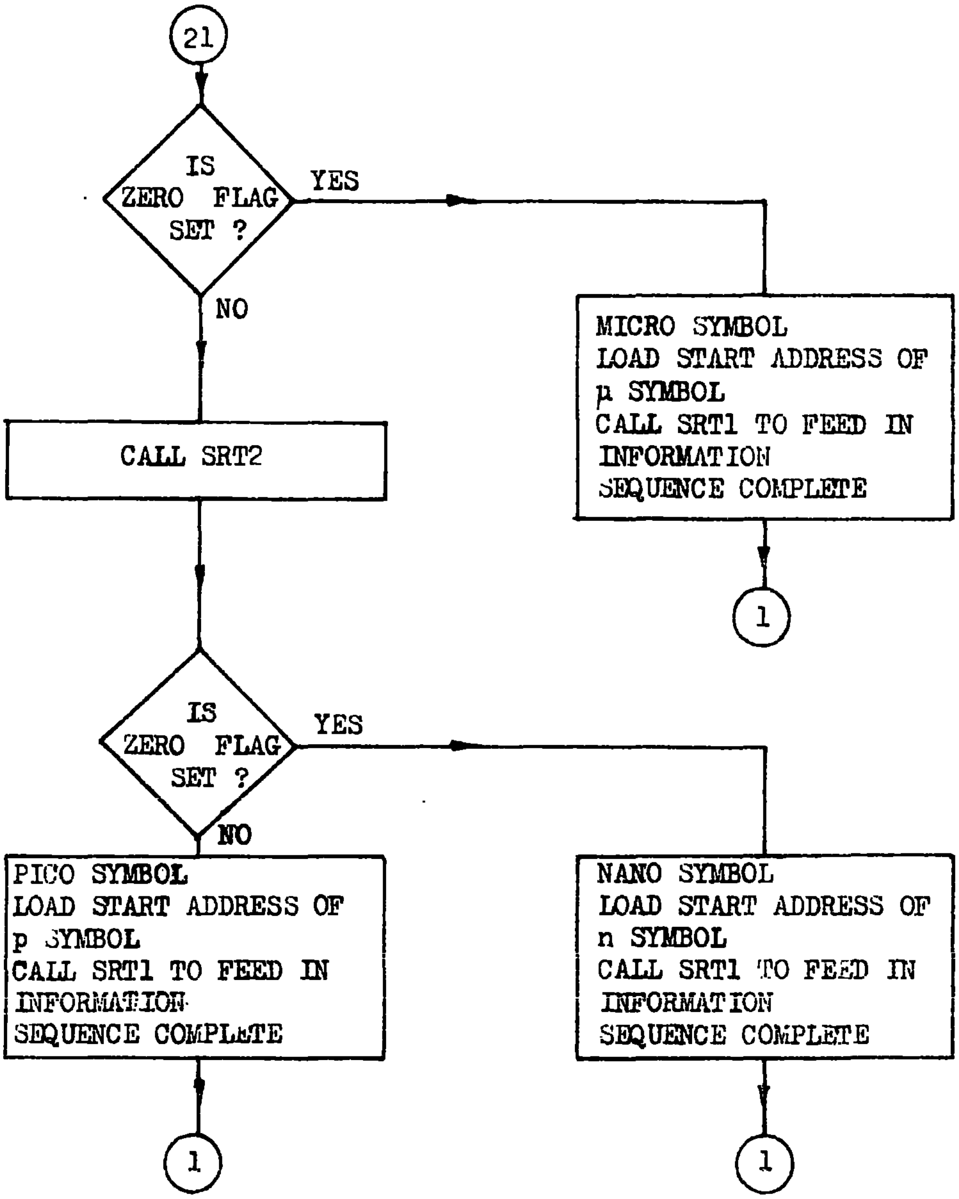

14
LOAD REG. A WITH
REG. B
SET FLAGS IN
ACCORDANCE WITH
THE STATUS OF A
IS
ZERO FLAG
SET ?
YES
NO
FIRST CHARACTER
NOT ZERO
CALL SRT4 -
SEARCH FOR CORRECT
CHARACTER &
STORE MODIFICATION
FIRST CHARACTER
ZERO
NO STORE
MODIFICATION
NO FURTHER
DATUM MODIFICATION
CALL SRT5 FOR
DATUM MODIFICATION
IN PREPARATION FOR
NEXT CHARACTER
9


9
HALT & AWAIT INTERRUPT TO INPUT SECOND CHARACTER INFORMATION
INTERRUPT
INPUT SECOND CHARACTER INFORMATION INTO A REG.
SET FLAGS IN ACCORDANCE WITH THE STATUS OF REG. A
IS ZERO FLAG SET ?
YES
NO
SECOND CHARACTER NOT ZERO
CALL SRT4 - SEARCH FOR CORRECT CHARACTER & STORE MODIFICATION
SECOND CHARACTER ZERO
TEST FOR STATUS OF FIRST CHARACTER (HELD IN REG. B) -
COMPARE REG. A WITH REG. B
IS ZERO FLAG SET ?
YES
NO
15
16
17


15
16
17
SECOND CHARACTER ZERO FOLLOWING NON-ZERO FIRST CHARACTER
LOAD START ADDRESS OF ZERO CHARACTER
CALL SRT1 TO FEED IN INFO.
SECOND CHARACTER ZERO FOLLOWING ZERO FIRST CHARACTER
NO STORE MODIFICATION
NO DATUM MODIFICATION
CALL SRT5 TO MODIFY DATUM FOR NEXT CHARACTER
HALT & AWAIT INT. TO INPUT THIRD CHARACTER
INTERRUPT
INPUT THIRD CHARACTER INTO REG. A
SET FLAGS IN ACCORDANCE WITH THE STATUS OF REG. A
IS ZERO FLAG SET ?
YES
NO
18
19


18
19
THIRD CHARACTER NOT ZERO
CALL SRT4 -
SEARCH FOR CORRECT CHARACTER & STORE MODIFICATION
THIRD CHARACTER ZERO
LOAD START ADDRESS OF ZERO CHARACTER
CALL SRT1 TO FEED IN INFORMATION
HALT & AWAIT INTERRUPT TO INPUT FOURTH CHARACTER (EXPONENT)
INTERRUPT
INPUT FOURTH CHARACTER INTO REG. A
CALL SRT5 TO MODIFY DATUM
TEST FOR ZERO EXPONENT
CLEAR REG. D
COMPARE REG. D WITH REG. A
IS ZERO FLAG SET ?
YES
NO
EXPONENT NON-ZERO
CALL SRT2 -
SEARCH FOR CORRECT EXPONENT CHARACTER
EXPONENT ZERO
NO STORE MODIFICATION
SEQUENCE COMPLETE
20
1


20
IS ZERO FLAG SET ?
YES
NO
CALL SRT2
KILO SYMBOL
LOAD START ADDRESS OF
k SYMBOL
CALL SRT1 TO FEED IN
INFORMATION
SEQUENCE COMPLETE
1
IS ZERO FLAG SET ?
YES
NO
CALL SRT2
MEGA SYMBOL
LOAD START ADDRESS OF
M SYMBOL
CALL SRT1 TO FEED IN
INFORMATION
SEQUENCE COMPLETE
1
IS ZERO FLAG SET ?
YES
NO
CALL SRT2
MILLI SYMBOL
LOAD START ADDRESS OF
m SYMBOL
CALL SRT1 TO FEED IN
INFORMATION
SEQUENCE COMPLETE
1
21


21
IS ZERO FLAG SET ?
YES
NO
MICRO SYMBOL
LOAD START ADDRESS OF
µ SYMBOL
CALL SRT1 TO FEED IN
INFORMATION
SEQUENCE COMPLETE
1
CALL SRT2
IS ZERO FLAG SET ?
YES
NO
PICO SYMBOL
LOAD START ADDRESS OF
p SYMBOL
CALL SRT1 TO FEED IN
INFORMATION
SEQUENCE COMPLETE
1
NANO SYMBOL
LOAD START ADDRESS OF
n SYMBOL
CALL SRT1 TO FEED IN
INFORMATION
SEQUENCE COMPLETE
1

3.3 Subroutine SRT1

It is the purpose of this subroutine to retrieve the necessary base address modifiers from the symbol store and feed them to the display store via OUT1 and OUT2, the data being changed in response to an INT signal. The process continues until all the modifiers relating to a particular symbol, or character, have been absorbed, when control is returned to the master segment.

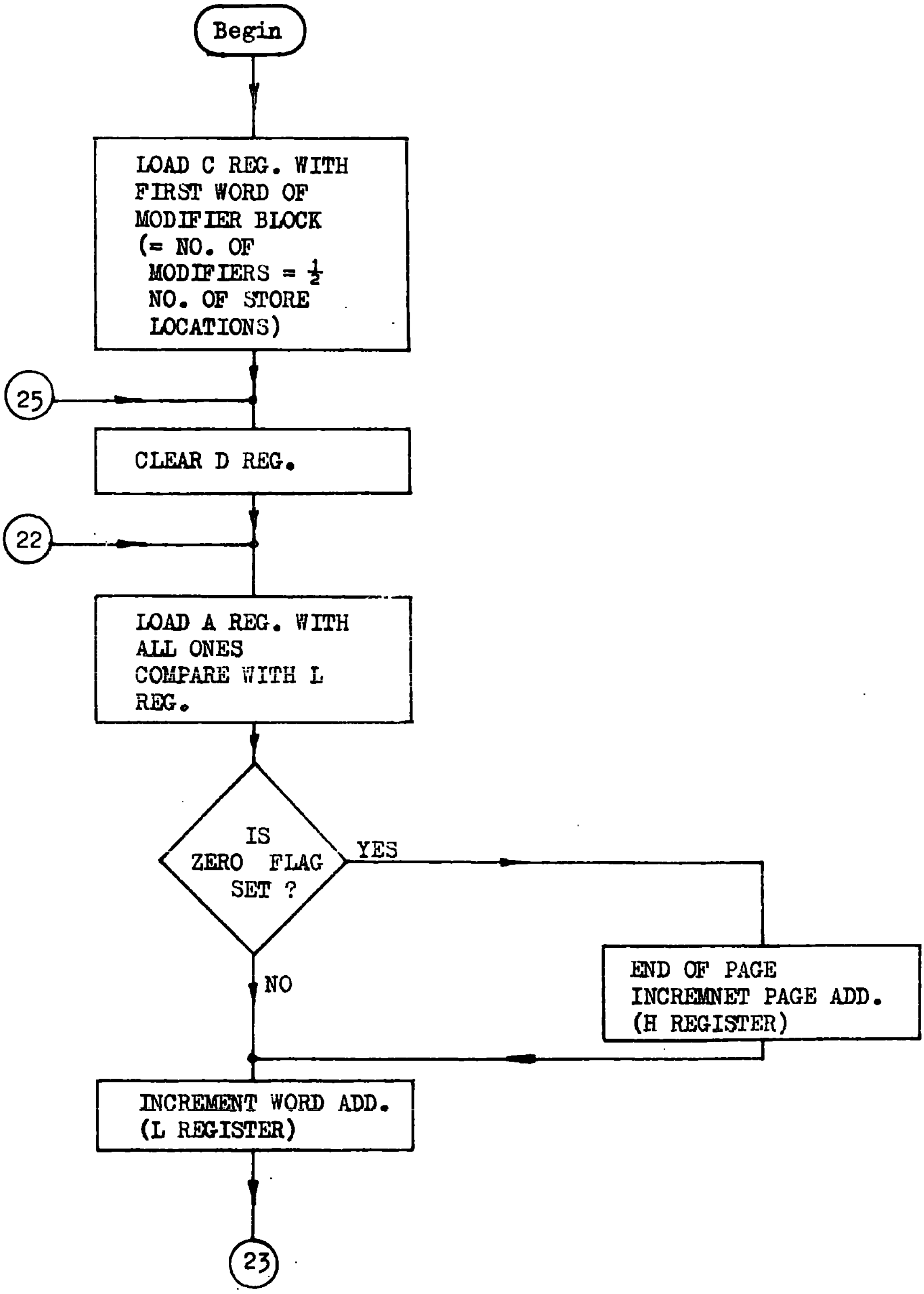
The first action is to load the C register with the contents of the memory location referred to by the H and L registers, the status of H and L being determined by the selection procedures outlined in the discussion of the master segment, and being set to the code which is the start address of the modifier block relating to the symbol to be displayed. The first location in the modifier block (which is fed into register C) is not itself a modifier but relates to the number of modifiers in that block. This information is necessary since the number of modifiers varies greatly from block to block, e.g. 2_8 for a horizontal short circuit symbol, 36_8 for a horizontal capacitor. It will be recalled that each modifier consists of two parts, a line modification, fed to output port OUT2, and an intra-line modification, fed to output port OUT1. Thus the figure now held in register C, which refers to the number of modifiers in the block, is, in fact, equal to half the number of store locations in the block. After each modifier has been processed register C is decremented, thus the modification sequence is at an end when register C becomes equal to zero.

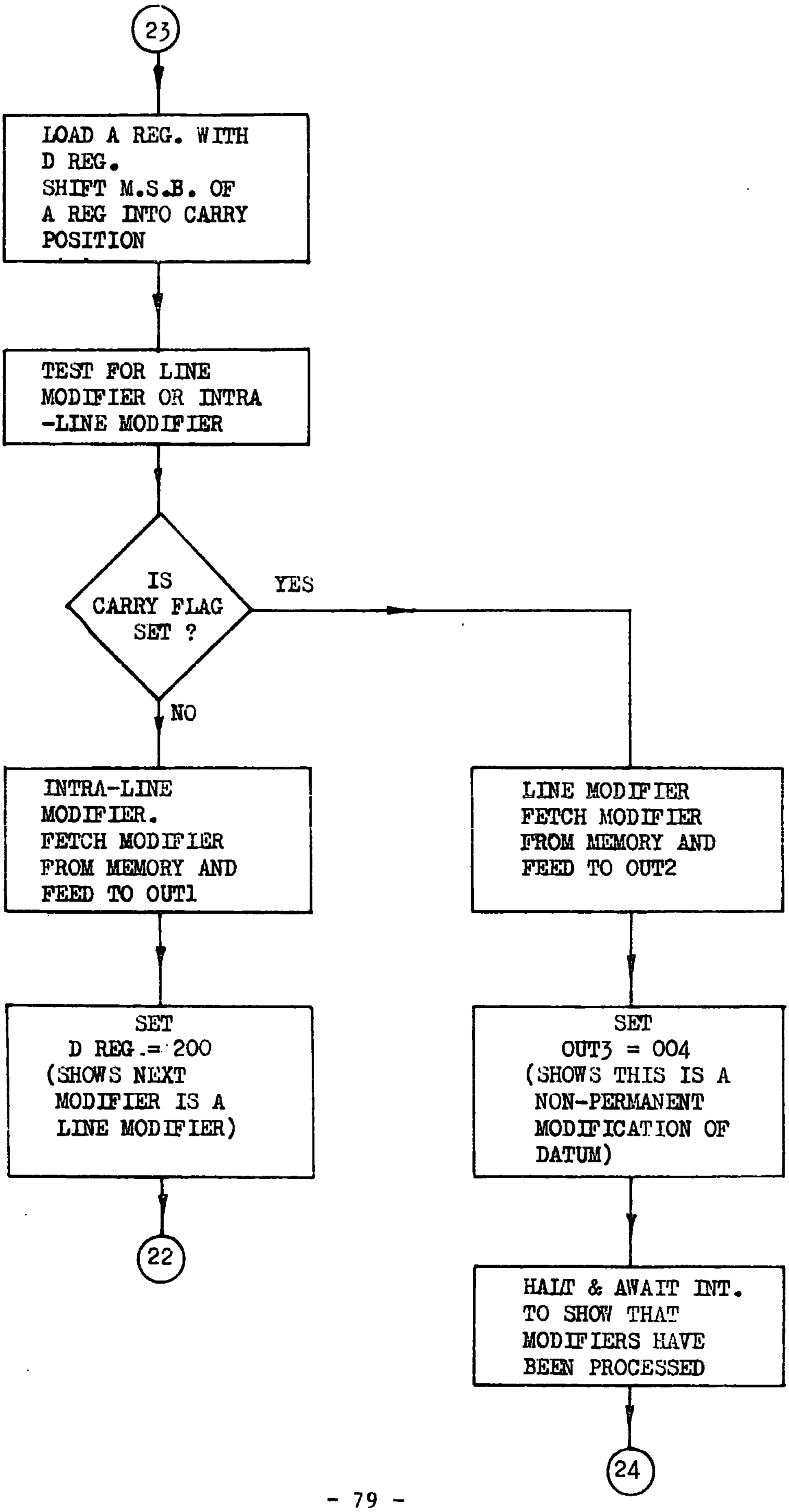
Initially register D is set to zero. It is the most significant bit (D_7) of this register that records whether the information retrieved from the modifier block is a line modifier

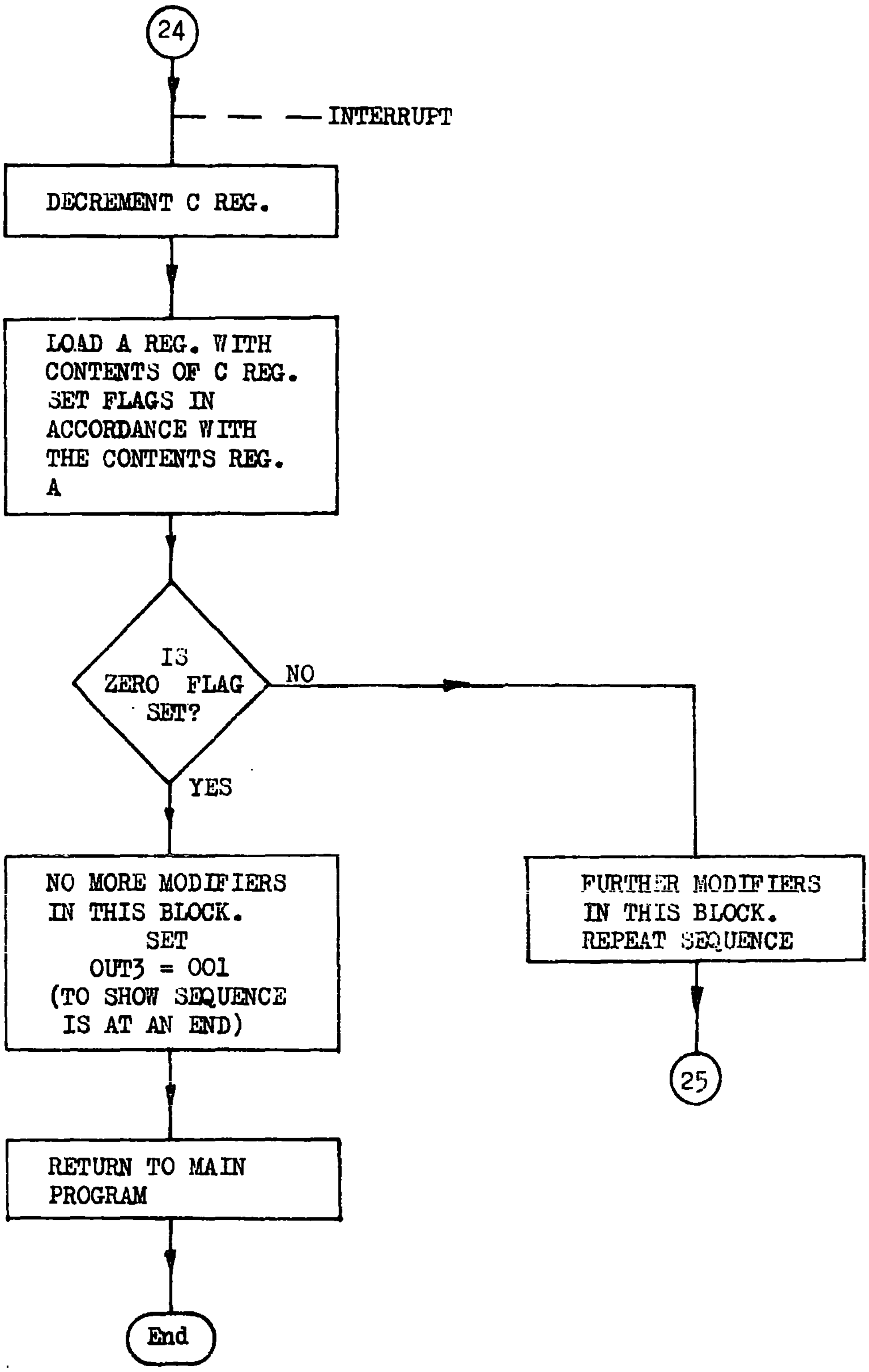
(D_7 = 1), or an intra-line modifier (D_7 = 0). A test is made to determine if the L register is all ones. If it is then the next location to be accessed is on the next page and the page address (register H) is incremented. Following this the L register itself is incremented. The status of D_7 is tested to determine whether a line modifier is to be processed next, or an intra-line modifier. If it is to be an intra-line modifier the correct data is retrieved from the symbol store and fed to output port OUT1. D_7 is set to a one to show that the next modifier will be a line modifier. If on testing D_7 is already a one then the retrieved information is fed to output port OUT2. OUT3 is then set to 004 and the program brought to a halt. The STP signal, together with the OUT3 coding, cause the OUT1, OUT2 data to be processed by the display store hardware.

When the data has been processed an INT signal is received and the program continues. Register C is decremented and tested for zero status. If register C is not zero then the sequence described above is repeated, using the next data in the modifier block. If, however, register C is set at zero then all the modifiers in this block have been processed. The coding of OUT3 is changed, to show that this sequence is at an end, and control is returned to the master segment.

Subroutine SRT1


Begin
LOAD C REG. WITH FIRST WORD OF MODIFIER BLOCK (= NO. OF MODIFIERS = ½ NO. OF STORE LOCATIONS)
25
CLEAR D REG.
22
LOAD A REG. WITH ALL ONES COMPARE WITH L REG.
IS ZERO FLAG SET ?
YES
END OF PAGE INCREMNET PAGE ADD. (H REGISTER)
NO
INCREMENT WORD ADD. (L REGISTER)
23

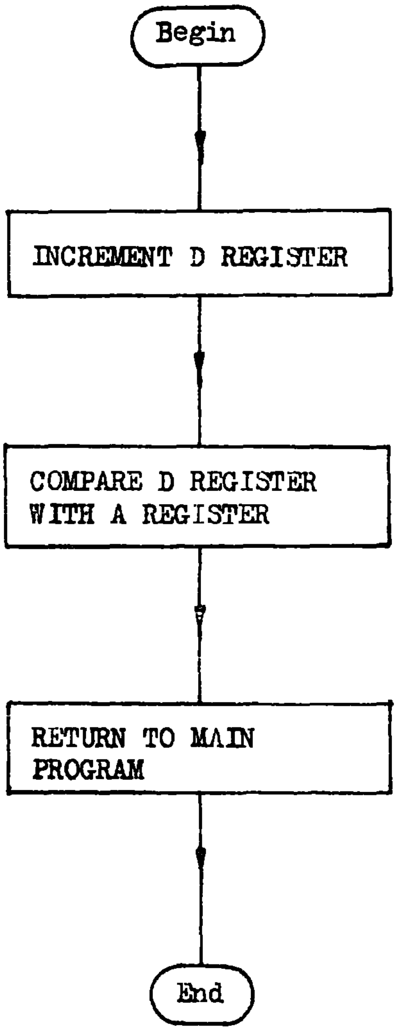

23
LOAD A REG. WITH D REG. SHIFT M.S.B. OF A REG INTO CARRY POSITION
TEST FOR LINE MODIFIER OR INTRA -LINE MODIFIER
IS CARRY FLAG SET ?
YES
NO
INTRA-LINE MODIFIER. FETCH MODIFIER FROM MEMORY AND FEED TO OUT1
LINE MODIFIER FETCH MODIFIER FROM MEMORY AND FEED TO OUT2
SET D REG.= 200 (SHOWS NEXT MODIFIER IS A LINE MODIFIER)
SET OUT3 = 004 (SHOWS THIS IS A NON-PERMANENT MODIFICATION OF DATUM)
22
HALT & AWAIT INT. TO SHOW THAT MODIFIERS HAVE BEEN PROCESSED
24


24
INTERRUPT
DECREMENT C REG.
LOAD A REG. WITH CONTENTS OF C REG. SET FLAGS IN ACCORDANCE WITH THE CONTENTS REG. A
IS ZERO FLAG SET?
NO
YES
NO MORE MODIFIERS IN THIS BLOCK. SET OUT3 = 001 (TO SHOW SEQUENCE IS AT AN END)
FURTHER MODIFIERS IN THIS BLOCK. REPEAT SEQUENCE
25
RETURN TO MAIN PROGRAM
End

3.4 Subroutine SRT2

This is a simple subroutine, but one which is widely used when searching for symbol and character types. Register D is incremented and then compared with register A. This comparison has the effect of setting the zero flag if A = D, thus when control is returned to the master segment the value of register A may be determined.

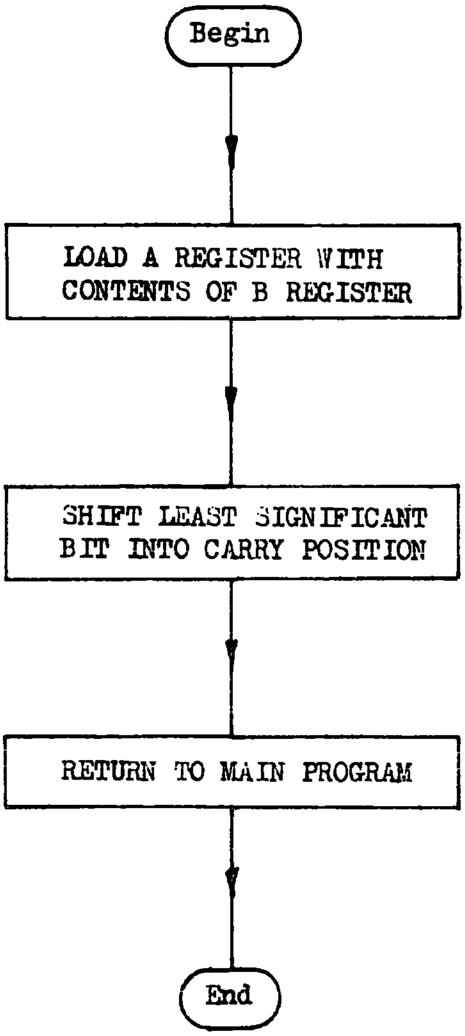
Subroutine SRT2



3.5 Subroutine SRT3

Again this is a very simple subroutine, which is used four times in the master segment, for the determination of symbol orientation (UP or DOWN). The orientation status is, at the stage when this subroutine is called upon, recorded in register B, bit 0. This data is fed into the A register and then shifted into the carry flag position. Thus on return to the main segment the carry flag may be tested to determine the symbol orientation.

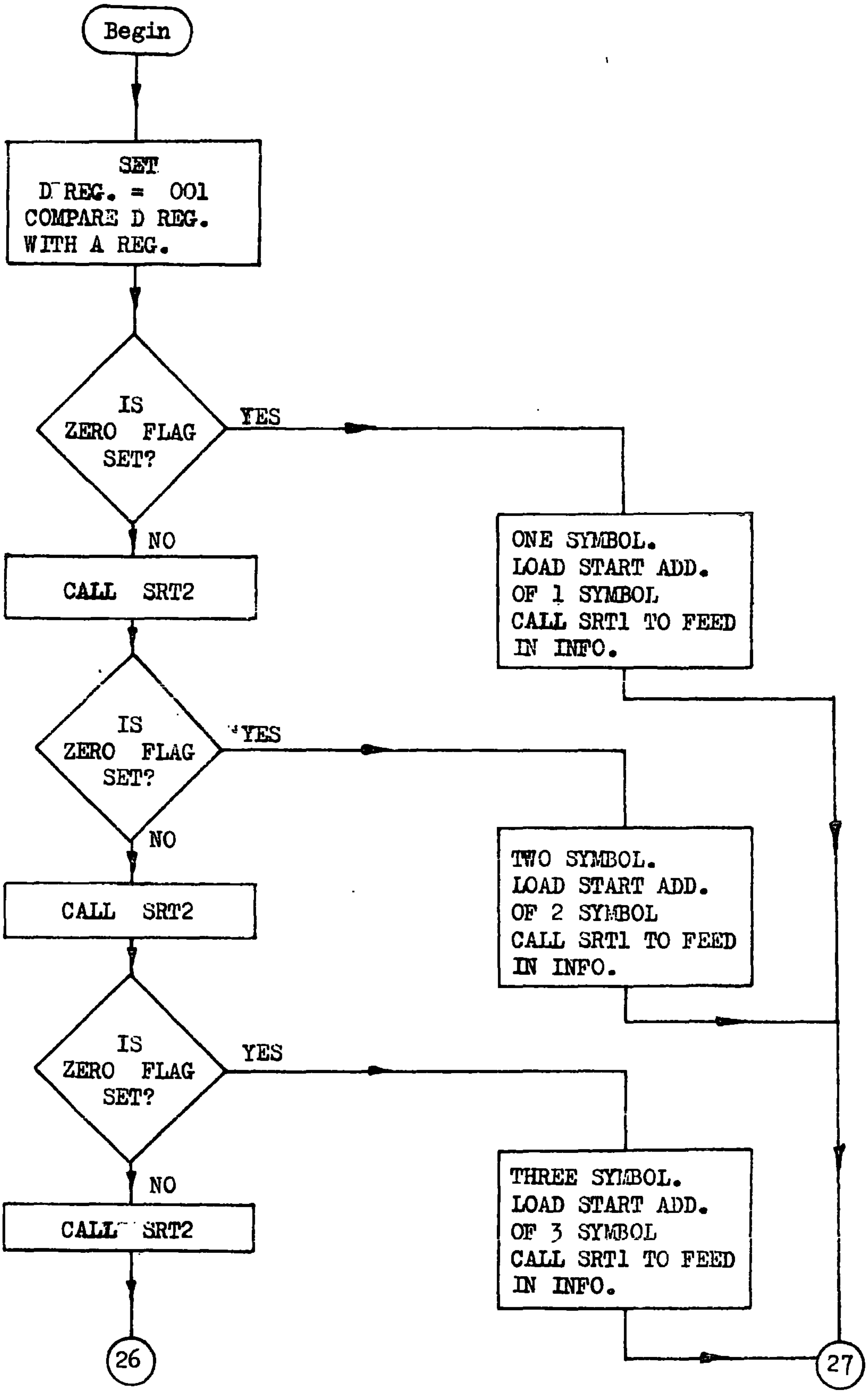
Subroutine SRT3

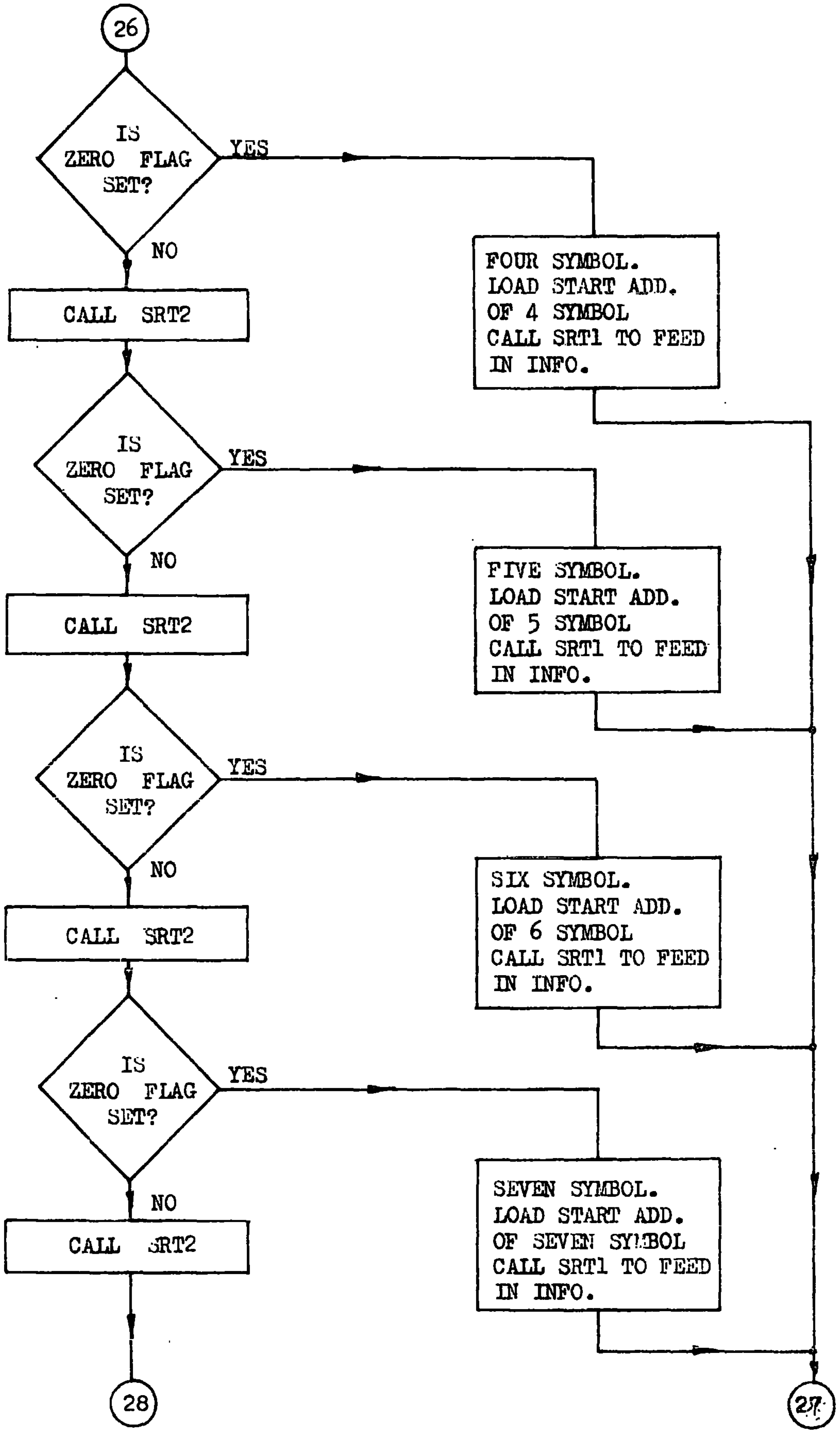

Begin
LOAD A REGISTER WITH CONTENTS OF B REGISTER
SHIFT LEAST SIGNIFICANT BIT INTO CARRY POSITION
RETURN TO MAIN PROGRAM
End

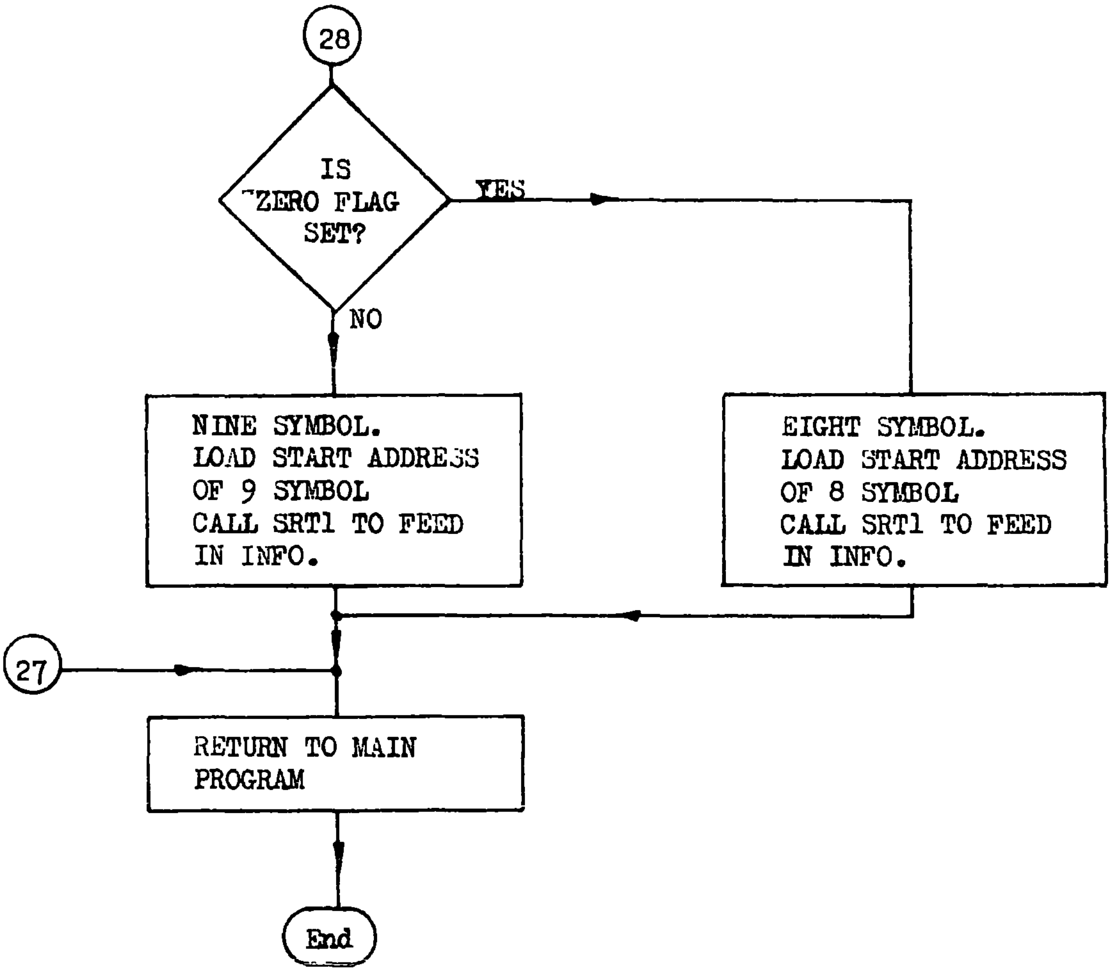
3.6 Subroutine SRT4

This subroutine is used in the search for the correct numeral (other than zero) to be used for the first, second and third characters. It consists of incrementing register D (using SRT2) and testing for equivalence between register A and register D. When equivalence is detected the start address of the correct data modifier block is fed into registers H and L and the modifiers processed using SRT1.

Subroutine SRT4




26
IS ZERO FLAG SET?
YES
NO
FOUR SYMBOL. LOAD START ADD. OF 4 SYMBOL CALL SRT1 TO FEED IN INFO.
CALL SRT2
IS ZERO FLAG SET?
YES
NO
FIVE SYMBOL. LOAD START ADD. OF 5 SYMBOL CALL SRT1 TO FEED IN INFO.
CALL SRT2
IS ZERO FLAG SET?
YES
NO
SIX SYMBOL. LOAD START ADD. OF 6 SYMBOL CALL SRT1 TO FEED IN INFO.
CALL SRT2
IS ZERO FLAG SET?
YES
NO
SEVEN SYMBOL. LOAD START ADD. OF SEVEN SYMBOL CALL SRT1 TO FEED IN INFO.
CALL SRT2
28
27


28
IS
ZERO FLAG
SET?
YES
NO
NINE SYMBOL.
LOAD START ADDRESS
OF 9 SYMBOL
CALL SRT1 TO FEED
IN INFO.
EIGHT SYMBOL.
LOAD START ADDRESS
OF 8 SYMBOL
CALL SRT1 TO FEED
IN INFO.
27
RETURN TO MAIN
PROGRAM
End

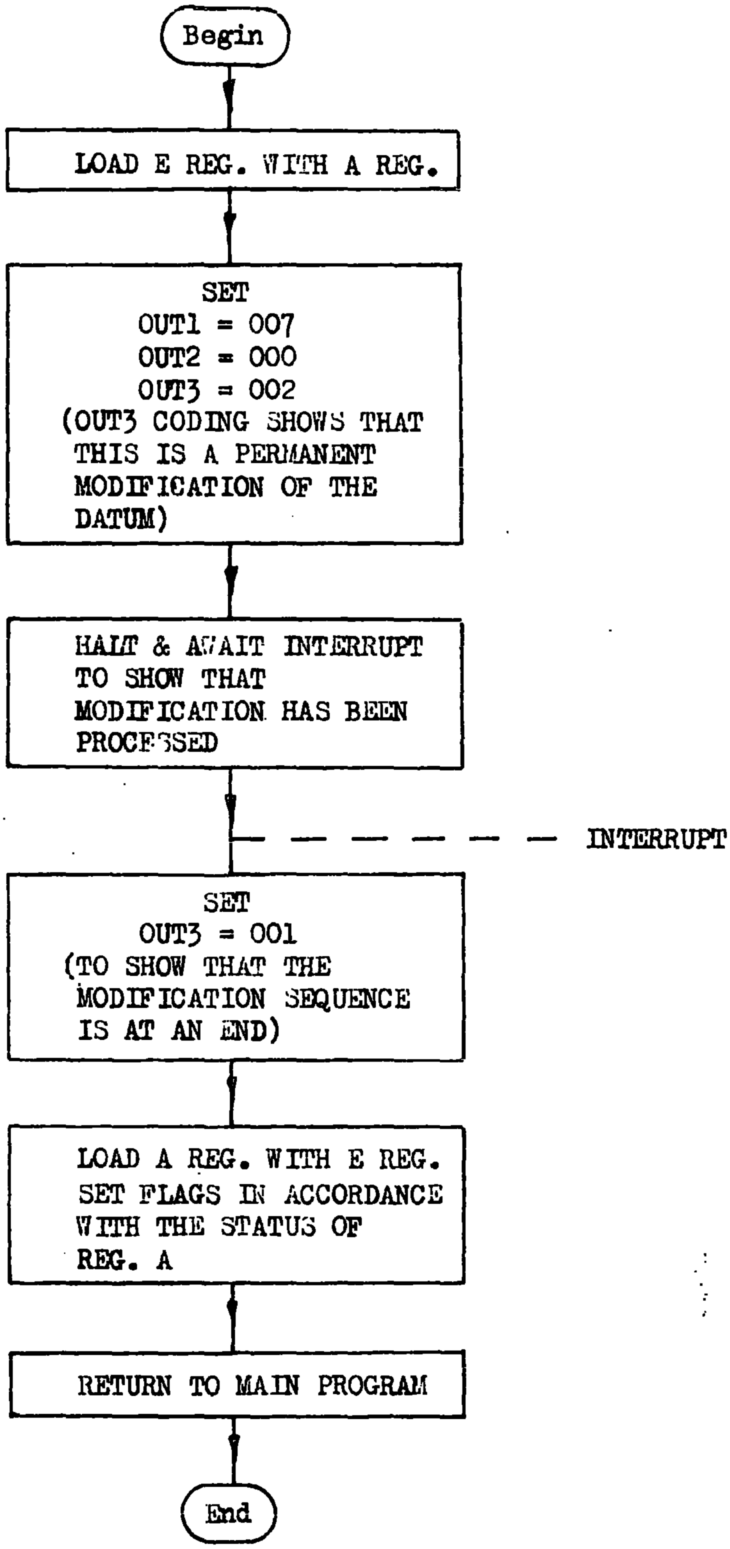
3.7 Subroutine SRT5

This subroutine is used to modify the base address in preparation for the second, third and fourth characters. It cannot be used for the base address modification for the first character since that is a special case, shifting values being different from those used in SRT5.

The subroutine begins by preserving the information held in the A register (which refers to the character type) by feeding it into register E. The necessary line shift and intra-line shift are then fed to OUT2 and OUT1 respectively. OUT3 is set to 002, to show that the OUT1, OUT2 data is to alter the base address, and the data is processed when the program comes to a halt and the STP signal is enabled.

When an INT signal is received the program continues. The OUT3 coding is changed to 001, to show that the base address shifting is at an end. The character type information is returned to the A register from the E register, and the status flags set in accordance with that information. Control is then returned to the master segment.

Subroutine SRT5


Begin
LOAD E REG. WITH A REG.
SET
OUT1 = 007
OUT2 = 000
OUT3 = 002
(OUT3 CODING SHOWS THAT THIS IS A PERMANENT MODIFICATION OF THE DATUM)
HALT & AWAIT INTERRUPT TO SHOW THAT MODIFICATION HAS BEEN PROCESSED
INTERRUPT
SET
OUT3 = 001
(TO SHOW THAT THE MODIFICATION SEQUENCE IS AT AN END)
LOAD A REG. WITH E REG. SET FLAGS IN ACCORDANCE WITH THE STATUS OF REG. A
RETURN TO MAIN PROGRAM
End

CHAPTER 4

HARDWARE

4.1 The Hardware Design

During the following discussions attention is drawn to the schematic diagrams of Appendix IV. These define the detailed logic structure of the prototype unit. It will be noted that in some cases the gating is not in its minimal form but it should be borne in mind that the many connections made during modifications had to utilise whatever spare gates were available on the board that was being modified.

The hardware consists of nine general purpose printed circuit boards. Except in one case this physical breakdown of the circuitry also provides a convenient division of the design into individual sections, each of which will be described in turn. The exception to the above comment is the structure of register A, which, for reasons of space, had to be built on two separate boards. In the description of this unit register A will be considered as a whole.

The various boards are listed below:-

		Schematic Diagram No. (Appendix IV)
Board No.	1. Adder Unit.	S.1
	2. Register A, Part I.	S.2

			Schematic Diagram No. (Appendix IV)
Board No.	3.	Register A, Part II.	S.3
	4.	Register B.	S.4
	5.	Register C.	S.5
	6.	Magnitude Comparators.	S.6
	7.	Main Store.	S.7
	8.	Insertion/Deletion Control.	S.8
	9.	Monitor Control.	S.9

4.2 The Adder Unit

The Adder Unit consists of a 17-bit register, which is used to hold the base address and to which all base address modifiers are added, using the further circuitry contained on this board. A 17-bit, rather than an 18-bit, register is used to hold the base address because of space restrictions on this board. The only limitation this places on the operation of the unit is that all base addresses must refer to positions in the first field, where the most significant bit of the base address is a zero. This causes no problems whatsoever when the V.D.U. is used to display circuit diagrams, indeed there is some advantage in dispensing with field interlacing (and hence with the field bit) for this type of display (see section 5.1). However, as the packing density is considered to be too high (see Appendix IV), it is suggested that the Adder Unit should utilise two boards rather than one, in which case there would be plenty of room for the extra bit position in the base address register.

The base address, from the master computer, is fed into the base address register on receipt of a clock-in signal, BASE ADDRESS

LOAD. It is essential that before the new base address is fed into the register, the register is cleared by the signal CLEAR BASE ADDRESS.

Once the base address is staticised in the base address register it is fed to the inputs of an 18-bit adder network, where it is summed with the output of the OUT1, OUT2 ports (in the manner described in section 2.3). The result of this addition is presented to other units of the V.D.U. for further processing. However, should the OUT1, OUT2 information be accompanied by OUT3 = 002, the base address itself is to be changed (in preparation for the display of characters). If this is the case then, the appearance of the STP signal, which occurs after the OUT1, OUT2 and OUT3 data has been stabilised, causes the new base address to be clocked into the base address register. Figure 4.1 shows one element of the register/adder arrangement to illustrate t: is process.

This board also contains a bistable which is used to staticise the interrupt status. An interrupt will be recorded:-

(i) Following a change of base address.

(ii) Following the completion of an INSERT or DELETE operation (this signal is derived from the Insert/Delete Control board).

(iii) On receipt of a signal from the master computer when new information is to be fed into the microprocessor.

The interrupt signal is held until the STP signal goes off, showing that the microprocessor has actioned the interrupt.

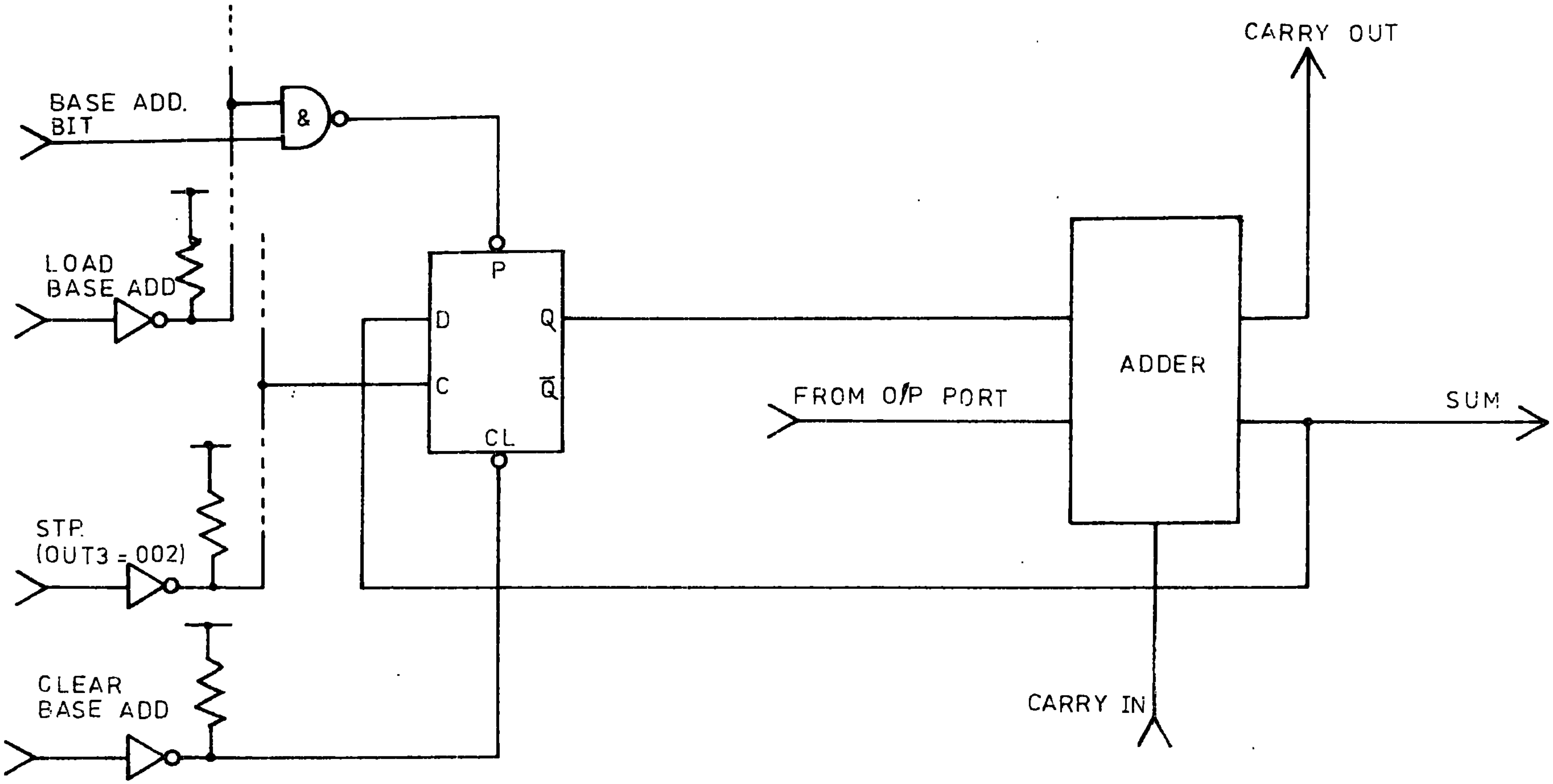


FIG 4.1 ADDER UNIT (DETAIL)

4.3 The A Register

Register A consists of a 19-bit register which is used to hold the results of the additions performed by the Adder Unit. A diagram of one stage of the register is shown in figure 4.2. One bit of the A register holds the flag bit, which is derived directly from the output of the microprocessor. The information is fed into the A register by the signal PST REG A (preset register A) which is produced by the occurrence of OUT3 = 004 and a STP signal. Register A is cleared by the status OUT3 = 004 and $\overline{\text{STP}}$, which will precede any feed-in status. The signal PST REG A should be short in order not to slow up the store modification process. However it must be greater than 30 ns, in order to ensure the correct setting of the SN 7474 bistables[19] used in the A register. The signal width used in the prototype unit was 90 ns. The information held in the A register is fed back on itself (keeping the contents of A constant) until, under the command of the Insertion/ Deletion Control board, the information path changes (as described in section 2.4).

4.4 Registers B and C

Each of these registers occupy a single board, however their structure and operation is so similar that they may be described together. Figure 4.3 shows an example of a single stage, which applies to both register B and register C.

Registers B and C form the first two word positions of the display store. In each case the input to the register may come from one of two sources, under the control of signals from the Insertion/ Deletion Control board. During a read sequence the input to B will be derived from the last register of the display store, known as

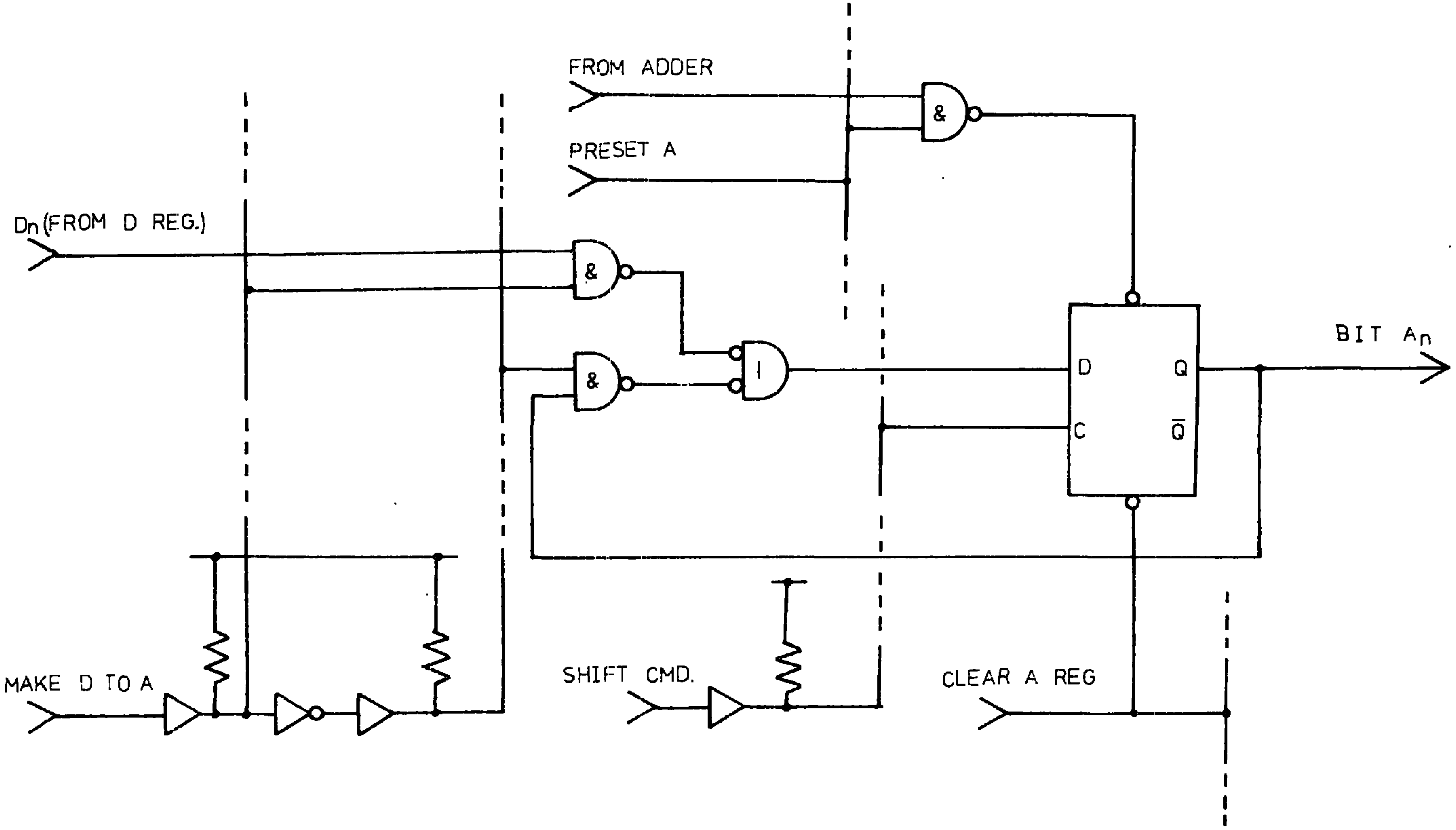


FIG. 4.2 REGISTER A (DETAIL)

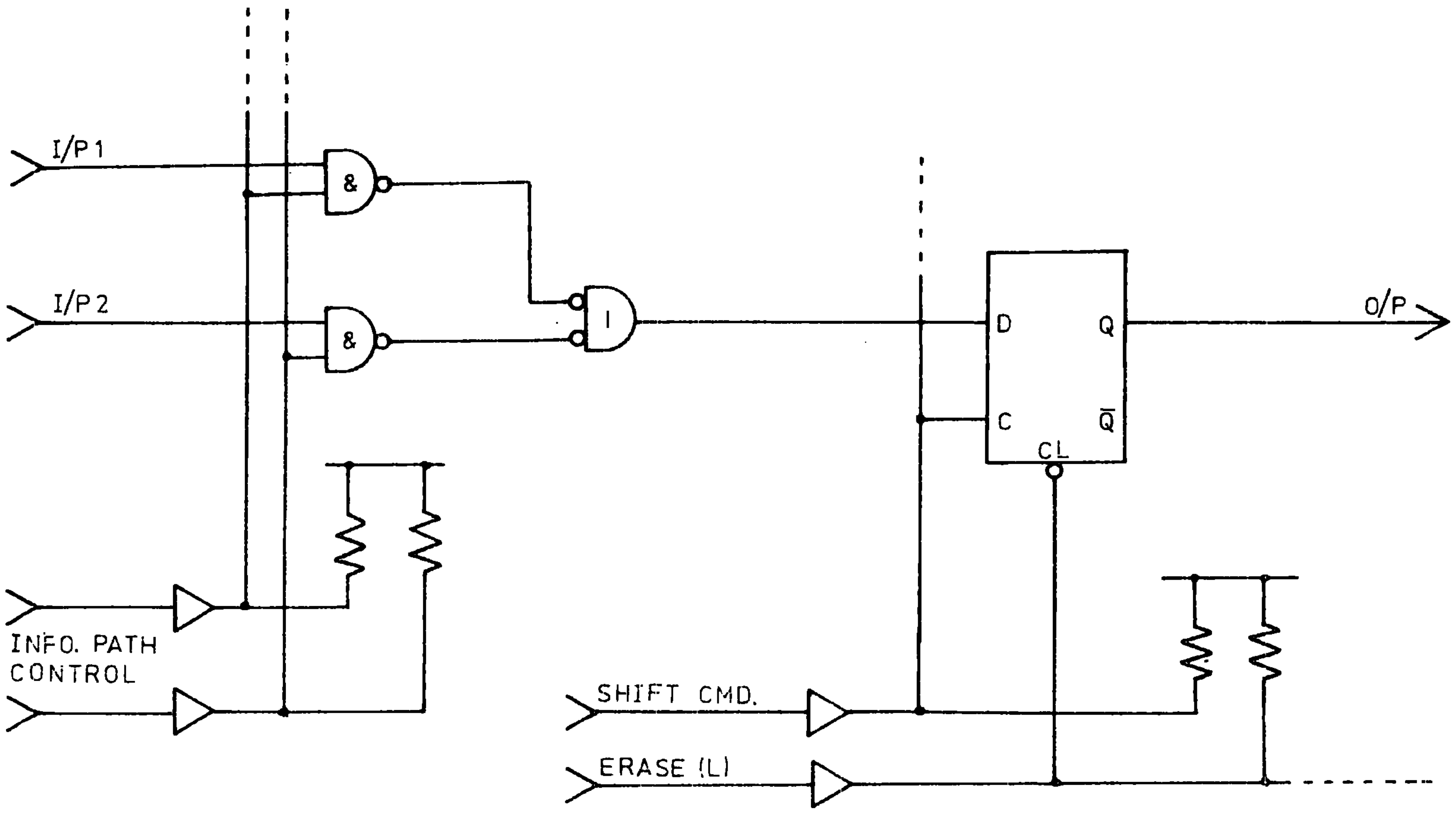


FIG 4.3 REGISTERS B & C (DETAIL)

register D, and the output of register B forms the input to register C. However during the insertion process the input to register B will, at some point, be derived from register A. Whereas during the delete process the input to register C is derived from register D (see section 2.4.2).

It is via the B and C registers that the ERASE command is made effective. The ERASE signal forces both registers into an all zero condition, a state which is then shifted through the rest of the store. As on other boards, signals with a heavy loading are buffered using the SN 7407. In order to make the most efficient use of the available gates, the buffering for both registers is carried out on the B register board.

4.5 Magnitude Comparators

The comparators may be divided into three sections. The first two sections are concerned with comparing the status of register A (other than the flag bit) with the status of registers B and D (again excluding the flag bit). The signals derived from these comparisons cause the Insertion/Deletion Control Board to organise the information path during Insert and Delete operations. The third comparator section compares the status of register D with the status of the counter which defines the position of the raster on the monitor screen (signals E_0 - E_{17}). When equivalence is detected between these two sets of signals the status of the display is changed and the store shifted through one word position (see section 2.5).

It will be noted that the organisation of this final comparator section is rather different from that of the first two sections. Its more parallel structure was required in order that the equivalence signal, produced during the high speed read operation, was long enough to provide the required shift command.

4.6 The Display Store

A single element of the 19-bit wide store is shown in figure 4.4. The Display Store board holds the bulk of the refresh memory in nineteen parallel 64-bit shift registers. In a commercial system a much longer store would be required, and this might occupy several boards. As it is necessary to run the MOS registers at 11 V, in order to achieve the required shift rate, interfacing circuitry is included at both the start and the finish of the store, to convert from TTL to MOS signal levels and vice versa.

The conversion from TTL signals levels to the signal levels required to drive the MOS shift registers is carried out by using SN 7407 open collector buffers with a pull-up resistor to the MOS supply rail. The pull-up resistors need to be as small as possible in order that the signals reach a "1" level as fast as possible. However they must be large enough to protect the output transistors of the SN 7407 units from excessive currents during the time they are switched on. The conversion from MOS signal levels to TTL signal levels is easily achieved by means of the CD 4049 inverting interface unit.

4.7 Insertion/Deletion Control

The Insertion/Deletion Control board is perhaps the most complicated section of the whole design. As may be seen from figure S.8, Appendix IV, the gating structure is quite involved, and does not follow the regular pattern of the other boards.

The action of this board may be split up into four sections:-

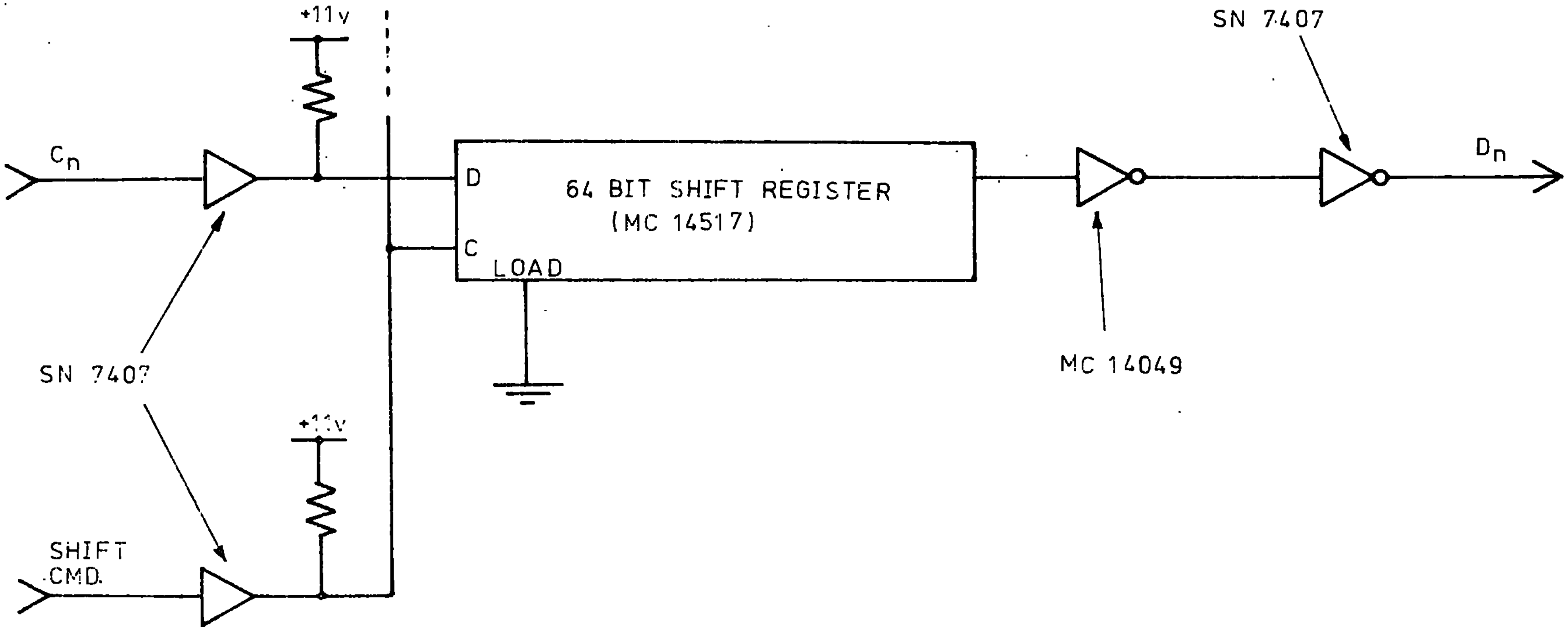


FIG 4.4 DISPLAY STORE ELEMENT

(i) the control of the information path during Insert operations,

(ii) the control of the information path during Delete operations,

(iii) the production of an interrupt signal to show that an Insert, or Delete, operation has been completed,

(iv) the control of the signal to shift information through the display store during Insert, Delete Erase and Read operations.

4.7.1 Information Path Control (Insert)

A detailed diagram of the gating relevant to the control of the information path during the Insert operation is shown in figure 4.5. The gates operate under the command of signals produced on the Magnitude Comparator board. These result in the formation of commands which cause the information path to change from D, B, C, ... D to D, A, B, C, ... D in order to facilitate the introduction of new data to the display store. The operation is described in detail in section 2.4.1, but in general the conditions for information insertion are:-

(i) $A < D$ and $B = 0$ (when the new address is smaller than any other address in the information block).

(ii) $B < A < D$

(iii) $A > B$ and $D = 0$ (when the new address is larger than any other address in the information block).

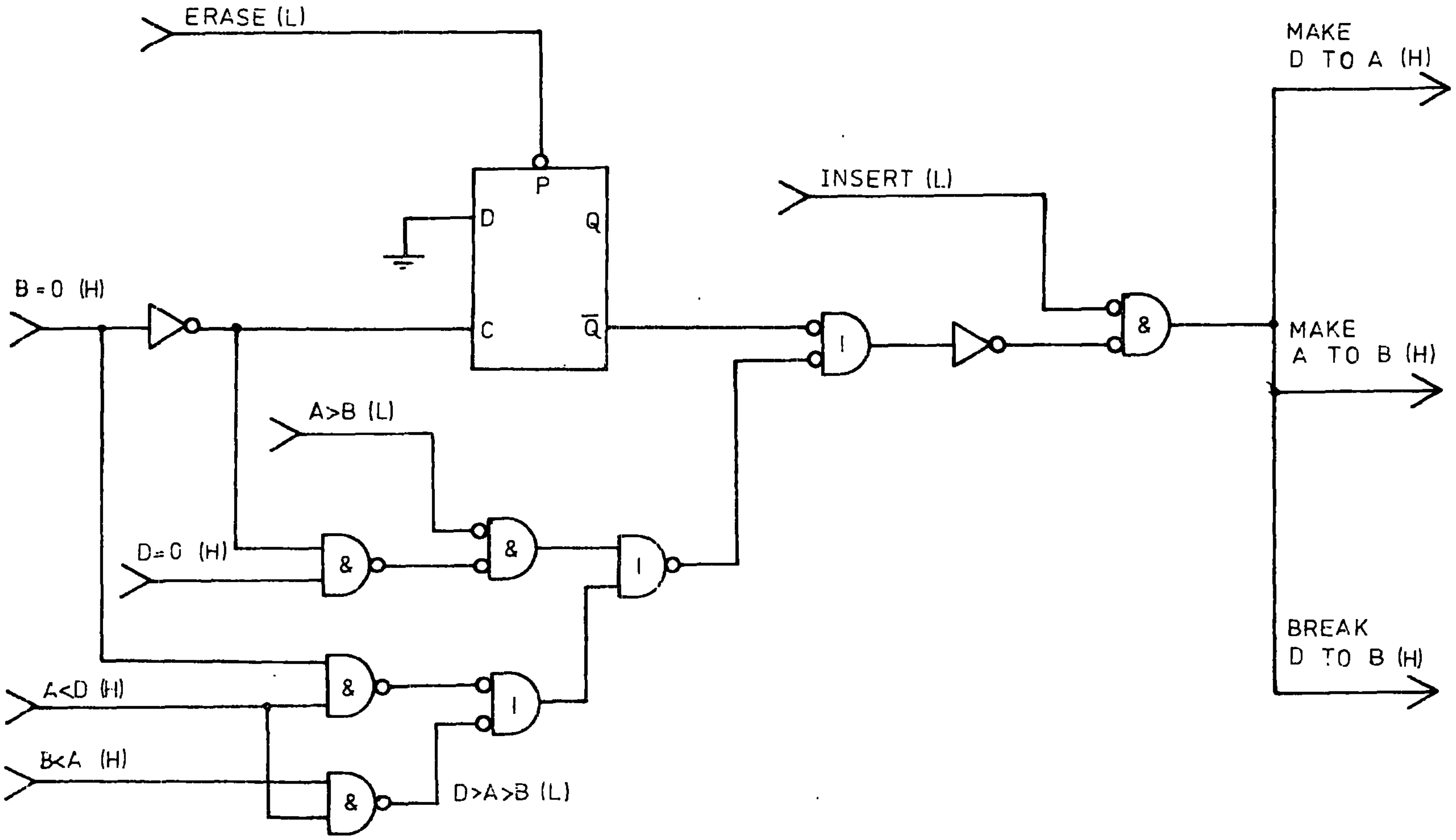


FIG 4.5 INSERT CONTROL

Special consideration must be given to the insertion of the first address word into an empty store. Condition (iii), which initially does apply, is not sufficient to cover this case since some bits of the address word are shifted into register B faster than others. At some stage the condition A > B no longer holds and the signal path A to B is cancelled before all the information is staticised in register B. This problem was experienced during the commissioning of the equipment, and the solution adopted was the introduction of a bistable which is set by the ERASE command. Thus whenever the display store is completely emptied this bistable is set, and, as long as it is set, it maintains the information path D, A, B, C, ... D. The bistable is not reset until register A goes to an all zeros condition, that is until after the slowest bit transfer from register A to register B has taken place.

4.7.2 Information Path Control (Delete)

The section of the control board which organises the information path during a Delete operation is shown in figure 4.6. Again the Delete operation itself has already been discussed in section 2.4.2. The main event is the reduction of the information path from D, B, C, ... D to D, C ... D when A = B. This is achieved by using the status A = B (other than when A = 0) during the presence of a DELETE signal to set a bistable. This breaks the path D to B and completes the paths D to C and D to A. This situation continues until the end of the information block is reached, when the status A = D = 0 resets the bistable.

4.7.3 Interrupt Control

At the end of each Insert and Delete operation an interrupt

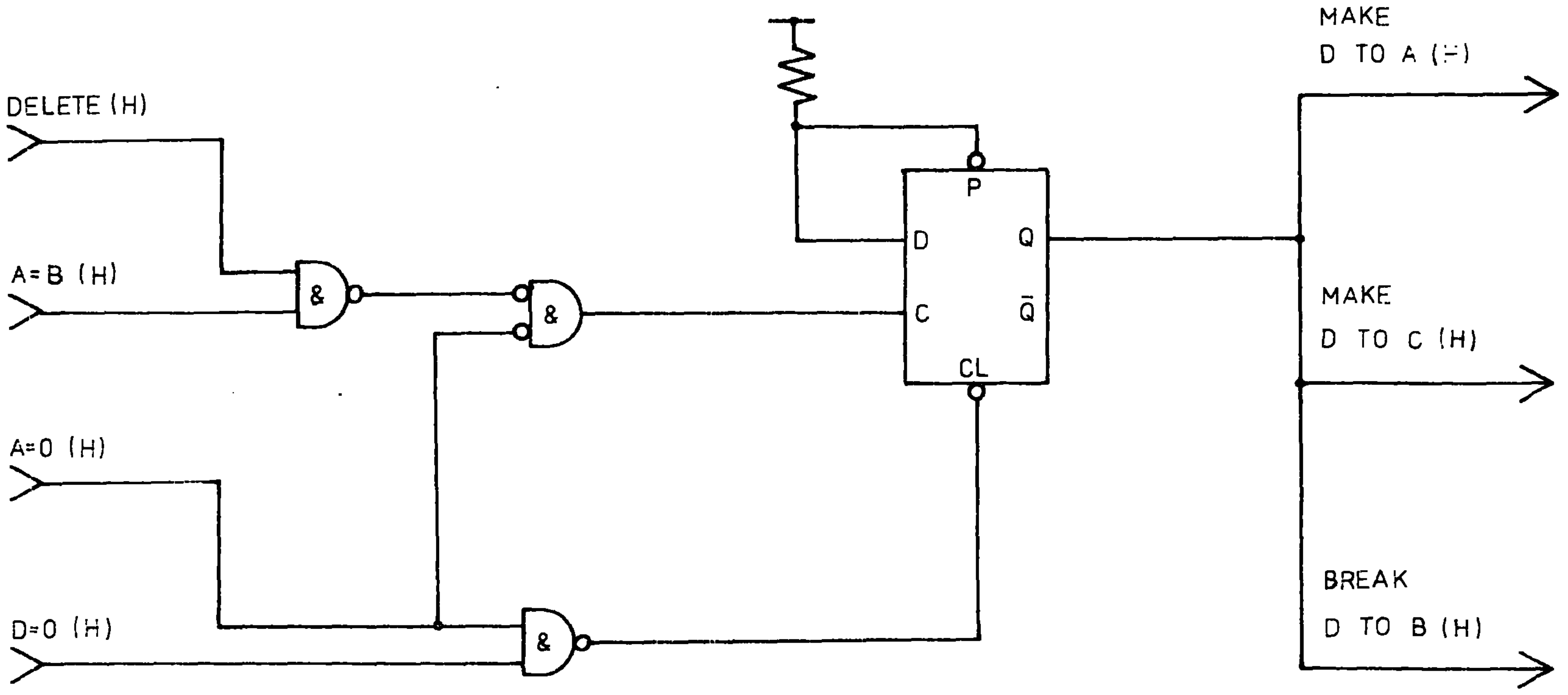


FIG. 4.6 DELETE CONTROL

signal must be sent to the microprocessor in order that the next address modifiers might be presented. The organising logic for the control of such interrupts is shown in figure 4.7. Interrupts are sent to the microprocessor, via the Adder Unit, when register A becomes zero in the case of an Insert operation, or when register B becomes zero in the case of a Delete operation. The use of a clocked bistable ensures that an interrupt is only sent when register A goes to an all zero state, and not merely when register A is all zeros (as occurs quite naturally during Insert and Delete operations). This bistable is reset by a return signal from the interrupt bistable on the Adder board, showing that the interrupt has been recorded there, and also by an ERASE signal, to ensure that it is in the correct state after the system has been powered-up. Before an interrupt signal is sent it is also a requirement that the signals STP and OUT3 = 004 are present, showing that the microprocessor is in the correct state to accept this form of interrupt.

4.7.4 Shift Command Control

A detailed diagram of the circuitry controlling the shifting of information in the display store is shown in figure 4.8. Address words are shifted through the store in response to:-

(i) MOVE STORE signals from the Monitor Control board (see next section) during a read operation,

(ii) register D being in an all zero state (in order that the first word of the information block is ready for the start of the next frame, see section 2.5),

(iii) signals STP.OUT3(= 004) during Insert and Delete operations,

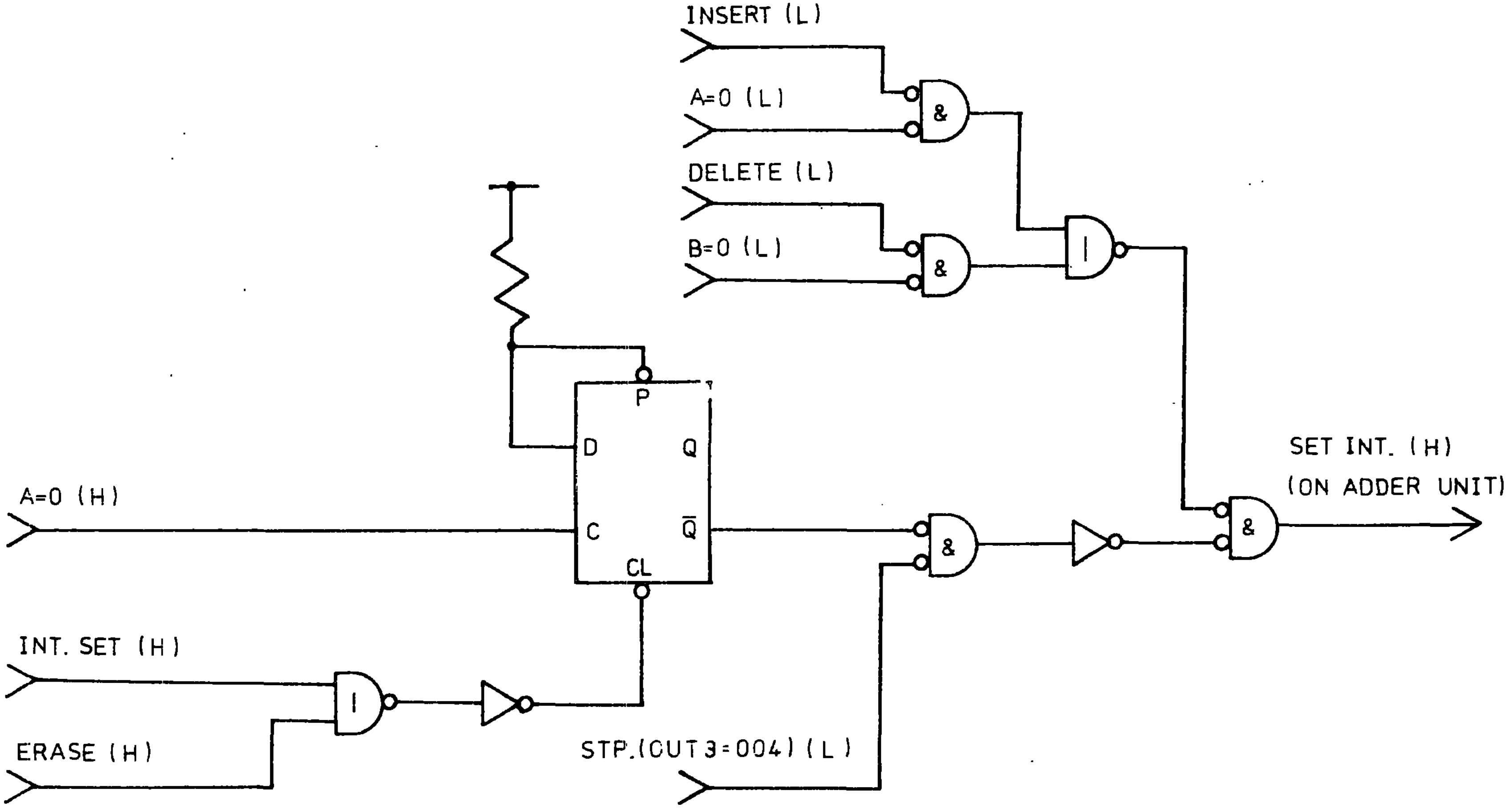


FIG. 4.7 INTERRUPT CONTROL

(iv) an ERASE command, when the all zero status of registers B and C is shifted through the whole store.

Each MOVE STORE pulse causes only a single shift through the store, but the conditions mentioned in (ii), (iii) and (iv) above cause a sub-harmonic of the main clock frequency to shift information through the store until the signal in question (D = 0, STP.OUT(= 004), or ERASE) is removed.

In the original design the appearance of the STP signal (when OUT3 = 004) immediately caused the CLK II signal to be fed through to the shift signal of the store. If the appearance of the STP signal coincided with a low state on the CLK II line, however, this resulted in the immediate production of an edge on the shift line of the store. This led to the corruption of information, since it was only on the appearance of the same STP signal that the result of the addition of the OUT1/OUT2 information to the base address is fed into register A. The introduction of the bistable shown in figure 4.8 delays the effect of the STP signal on the shift line until after the clock next goes high. This allows at least half a CLK II period between the appearance of the STP signal and the first shifting signal being sent out to the display store, ample time for the contents of registers A, B and D to determine the correct information path. This bistable is disabled when register A is cleared, that is at the end of the Insert, or Delete, operation.

4.8 Monitor Control

The Monitor Control board contains the line and line-section counters which, together with the necessary gating, provide monitor synchronising (line and field) and video signals.

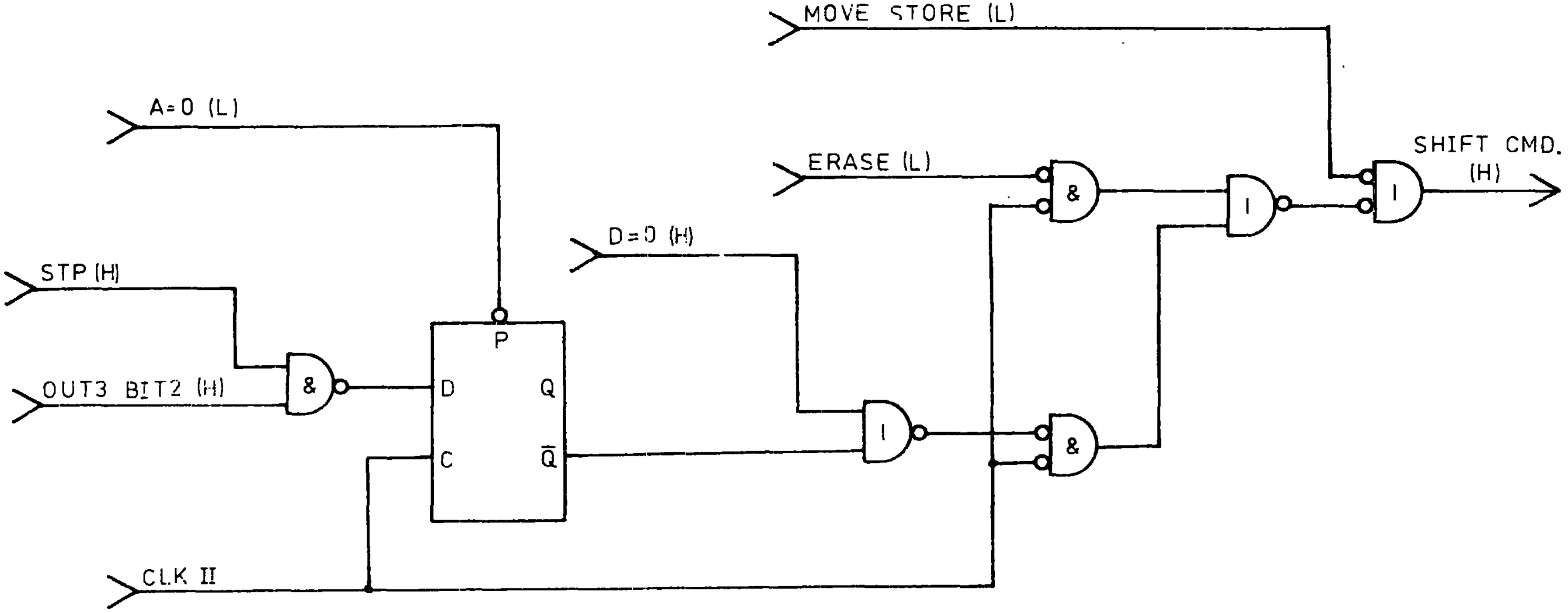


FIG 4.8 SHIFT COMMAND CONTROL

The counter may be divided into two parts. The first part consists of an 8-bit synchronous counter which, together with the clock signal itself, divides each line into 512 sections. When all eight bits are at one level (at the end of each line) a monostable is fired to provide the 12 μs line blanking pulse. After a delay of 1.5 μs, provided by the open collector/Schmitt trigger arrangement detailed in figure 4.9, a second monostable is fired to provide a line sync. pulse of 4.7 μs[5]. A similar operation occurs at the end of each field when the second counter block, the line counter, is set to all ones. This condition is used to fire the 3.1 ms field blanking pulse (this pulse covers the time to draw the unused, 312-255 lines), after a delay of 1 ms the 1.2 ms field sync. pulse is fired.

The field blanking pulses are fired as soon as the less significant half of the line section counter is in an all ones condition, and when the line counter itself is also set to all ones, i.e. half-way through the 256th line of each field. However it is necessary to distinguish between the first and second fields in order to achieve the line interlacing described in section 2.3. The final stage of the line counter allows this distinction to be made, since it is in fact a field flag, being set to zero during the first field, and set to one during the second field. If the field flag is zero when the field blanking pulse is fired then the line section count is allowed to continue without interruption and a line sync. pulse is fired at the end of the 256th line (within the field blanking period). The line count (other than the field flag) is held at zero during the field blanking period, the line sync. pulses, however, continue. If the field flag is set when the field blanking pulse is fired, at the end of the second field, a monostable is fired which resets the line

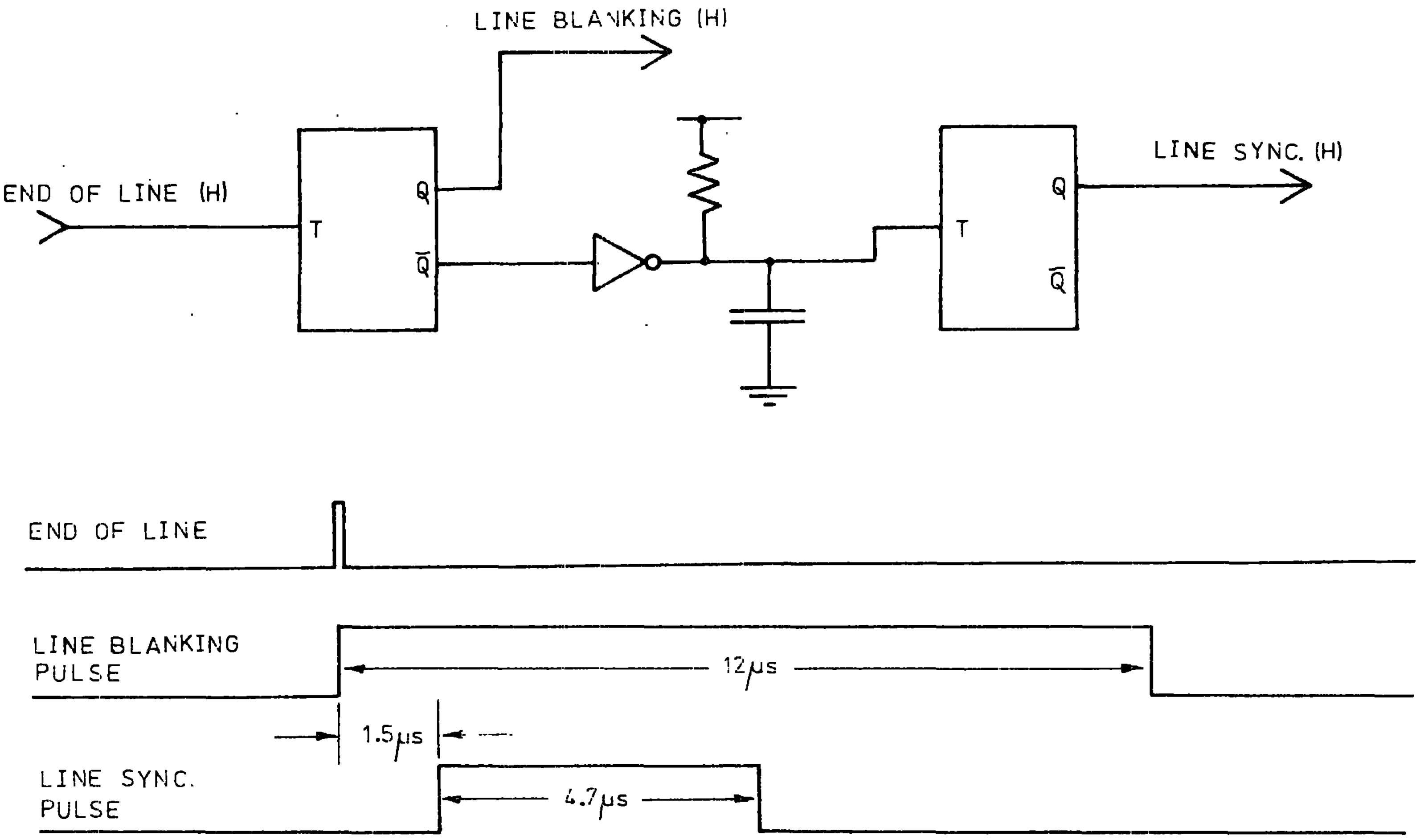


FIG 4.9 LINE BLANKING/SYNCHRONISATION

counter. The next line sync. pulse is not then produced until the line section counter again reaches an all ones condition, that is a time equivalent to one-and-a-half lines after the previous line sync. pulse. This sequence means that the line sync. pulses of the first field are half a line out of step with the line sync. pulses of the second field, and thus the two fields are interlaced, see figure 4.10.

If, as a result of the comparison of the line and line section counters with the contents of register D, an equivalence signal is received, from the comparator board, a monostable is fired to stretch the pulse to a width suitable for the shifting of the display store. This pulse is sent to the Insertion/Deletion Control board as the MOVE STORE signal, but is inhibited during INSERT and DELETE operations in order that spurious shift signals do not interfere with those processes.

The equivalence signal is also needed to control the video signal. The video control section is detailed in figure 4.11. The general level of the display is determined by the setting of the bistable. If a white-on-black display is required the signal is set to the black level at the start of each field by using the field blanking pulse to clear the bistable. If a black-on-white display is required the field blanking signal is used to preset the bistable to a one level. If an equivalence signal should appear when the flag bit of register D is at a one level then the video intensity change is to last for only one line section. In this case the stretched equivalence signal alters the status of the video signal, from black to white or white to black, without changing the status of the bistable. Thus when the equivalence signal is removed the video signal returns to its original level. However if the equivalence signal appears when the

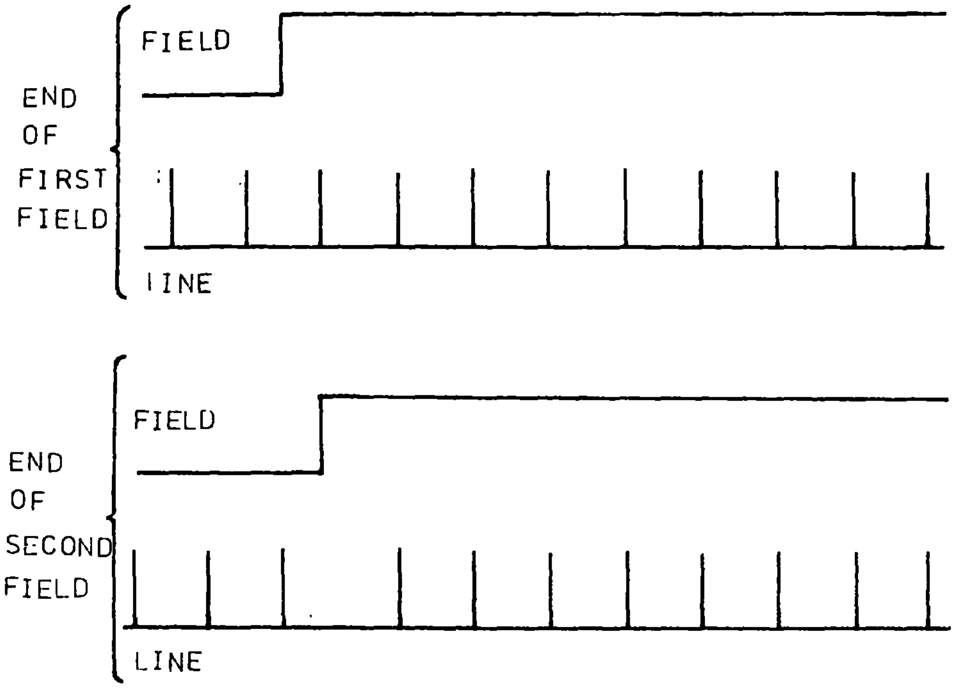


FIG 4.10a INTERLACING SIGNALS

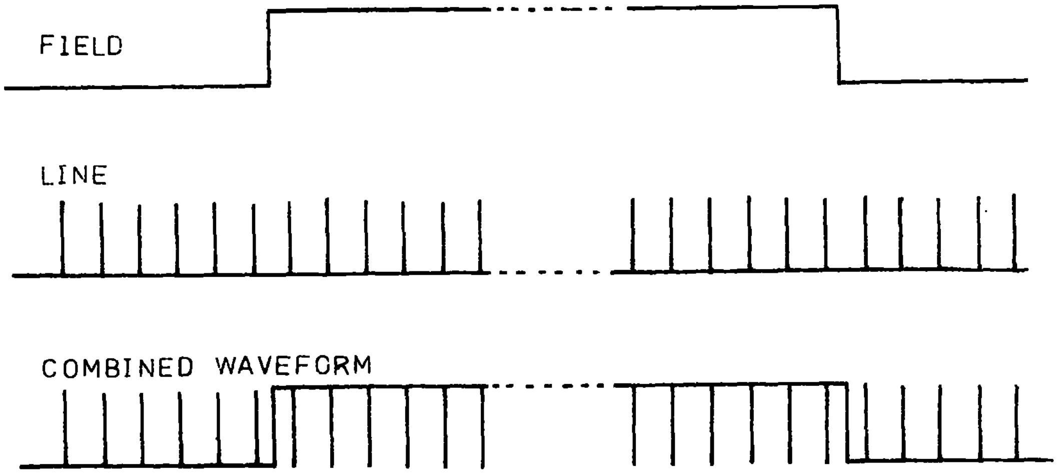


FIG 4.10b COMBINED SYNC. SIGNAL

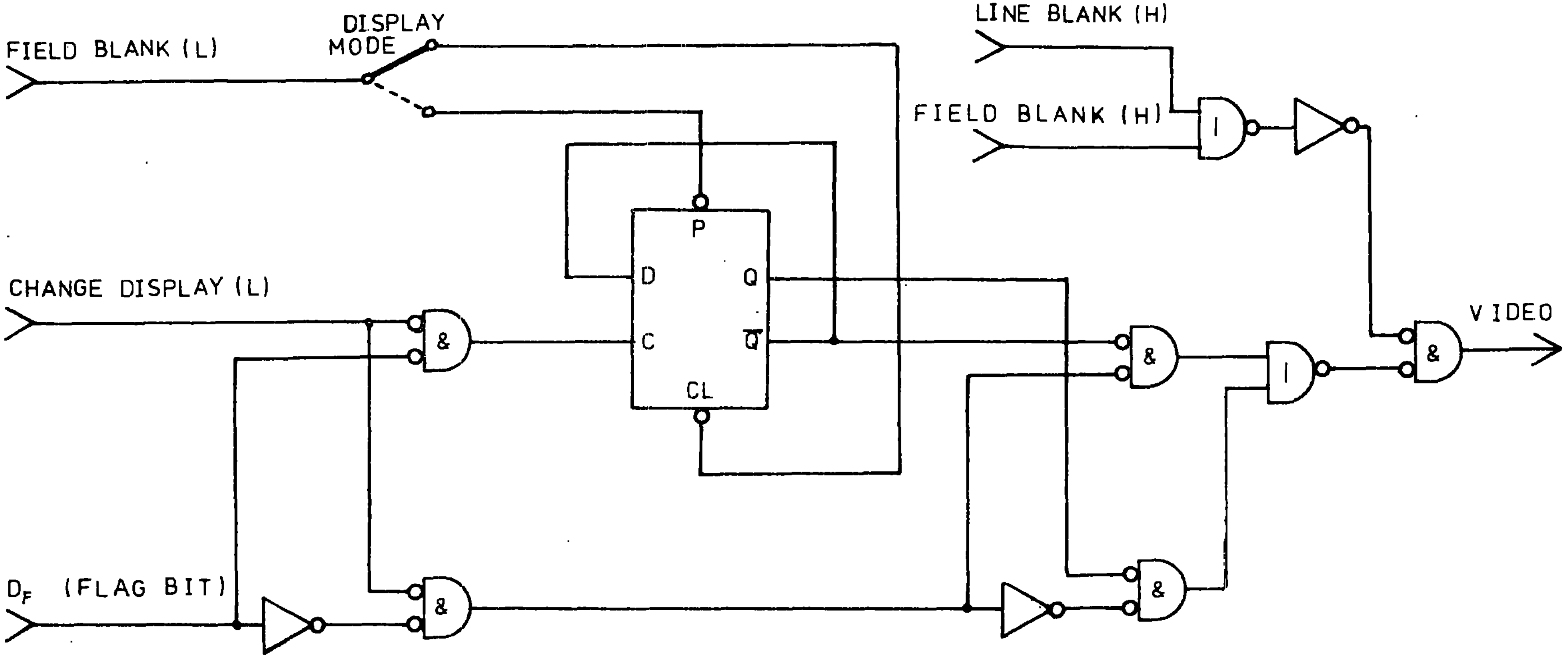


FIG. 4.11 VIDEO CONTROL

D register flag bit is low a permanent change in video level is required, and thus the status of the bistable is altered.

4.9 Cost

In determining the cost of the prototype unit no allowance has been made for the provision of a television monitor, since a monitor purchased for a conventional use could serve as the display device. Only those components of the SIM-8 microcomputer that are necessary to the operation of the V.D.U. are included in the pricing, e.g. no use is made of the 1K of R.A.M. available in the SIM-8 and so is not included in the estimated cost.

Two overall prices are given. One bases all costs, other than discrete components such as resistors and capacitors, at their unit price, the second costs all components, other than the microprocessor and power supplies (which are used only once in each unit), at 100 off prices.

Price Breakdown:

	Unit Pricing	100 off Pricing
	£	£
Random Logic	88	60
Display Store (10 x MC 14517CP)	33	22
Discrete Components	(10)	10
General Purpose P.C. Boards	43	34
P.C. Boards Edge Connectors (Wire-wrap)	15	12
8008 Microprocessor	20	(20)
P.R.O.M.s (1702A)	77	63
Microprocessor Units (O/P Latches, Interrupt Circuitry, etc.)	18	11
Power Supplies:		
5 V, 9 A	35	(35)
12 V, 0.5 A	17	(17)
TOTAL	356	284

The program used in the prototype unit occupies three 1702A P.R.O.M.s and the display data uses four more. This however does not include a full set of display information for vertical components, to include this would require another P.R.O.M. - eight in all. The display store of the prototype unit is only 64 words long, which is adequate for demonstrating the principles of the design but not sufficient for a commercial unit. It is considered that a display store of 1,024 words would be required in such a device. A large store however will cost proportionately less than a small one, since a large store will justify the implementation of shift-register multiplexing (outlined in section 1.6.1) and allow the use of slower, and therefore cheaper, shift registers. Such a store of 1,024 words may be constructed using the Signetics 2527, 3 MHz, 256-bit shift register, at a total device cost of £203, giving a machine cost of approximately £467. However it is one of the main features of the design that the unit may be used with only a small, inexpensive display store, which may be expanded at a later date.

CHAPTER 5

DISPLAY STYLE & PROJECT REVIEW

5.1 Display Format

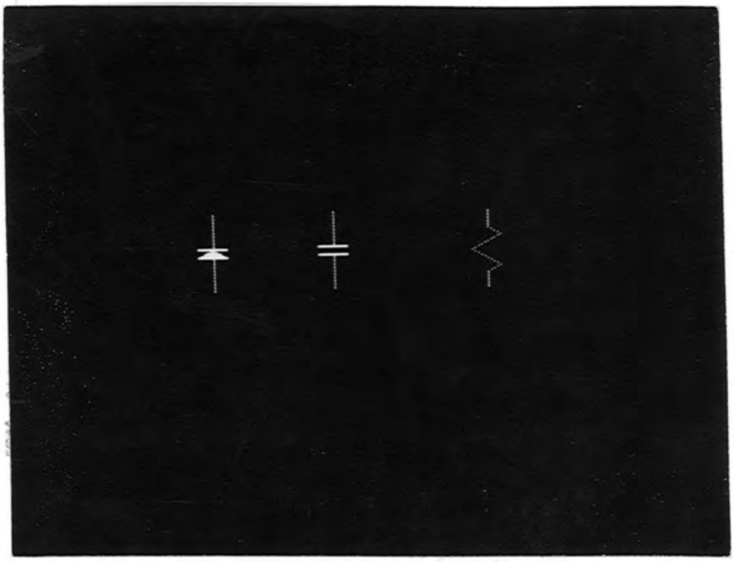
Plates P5.1 to P5.15 illustrate the type of display produced by the V.D.U. Unfortunately, as the display store was just 64 words long, it was possible to display only a single horizontal circuit symbol and up to three characters (depending upon the symbol and character types) at any one time. The restriction on the display of vertical characters is even greater because of the large number of addresses required to describe the component leads. It was, in fact, necessary to truncate the leads of vertical components in order to display even one of them. Plate P5.1 showing three vertical components is a multi-exposure photograph, only one of the symbols was displayed at any one time.

As only a small display store was available a rather stylised form was adopted for the display of voltage and current sources, as shown in plates P5.2 and P5.3. This allowed the overall size of the symbol to be smaller than if a circle had been used, while still maintaining the clarity of the display. It will be noted that the symbolic form of the inductor (see plates P5.4 and P5.9) is different from that of the other components. Originally this too was in order to save on the number of addresses required for component description. However it is suggested that in any future work this format is adopted for all circuit components. This will allow common information (that relating to the leads and an empty box) to be used for all symbols, with the need to store only a small amount of information, this being a single letter R, C, V, etc., in order to distinguish between symbol types.

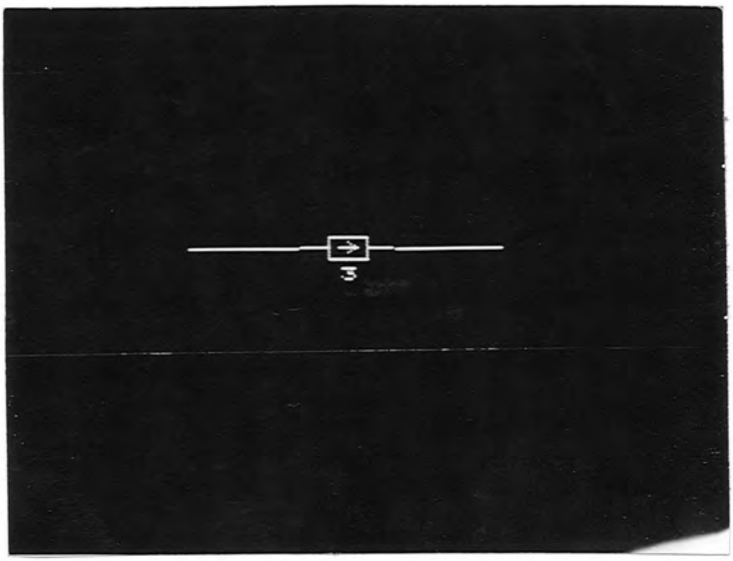
The photographs shown in this chapter also serve to illustrate the major drawback with the addressing style adopted. This is that new information inserted into the display store cannot relate to screen positions which are immediately adjacent to positions already held in store, as this would require the store to shift in zero time. Thus it is not possible for the specific symbol information, relating for example to a capacitor, to be inserted between the common lead information, without taking account of this restriction. One solution to this problem is to put the lead information relating to a symbol in a different field from that used by the lead section of the individual symbol design. This allows the separate lead sections to be close together on the screen, but widely separated in time. However this does result in a slight step in the leads of the displayed component, e.g. Plate P5.5. Another possible solution is to introduce a small gap in the leads to allow time for the display store to shift to the next address (e.g. Plates P5.6 and P5.7), however this is not considered to give a very satisfactory display. Perhaps the best solution is to eliminate the interlacing between fields, so that the second field is directly superimposed on top of the first field. This allows parts of the image to be immediately adjacent to one another, or even to overlap, and yet still be widely separated in time. Such action does of course mean that the vertical definition is reduced, there now being only 256 separate lines instead of 512. However for the display of circuit diagrams this is considered to be quite sufficient, and has the advantage that in some cases the amount of information required to describe a particular component can be reduced. Plates P5.8, P5.9 and P5.10 are examples of the type of image displayed when interlacing is not used (all the other plates employ interlacing).

The style of the display may be either a white image on a black background, or a black image on a white background. If the ambient light intensity was high the white-on-black display was found to be the most satisfactory. However, when the operator is alert, it is possible to discern a slight flicker on some parts (particularly long horizontal lines) of the image. This is due to the fact that the refresh rate is only 25 frames/sec. This flicker becomes less discernible the longer one works with the display. With the black-on-white image no flicker is noticeable because of the high background illumination, and this gives a very easy to read display as long as the ambient lighting is not too intense. It is indeed unfortunate that the photographs of this display mode cannot adequately convey its merits compared to the white-on-black image.

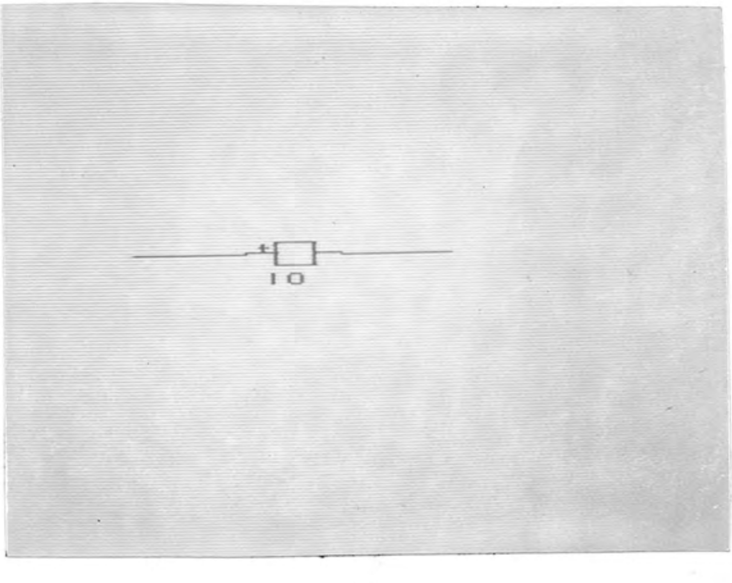
P5.1

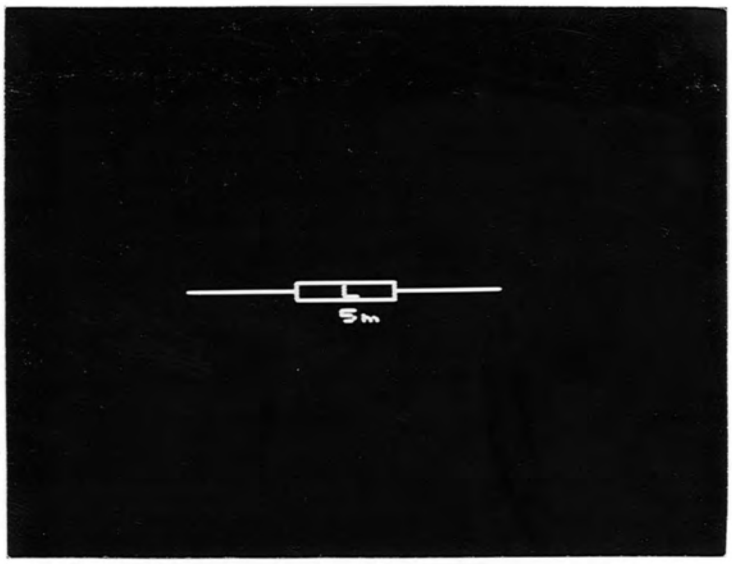


P5.2

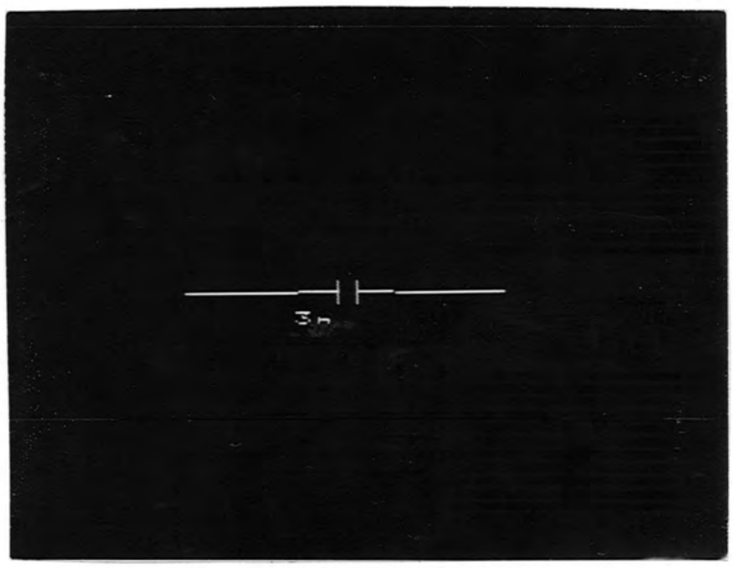


P5.3

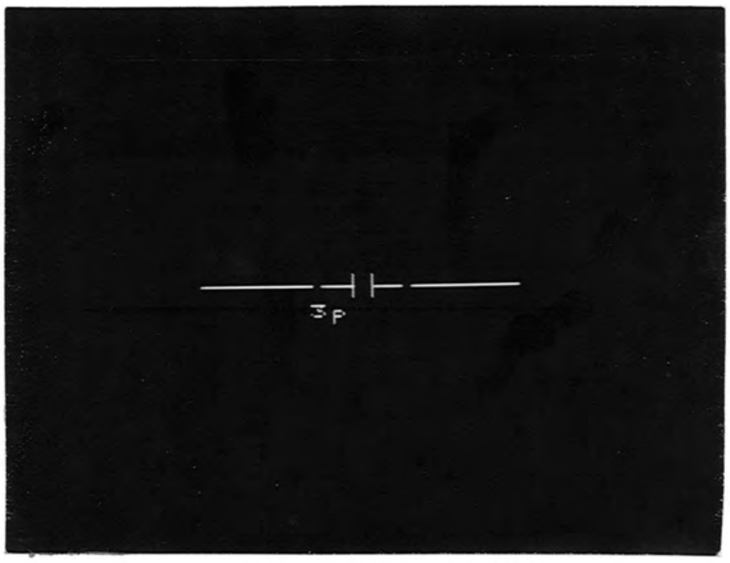




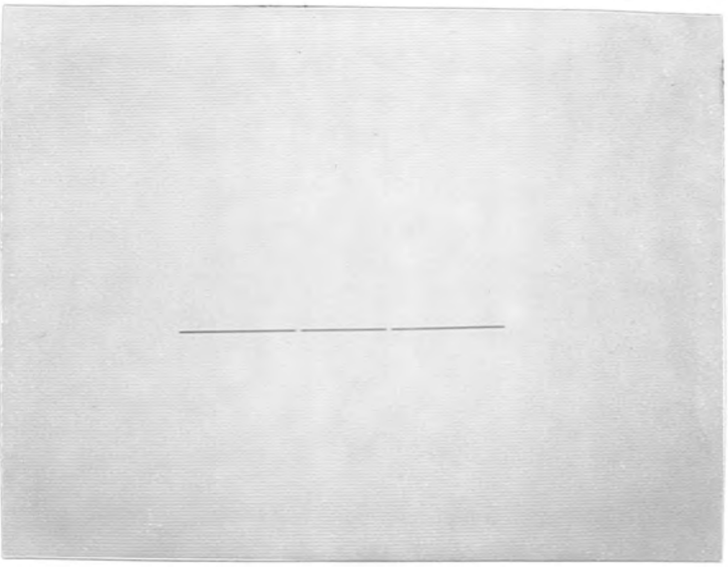
P5.4



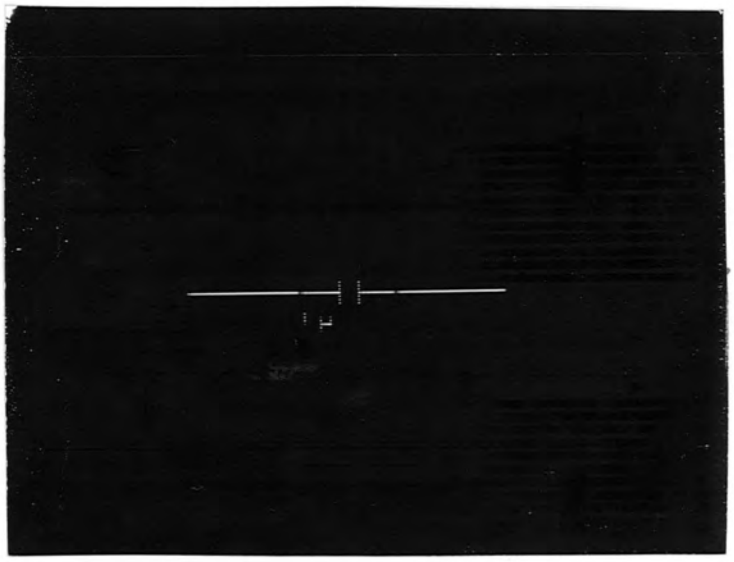
P5.5



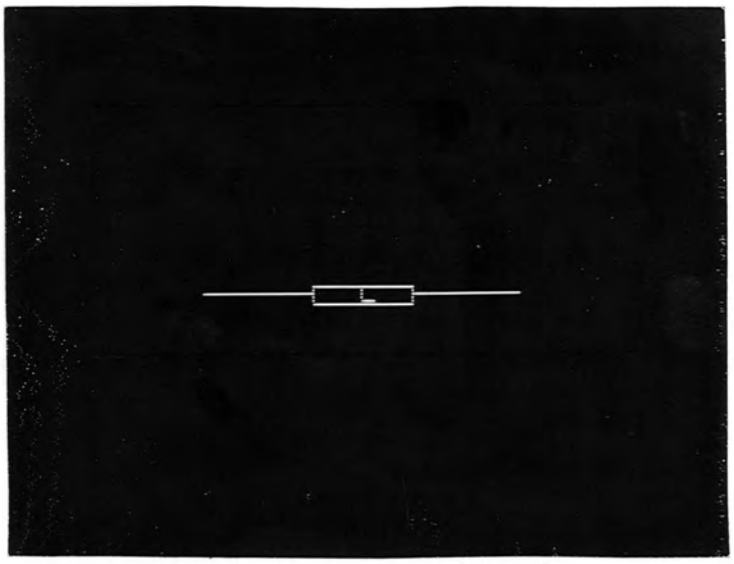
P5.6



P5.7

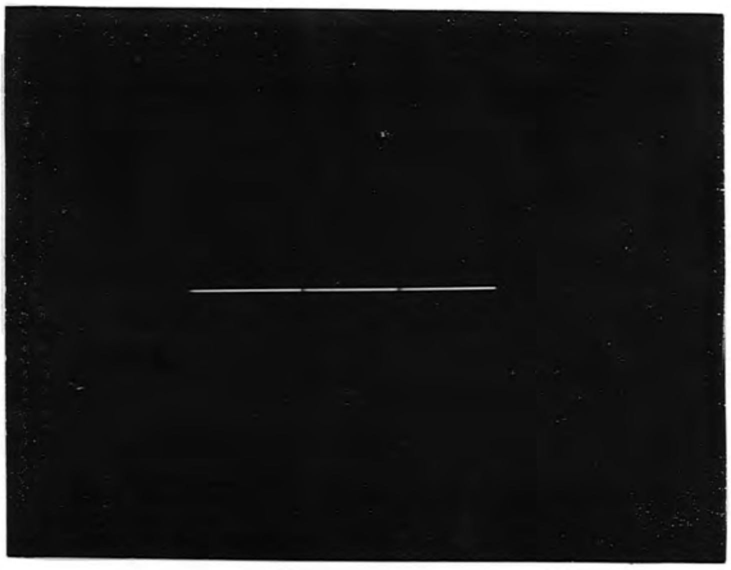


P5.8

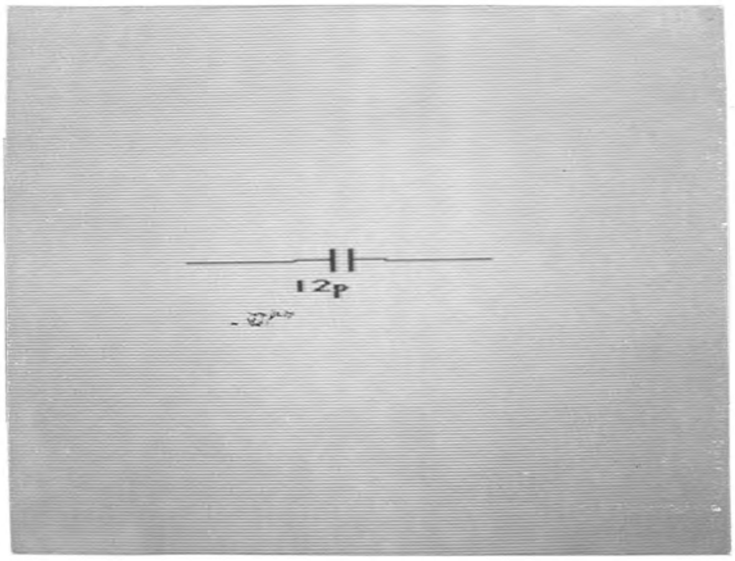


P5.9

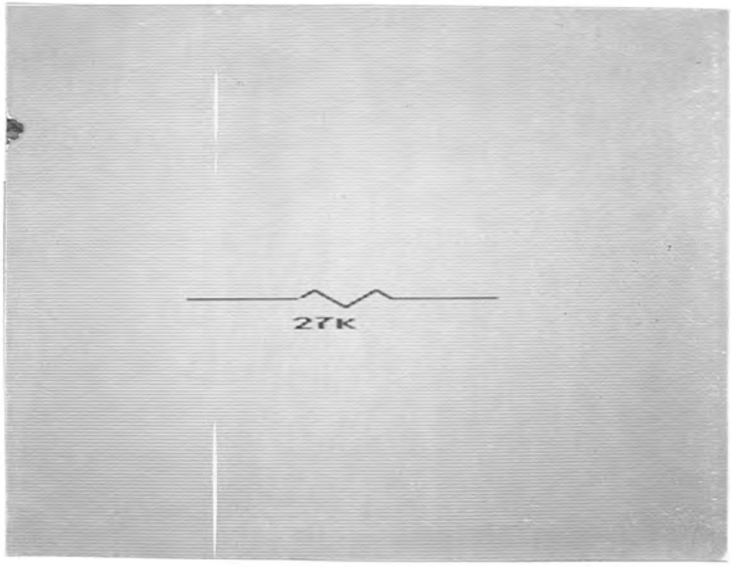
P5.10



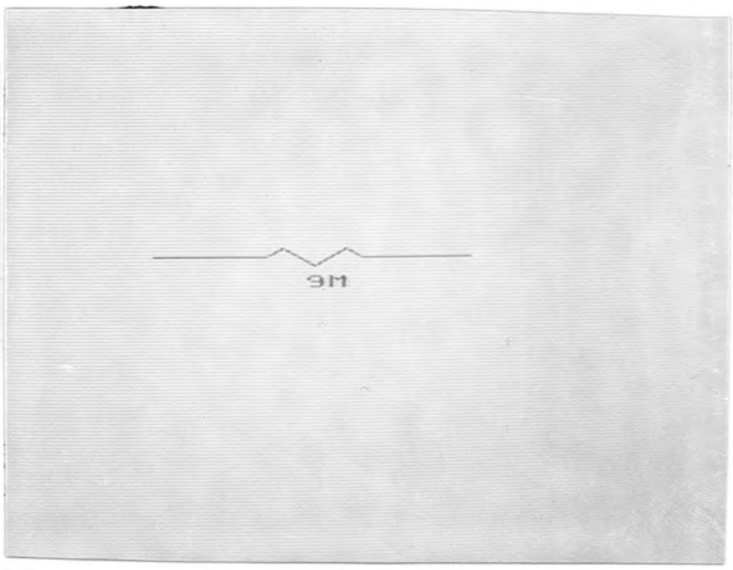
P5.11



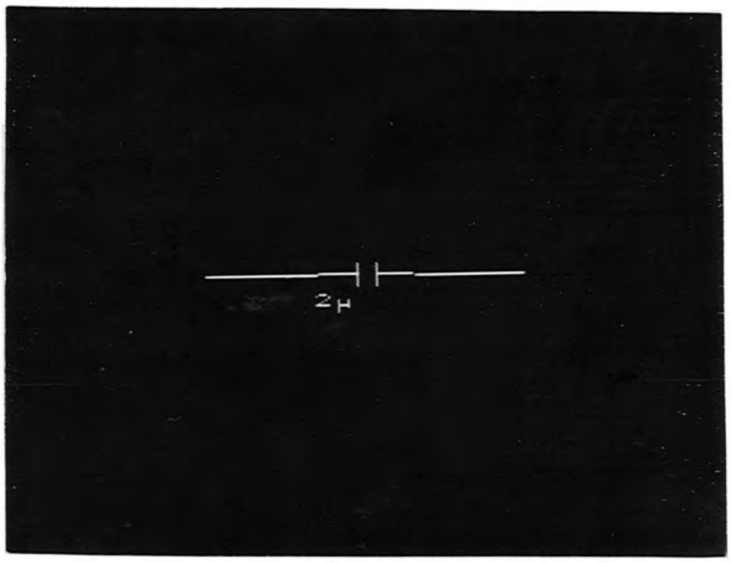
P5.12



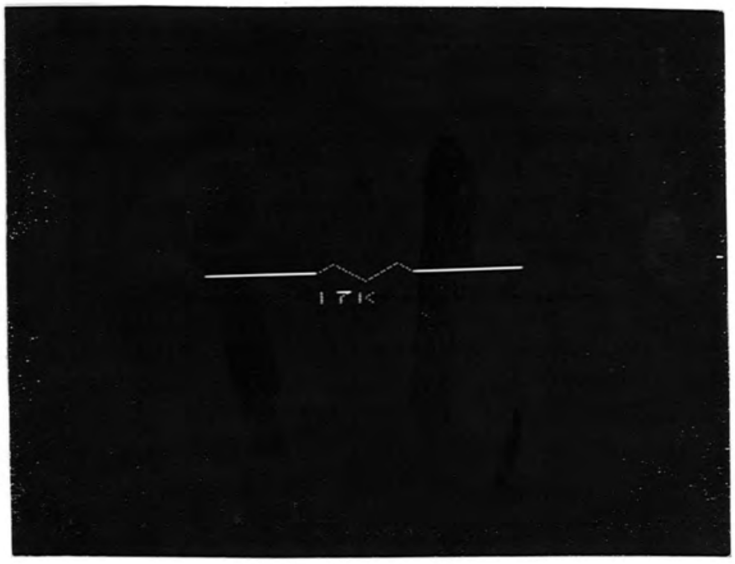
P5.13



P5.14



P5.15



5.2 Project Review

The project was, on the whole, quite successful, as the photographic plates illustrate. Many of the problems experienced resulted from the attempt to run the M.O.S. display store at maximum speed, resulting in this section of the unit being quite sensitive to changes in supply potential. Noise problems were experienced in the rest of the hardware and this is considered to be due to the very high packing density employed. In the later stages of the commissioning contact problems were experienced with the printed circuit board edge connectors, particularly those of the Insertion/Deletion Control and Monitor Control boards, these being the boards that had most often been removed from the rack for modification.

It is believed that the most important development of this project was the design of an image addressing format that allows the display of precise (if simple) designs on a raster scan display without recourse to stores of enormous capacity. Indeed it is one of the major advantages of the technique employed that the system may be operated with only a very small amount of storage, which may be expanded at a later date, perhaps when more funds become available.

Though the particular design described in this thesis is restricted to the display of circuit diagrams, this restriction is only imposed by the design of the software, and thus is easily overcome. It is suggested, in fact, that the use of the microprocessor to actually develop images, under software control, rather than simply organising the transfer of information from symbol store to display store, might be the most interesting use for this type of unit.

As time goes on the cost of semiconductor storage will continue to fall and eventually a point will be reached when there is no economic advantage in using an efficient storage system. However it is apparent that the time has not yet been reached, and that the type of unit described in this thesis could be very useful in situations where some graphic facilities are required but where financial resources are limited.

APPENDIX I

PROGRAM LISTING

The following pages contain a program listing of the software described in Chapter 3. This listing was prepared on a PDP-8E computer using a Cross-Assembler program for the Intel 8008 microprocessor. The first column gives the octal address of the instruction, the next column the instruction in octal, this is followed by the instruction mnemonic.

```
                  DAT 0400

0400    006                 LAI 376   /SHOWS START OF A NON-MODIFYING
0401    376
0402    127                 OUT 3     /SEQUENCE
0403    000       START     HLT
0404    103                 INP 1     /INPUT HORZ OR VERT, UP OR DOWN
0405    012                 RRC       /IDENTIFY HORZ OR VERT COMPONENTS
0406    310                 LBA
0407    100                 JFC VERT
0410    024
0411    001
0412    056                 LHI 004   /START OF HORZ LEADS
0413    004
0414    066                 LLI 000
0415    000
0416    106                 CAL SRT1  /FOR NODE MODIFICATION
0417    366
0420    002
0421    104                 JMP HCOMP /JMPS TO TEST FOR HORZ COMPONENTS
0422    243
0423    001
0424    056       VERT      LHI 006   /START OF VERT LEADS
0425    006
0426    066                 LLI 313
0427    313
0430    106                 CAL SRT1
0431    366
0432    002
0433    000                 HLT
0434    103                 INP 1     /INPUT COMPONENT INFO
0435    036                 LDI 000
0436    000
0437    106                 CAL SRT2  /USED TO MODIFY D IN COMP TEST
0440    035
0441    003
0442    150                 JTZ VRES  /INPUT CODE 001
0443    100
0444    001
0445    106                 CAL SRT2
0446    035
0447    003
0450    150                 JTZ VCAP  /INPUT CODE 002
0451    112
0452    001
0453    106                 CAL SRT2
0454    035
0455    003
0456    150                 JTZ VIND  /INPUT CODE 003
0457    124
0460    001
0461    106                 CAL SRT2
0462    035
0463    003
0464    150                 JTZ VSHORT /INPUT CODE 004
0465    136
0466    001
0467    106                 CAL SRT2
```

```
0470    035
0471    003
0472    150             JTZ VVOLT /INPUT CODE 005
0473    150
0474    001
0475    104             JMP VAMP  /INPUT CODE 006
0476    202
0477    001
0500    056     VRES    LHI 007   /START OF RES MODIFIERS
0501    007
0502    066             LLI 000
0503    000
0504    106             CAL SRT1
0505    366
0506    002
0507    104             JMP CHARIV /TO INPUT 1ST CHARACTER
0510    106
0511    002
0512    056     VCAP    LHI 007   /START OF CAP MODIFIERS
0513    007
0514    066             LLI 063
0515    063
0516    106             CAL SRT1
0517    366
0520    002
0521    104             JMP CHARIV
0522    106
0523    002
0524    056     VIND    LHI 007   /START OF IND MODIFIERS
0525    007
0526    066             LLI 142
0527    142
0530    106             CAL SRT1
0531    366
0532    002
0533    104             JMP CHARIV
0534    106
0535    002
0536    056     VSHORT  LHI 007   /START OF SHORT INFO
0537    007
0540    066             LLI 241
0541    241
0542    106             CAL SRT1
0543    366
0544    002
0545    104             JMP START   /NO CHARACTERS FOR SHORTS
0546    003
0547    001
0550    106     VVOLT   CAL SRT3  /TEST FOR UP OR DOWN
0551    040
0552    003
0553    140             JTC UPVV
0554    170
0555    001
0556    056             LHI 007   /START OF DOWN VOLT SYMBOL
0557    007
```

```
0560   066            LLI 012
0561   012
0562   106            CAL SRT1
0563   366
0564   002
0565   104            JMP COMSV /FOR COMMON PART OF I/V SYMBOLS
0566   231
0567   001
0570   056     UPVV   LHI 012   /START OF UP VOLT SYMBOL
0571   012
0572   066            LLI 333
0573   333
0574   106            CAL SRT1
0575   366
0576   002
0577   104            JMP COMSV
0600   231
0601   001
0602   106     VAMP   CAL SRT3  /TEST FOR UP OR DOWN
0603   040
0604   003
0605   140            JTC UPVA
0606   222
0607   001
0610   056            LHI 012   /START OF CURRENT DOWN SYMBOL
0611   012
0612   066            LLI 340
0613   340
0614   106            CAL SRT1
0615   366
0616   002
0617   104            JMP CHAR1V
0620   106
0621   002
0622   056     UPVA   LHI 012   /START OF CURRENT UP SYMBOL
0623   012
0624   066            LLI 345
0625   345
0626   106            CAL SRT1
0627   366
0630   002
0631   056     COMSV  LHI 012   /START OF COMMON INFO FOR
0632   012
0633   066            LLI 352   /VERTICAL SOURCES
0634   352
0635   106            CAL SRT1
0636   366
0637   002
0640   104            JMP CHAR1V
0641   106
0642   002
0643   000     HCOMP  HLT
0644   103            INP 1     /INPUT HORZ COMPONENT INFO
0645   036            LDI 000   /START OF COMPONENT SEARCH
0646   000
0647   106            CAL SRT2
```

```
0650    035
0651    003
0652    150              JTZ HRES   /INPUT CODE 001
0653    310
0654    001
0655    106              CAL SRT2
0656    035
0657    003
0660    150              JTZ HCAP   /INPUT CODE 002
0661    322
0662    001
0663    106              CAL SRT2
0664    035
0665    003
0666    150              JTZ HIND   /INPUT CODE 003
0667    334
0670    001
0671    106              CAL SRT2
0672    035
0673    003
0674    150              JTZ HSHORT /INPUT CODE 004
0675    346
0676    001
0677    106              CAL SRT2
0700    035
0701    003
0702    150              JTZ HVOLT /INPUT CODE 005
0703    360
0704    001
0705    104              JMP HAMP   /INPUT CODE 006
0706    012
0707    002
0710    056      HRES    LHI 004    /START OF RES MODIFIERS
0711    004
0712    066              LLI 011
0713    011
0714    106              CAL SRT1
0715    366
0716    002
0717    104              JMP CHARIN /TO INPUT 1ST CHARACTER
0720    050
0721    002
0722    056      HCAP    LHI 004    /START OF CAP MODIFIERS
0723    004
0724    066              LLI 104
0725    104
0726    106              CAL SRT1
0727    366
0730    002
0731    104              JMP CHARIN
0732    050
0733    002
0734    056      HIND    LHI 006    /START OF IND MODIFIERS
0735    006
0736    066              LLI 222
0737    222
```

```
0740    106                CAL SRT1
0741    366
0742    002
0743    104                JMP CHAR1H
0744    050
0745    002
0746    056      HSHORT    LHI 004    /START OF SHORT INFO
0747    004
0750    066                LLI 372
0751    372
0752    106                CAL SRT1
0753    366
0754    002
0755    104                JMP START  /NO CHARACTERS FOR SHORTS
0756    003
0757    001
0760    106      HVOLT     CAL SRT3  /TEST FOR UP OR DOWN
0761    040
0762    003
0763    140                JTC UPHV
0764    000
0765    002
0766    056                LHI 004    /START OF HORZ VOLT DOWN SYMBOL
0767    004
0770    066                LLI 355
0771    355
0772    106                CAL SRT1
0773    366
0774    002
0775    104                JMP COMSH /FOR COMMON PART OF I/V SYMBOLS
0776    041
0777    002
1000    056      UPHV      LHI 004    /START OF UP VOLT SYMBOL
1001    004
1002    066                LLI 317
1003    317
1004    106                CAL SRT1
1005    366
1006    002
1007    104                JMP COMSH
1010    041
1011    002
1012    106      HAMP      CAL SRT3
1013    040
1014    003
1015    140                JTC UPHA
1016    032
1017    002
1020    056                LHI 004    /START OF CURRENT DOWN SYMBOL
1021    004
1022    066                LLI 334
1023    334
1024    106                CAL SRT1
1025    366
1026    002
1027    104                JMP COMSH
```

```
1030   041
1031   002
1032   056        UPHA      LHI 004    /START OF CURRENT UP SYMBOL
1033   004
1034   066                  LLI 276
1035   276
1036   106                  CAL SRT1
1037   366
1040   002
1041   056        COMSH     LHI 004    /START OF COMMON INFO FOR
1042   004
1043   066                  LLI 201    /HORIZONTAL COMPONENTS
1044   201
1045   106                  CAL SRT1
1046   366
1047   002
1050   000        CHAR1H    HLT
1051   103                  INP 1      /TO INPUT 1ST CHARACTER
1052   310                  LBA
1053   006                  LAI 335
1054   335
1055   123                  OUT 1
1056   006                  LAI 366
1057   366
1060   125                  OUT 2      /TO MODIFY BASIC DATUM POSITION
1061   006                  LAI 375    /CODE 375 ON OUT 3 SHOWS THAT OUT 1
1062   375
1063   127                  OUT 3      /& OUT 2 INFO IS TO PERMANENTLY
1064   000                  HLT        /BASIC DATUM. INT SHOWS THAT MOD.
1065   006                  LAI 376    /HAS BEEN ACCEPTED. CHANGE CODE ON
1066   376
1067   127                  OUT 3      /OUT 3
1070   301                  LAB
1071   260                  ORA
1072   150                  JTZ CHAR2 /IF 1ST CHARACTER IS ZERO
1073   141
1074   002
1075   106                  CAL SRT4  /SEARCH FOR CHARACTER
1076   043
1077   003
1100   106                  CAL SRT5  /TO MODIFY BASIC DATUM FOR NEXT
1101   233
1102   003
1103   104                  JMP CHAR2 /CHARACTER
1104   141
1105   002
1106   000        CHAR1V    HLT
1107   103                  INP 1      /INPUT 1ST CHARACTER
1110   310                  LBA
1111   006                  LAI 363
1112   363
1113   123                  OUT 1
1114   006                  LAI 363
1115   363
1116   125                  OUT 2
1117   006                  LAI 375
```

```
1120    375
1121    127                 OUT 3
1122    000                 HLT
1123    006                 LAI 376
1124    376
1125    127                 OUT 3
1126    301                 LAB
1127    260                 ORA
1130    150                 JTZ CHAR2 /IF 1ST CHARACTER IS ZERO
1131    141
1132    002
1133    106                 CAL SRT4
1134    043
1135    003
1136    106                 CAL SRT5
1137    233
1140    003
1141    000         CHAR2   HLT
1142    103                 INP 1     /INPUT 2ND CHARACTER
1143    260                 ORA
1144    150                 JTZ BLANK /TEST FOR 2ND CHAR. OF ZERO  AFTER
1145    160
1146    002
1147    106                 CAL SRT4  /1ST CHAR. OF ZERO
1150    043
1151    003
1152    106                 CAL SRT5
1153    233
1154    003
1155    104                 JMP CHAR3
1156    176
1157    002
1160    271         BLANK   CPB
1161    150                 JTZ CHAR3
1162    176
1163    002
1164    056                 LHI 005   /START OF ZERO CHARACTER
1165    005
1166    066                 LLI 000
1167    000
1170    106                 CAL SRT1
1171    366
1172    002
1173    106                 CAL SRT5  /TO MODIFY BASIC DATUM FOR NEXT
1174    233
1175    003
1176    000         CHAR3   HLT       /CHARACTER
1177    103                 INP 1     /INPUT 3RD CHARACTER
1200    260                 ORA
1201    110                 JFZ NOT0  /IS 3RD CHARACTER ZERO
1202    216
1203    002
1204    056                 LHI 005   /START OF ZERO CHARACTER
1205    005
1206    066                 LLI 000
1207    000
```

```
1210    106                 CAL SRT1
1211    366
1212    002
1213    104                 JMP  CHAR4
1214    221
1215    002
1216    106       NOT0      CAL SRT4  /SEARCH FOR CHARACTER
1217    043
1220    003
1221    000       CHAR4     HLT
1222    103                 INP 1     /INPUT 4TH CHARACTER-THE EXPONENT
1223    106                 CAL SRT5  /TO PERMANENTLY MODIFY THE BASIC
1224    233
1225    003
1226    036                 LDI 000   /DATUM
1227    000
1230    273                 CPD
1231    150                 JTZ START  /JUMP TO END OF PROG. IF
1232    003
1233    001
1234    106                 CAL SRT2  /EXP.=0, NO MODS. REQD.
1235    035
1236    003
1237    150                 JTZ KILO  /IF EXP=+3, CODE 001
1240    304
1241    002
1242    106                 CAL SRT2
1243    035
1244    003
1245    150                 JTZ MEGA  /IF EXP=+6, CODE 002
1246    316
1247    002
1250    106                 CAL SRT2
1251    035
1252    003
1253    150                 JTZ MILLI /IF EXP=-3, CODE 003
1254    330
1255    002
1256    106                 CAL SRT2
1257    035
1260    003
1261    150                 JTZ MICRO /IF EXP=-6, CODE 004
1262    342
1263    002
1264    106                 CAL SRT2
1265    035
1266    003
1267    150                 JTZ NANO  /IF EXP=-9, CODE 005
1270    354
1271    002
1272    056                 LHI 006   /START OF PICO CHARACTER, IF EXP=
1273    006
1274    066                 LLI 171   /-12, CODE 006
1275    171
1276    106                 CAL SRT1
1277    366
```

```
1300    002
1301    104                 JMP START
1302    003
1303    001
1304    056       KILO      LHI 006    /START OF KILO CHAR.
1305    006
1306    066                 LLI 000
1307    000
1310    106                 CAL SRT1
1311    366
1312    002
1313    104                 JMP START
1314    003
1315    001
1316    056       MEGA      LHI 006    /START OF MEGA CHAR.
1317    006
1320    066                 LLI 031
1321    031
1322    106                 CAL SRT1
1323    366
1324    002
1325    104                 JMP START
1326    003
1327    001
1330    056       MILLI     LHI 006    /START OF MILLI CHAR.
1331    006
1332    066                 LLI 072
1333    072
1334    106                 CAL SRT1
1335    366
1336    002
1337    104                 JMP START
1340    003
1341    001
1342    056       MICRO     LHI 006    /START OF MICRO CHAR.
1343    006
1344    066                 LLI 136
1345    136
1346    106                 CAL SRT1
1347    366
1350    002
1351    104                 JMP START
1352    003
1353    001
1354    056       NANO      LHI 006    /START OF NANO CHAR.
1355    006
1356    066                 LLI 117
1357    117
1360    106                 CAL SRT1
1361    366
1362    002
1363    104                 JMP START
1364    003
1365    001
1366    327       SRT1      LCM       /FETCH NO. OF COUNTS=HALF NO.OF
1367    036       INSERT    LDI 000    /STORE POSITIONS. D REG. USED TO
```

```
1370    000
1371    006        TEST     LAI 377    /SHOW WHETHER WE ARE DEALING WITH
1372    377
1373    276                 CPL        /A LINE COUNT MODIFIER, D=200 USE
1374    110                 JFZ CHANGE /OUT2, OR AN INTRA-LINE MODIFIER
1375    000
1376    003
1377    050                 INH        /D=000 USE OUT1. INC H REG. IF
1400    060        CHANGE   INL        /L REG IS AT ITS LIMIT OF 377
1401    303                 LAD
1402    002                 RLC        /PUT MSB OF D IN CARRY POSTN.
1403    140                 JTC OP2    /IF INFO IS LINE MODIFIER
1404    015
1405    003
1406    307                 LAM        /FETCH MODIFIER FROM STORE
1407    123                 OUT 1      /OUT1=INTRA-LINE MODIFIER & FLAG
1410    036                 LDI 200
1411    200
1412    104                 JMP TEST
1413    371
1414    002
1415    307        OP2      LAM
1416    125                 OUT 2      /OUT2=LINE MODIFIER & FIELD BIT
1417    006                 LAI 373
1420    373
1421    127                 OUT 3      /TO SHOW THIS IS A NON-PERMANENT
1422    000                 HLT        /MODIFICATION OF THE DATUM. INT.
1423    021                 DCC        /SHOWS MODIFIER HAS BEEN PROCESSED
1424    302                 LAC
1425    260                 ORA
1426    110                 JFZ INSERT /HAVE ALL MODIFIERS BEEN ACTIONED?
1427    367
1430    002
1431    006                 LAI 376
1432    376
1433    127                 OUT 3     /CHANGE IN OUT 3 CODE SHOWS
1434    007                 RET        /MODIFYING SEQUENCE IS AT AN END
1435    030        SRT2     IND        /USED TO INCREMENT D WHEN SEARCHING
1436    273                 CPD        /FOR CHARACTERS
1437    007                 RET
1440    301        SRT3     LAB        /USED TO DETERMINE IF SYMBOL IS UP
1441    012                 RRC        /OR DOWN
1442    007                 RET
1443    036        SRT4     LDI 001    /CHARACTER SEARCH ROUTINE
1444    001
1445    273                 CPD
1446    150                 JTZ ONE    /FOR 1 SYMBOL
1447    133
1450    003
1451    106                 CAL SRT2
1452    035
1453    003
1454    150                 JTZ TWO    /FOR 2 SYMBOL
1455    143
1456    003
1457    106                 CAL SRT2
```

```
1460    035
1461    003
1462    150                     JTZ THREE /FOR 3 SYMBOL
1463    153
1464    003
1465    106                     CAL SRT2
1466    035
1467    003
1470    150                     JTZ FOUR  /FOR 4 SYMBOL
1471    163
1472    003
1473    106                     CAL SRT2
1474    035
1475    003
1476    150                     JTZ FIVE  /FOR 5 SYMBOL
1477    173
1500    003
1501    106                     CAL SRT2
1502    035
1503    003
1504    150                     JTZ SIX   /FOR 6 SYMBOL
1505    203
1506    003
1507    106                     CAL SRT2
1510    035
1511    003
1512    150                     JTZ SEVEN  /FOR 7 SYMBOL
1513    213
1514    003
1515    106                     CAL SRT2
1516    035
1517    003
1520    150                     JTZ EIGHT /FOR 8 SYMBOL
1521    223
1522    003
1523    056                     LHI 005   /START OF 9 SYMBOL
1524    005
1525    066                     LLI 315
1526    315
1527    106                     CAL SRT1
1530    366
1531    002
1532    007                     RET
1533    056             ONE     LHI 005  /LISTING OF START ADDRESSES FROM 1-9
1534    005
1535    066                     LLI 035
1536    035
1537    106                     CAL SRT1
1540    366
1541    002
1542    007                     RET
1543    056             TWO     LHI 005
1544    005
1545    066                     LLI 054
1546    054
1547    106                     CAL SRT1
```

```
1550    366
1551    002
1552    007                 RET
1553    056       THREE     LHI 005
1554    005
1555    066                 LLI 101
1556    101
1557    106                 CAL SRT1
1560    366
1561    002
1562    007                 RET
1563    056       FOUR      LHI 005
1564    005
1565    066                 LLI 130
1566    130
1567    106                 CAL SRT1
1570    366
1571    002
1572    007                 RET
1573    056       FIVE      LHI 005
1574    005
1575    066                 LLI 155
1576    155
1577    106                 CAL SRT1
1600    366
1601    002
1602    007                 RET
1603    056       SIX       LHI 005
1604    005
1605    066                 LLI 204
1606    204
1607    106                 CAL SRT1
1610    366
1611    002
1612    007                 RET
1613    056       SEVEN     LHI 005
1614    005
1615    066                 LLI 237
1616    237
1617    106                 CAL SRT1
1620    366
1621    002
1622    007                 RET
1623    056       EIGHT     LHI 005
1624    005
1625    066                 LLI 260
1626    260
1627    106                 CAL SRT1
1630    366
1631    002
1632    007                 RET
1633    340       SRTS      LEA           /USED FOR PERMANENT MODIFICATION
1634    006                 LAI 370       /OF DATUM FOR 2ND 3RD & 4TH CHARS.
1635    370
1636    123                 OUT 1
1637    006                 LAI 377
```

```
1640    377
1641    125                 OUT 2
1642    006                 LAI 375
1643    375
1644    127                 OUT 3
1645    000                 HLT        /INT SHOWS MODIFICATION HAS
1646    006                 LAI 376    /BEEN PROCESSED
1647    376
1650    127                 OUT 3      /CHANGE IN OUT3 CODE SHOWS END OF
1651    304                 LAE        /MODIFYING OPERATION
1652    260                 ORA
1653    007                 RET
            END
```

APPENDIX II

FORMATION OF ADDRESS MODIFIERS

For the display of a given symbol image, addresses are formed by the addition of address modifiers to the base address. The derivation of address modifiers for a particular symbol, the horizontal inductor shown in Plate P5.4, is described here in order to illustrate the technique employed.

Each complete circuit symbol consists of two major components; the leads, which are common to all horizontal (or vertical) symbols; and the particular design which is special to each symbol. The technique of deriving the correct address modifiers is the same for both components.

Each address modifier contains information as to:-

(i) The horizontal deflection (number of sections) of the point of interest from the base address.

This occupies bits 0-6 of OUT1 in the address modifier.

(ii) Whether the intensity change at the point of interest is "temporary" or "permanent".

This defines the status of the flag bit, OUT1, bit 7.

(iii) The vertical deflection (number of lines) of the point of interest from the base address.

This occupies bits 0-6 of OUT2.

(iv) The field in which the point of interest lies.

This defines the field bit, OUT2, bit 7.

The OUT1 port is filled before the OUT2 port and thus it is the OUT1 information which appears first in the symbol store. The general structure of the information in the symbol store is:-

Flag Bit: Horizontal Deflection (First Word)	Field Bit: Vertical Deflection (Second Word)

If we consider the common lead symbol as shown in figure AII.1 then we see that four address modifiers are required to define this image. These are:-

(i) The start of the first lead.

This occurs at a vertical deflection of three lines in the second field, this means that the field bit will be set. As the intensity change is "permanent" (i.e. greater than a single line section) the flag bit is not set. Thus the total address modifier will be:-

0:0 1:3

Or written as two eight bit words for OUT1 and OUT2, coded in octal (with the grouping XX XXX XXX):-

000 203

(ii) The end of the first lead.

Horizontal Deflection	43_8 sections.	
Change Permanent	Flag bit 0.	
Vertical Deflection	3 lines.	
Second Field	Field bit 1.	
Base Address Modifier	0:43	1:3
or	043	203

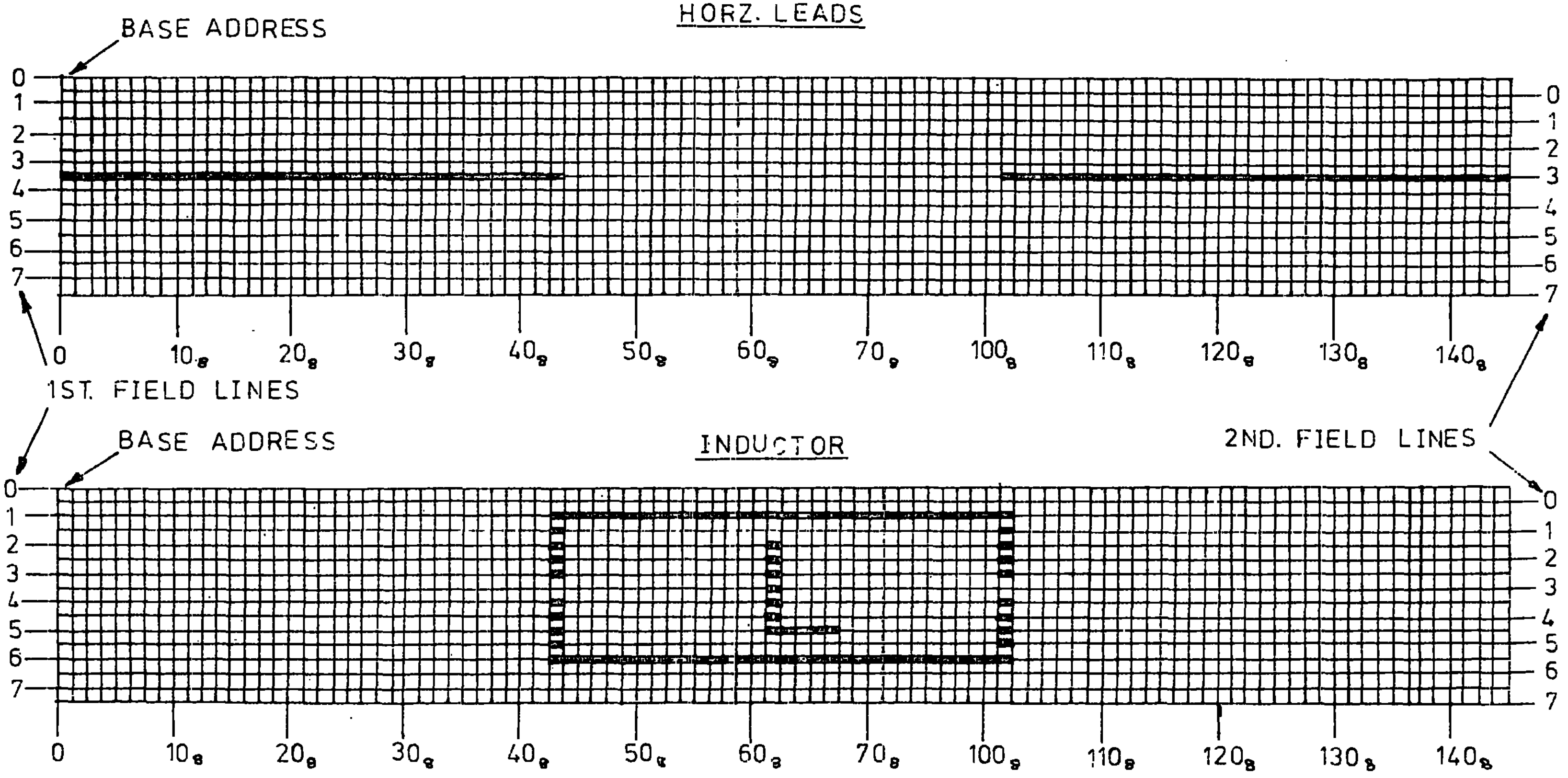


FIG A II.1 SYMBOL CODING

(iii) The start of the second lead.

Horizontal Deflection	101_8 sections.	
Change Permanent	Flag bit 0.	
Vertical Deflection	3 lines.	
Second Field	Field bit 1.	
Base Address Modifier	0:101	1:3
or	101	203

(iv) The end of the second lead.

Horizontal Deflection	144_8 sections.	
Change Permanent	Flag bit 0.	
Vertical Deflection	3 lines.	
Second Field	Field bit 1.	
Base Address Modifier	0:144	1:3
or	144	203

Thus the base address modifiers for the display of horizontal leads are: 000 203 043 203 101 203 144 203.

However the output latches of the SIM-8 microcomputer have the effect of inverting the signal level and thus, in order to avoid the expense of re-inverting the information, the address modifiers are stored in inverted form. This means that the above modifiers are stored as:-

377 174 334 174 276 174 233 174.

The software requires that the first information in a modifier block should give the number of modifiers in that block (in this case four). Thus the total address modifier block for horizontal leads becomes:-

004 377 174 334 174 276 233 174.

The address modifiers for the part of the image relating specifically to the inductor symbol are developed in exactly the same manner as were those for the common lead display. For instance the address modifier for the start of the first line of the inductor symbol defines a horizontal deflection of 42_8 sections and a vertical deflection of one line in the first field. The change is "permanent" and thus the flag bit is not set. This results in the address modifier:-

	0:42	0:1
or	042	001

Stored in inverted form as 335 376.

The complete listing for the inductor symbol is shown below.

MODIFIER		OCTAL CODING		INVERTED FORM	
0: 42	0:1	042	001	335	376
0:102	0:1	101	001	275	376
1: 42	0:2	242	002	135	375
1: 61	0:2	261	002	116	375
1:101	0:2	301	002	076	375
1: 42	0:3	242	003	135	374
1: 61	0:3	261	003	116	374
1:101	0:3	301	003	076	374
1: 42	0:4	242	004	135	373
1: 61	0:4	261	004	116	373
1:101	0:4	301	004	076	373
1: 42	0:5	242	005	135	372
0: 61	0:5	061	005	316	372
0: 66	0:5	066	005	311	372
1:101	0:5	301	005	076	372
0: 42	0:6	042	006	335	371
0:102	0:6	102	006	275	371
1: 42	1:1	242	201	135	176

MODIFIER		OCTAL CODING		INVERTED FORM	
1:101	1:1	301	201	076	176
1: 42	1:2	242	202	135	175
1: 61	1:2	261	202	116	175
1:101	1:2	301	202	076	175
1: 61	1:3	261	203	116	174
1: 42	1:4	242	204	135	173
1: 61	1:4	261	204	116	173
1:101	1:4	301	204	076	173
1: 42	1:5	242	205	135	172
1:101	1:5	301	205	076	172

As there are twenty-eight address modifiers in this block, the complete block will be preceded in store by the code 034 ($34_8 = 28_{10}$).

APPENDIX III

INTERFACE SIGNALS

Listed below are the signals necessary to interface the V.D.U. to the controlling computer. In all cases signals are at TTL levels.

SIGNAL (Level when true)	BOARD & PIN NO.	COMMENTS
ERASE (L)	Insert./Delete Control, A9	Clears the display. This should be the first command sent after switching on.
INSERT (L)	Insert./Delete Control, A5	Shows that the information presented by the microprocessor is to be fed into the display store.
DELETE (L)	Insert./Delete Control, B20	Shows that the information equivalent to that presented by the microprocessor is to be removed from the display store.
CLEAR BASE ADDRESS (H)	Adder Unit, B9	Sets the Base Address Register to zero prior to a new base address. Must be removed before enabling the BASE ADD. LOAD signal.
BASE ADDRESS LOAD (L)	Adder Unit, B14	Feeds the new base add. into the Base Address Register. Should be preceded by a CLEAR BASE ADD. signal.

SIGNAL (Level when true)	BOARD & PIN NO.	COMMENTS
BASE ADDRESS BIT 0	Adder Unit, B15	The status of these signals during a BASE ADD. LOAD defines the contents of the Base Address Register.
BIT 1	B16	
BIT 2	B17	
BIT 3	B18	
BIT 4	B19	
BIT 5	B20	
BIT 6	B21	
BIT 7	B22	
BIT 8	B23	
BIT 9	B32	
BIT 10	B33	
BIT 11	B34	
BIT 12	B35	
BIT 13	B36	
BIT 14	B37	
BIT 15	B38	
BIT 16	B39	
INT/(MP) (L)	Adder Unit, A3	Interrupts the microprocessor in order to present symbol and character code information.

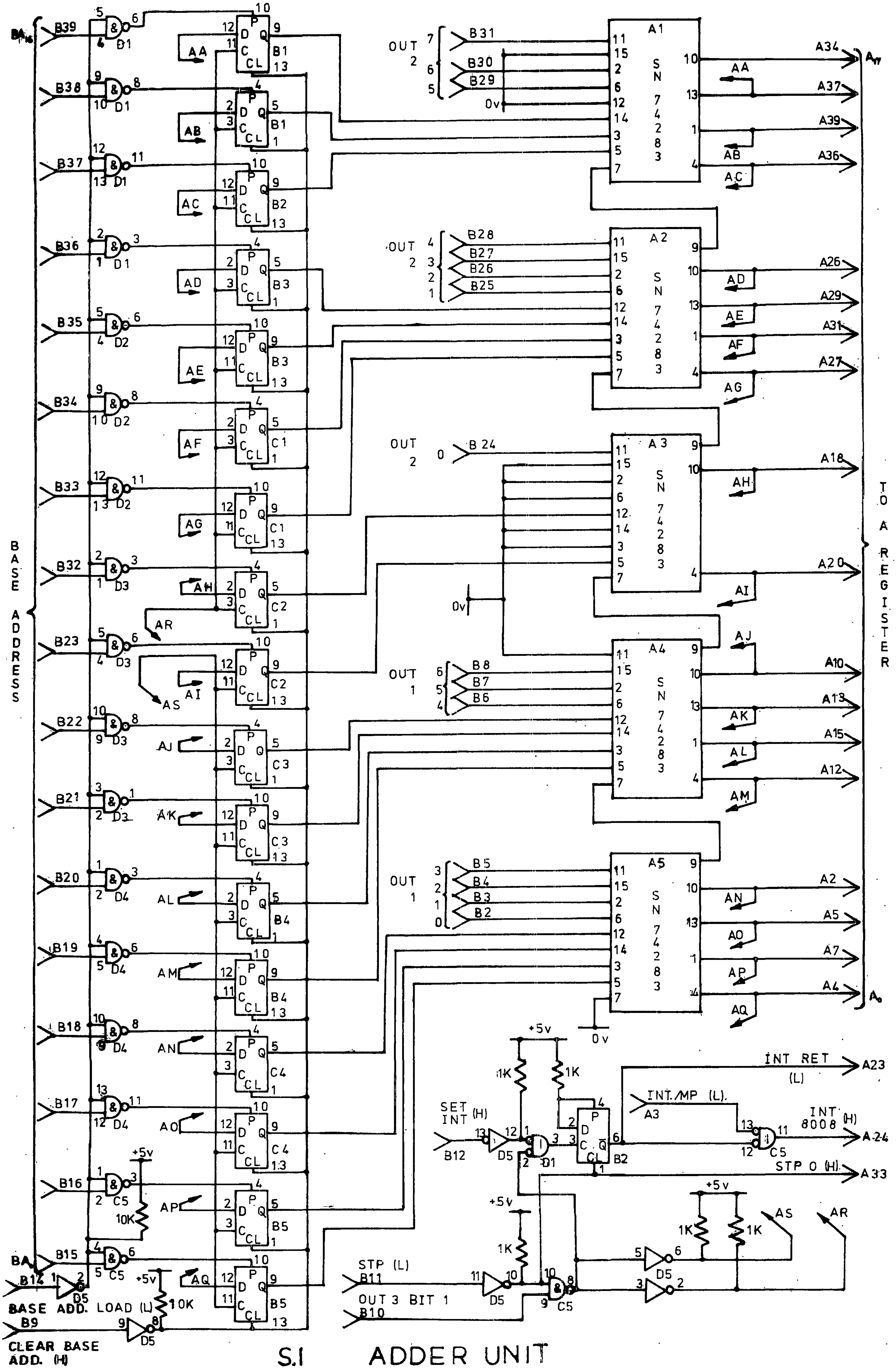
APPENDIX IV

SCHEMATIC DIAGRAMS

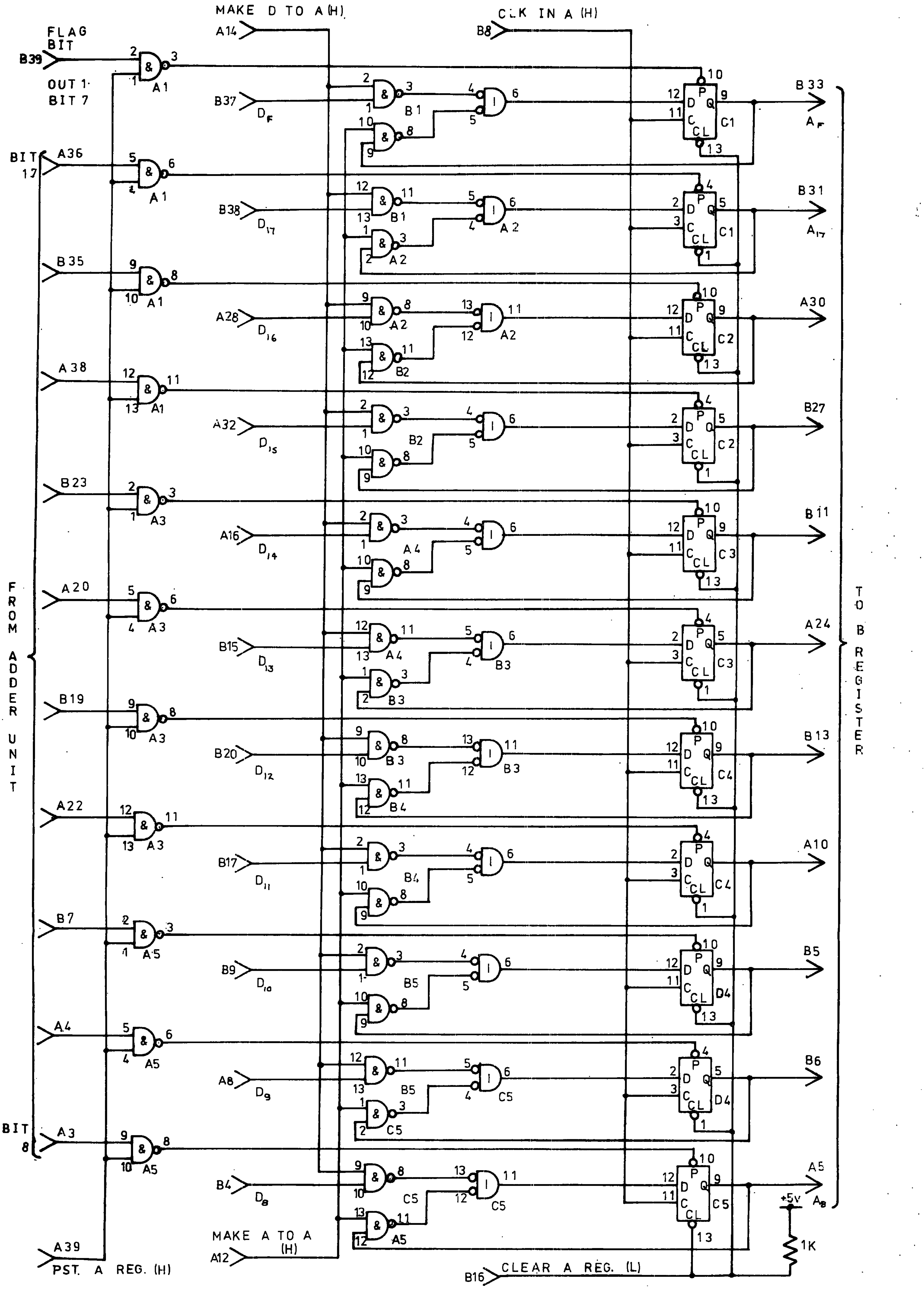
Each of the following schematic diagrams refers to a single printed circuit board. Integrated circuits (I.C.s) are mounted on the boards in four columns, A, B, C and D, column A being the nearest column to the edge connections. On most boards, each column is five I.C.s high - the exception to this is the Magnitude Comparator board, where each column has only four I.C.s. A particular contact may be defined by quoting the column letter A, B, C or D; the row number 1, 2, 3, 4, or 5; and the I.C. pin number. Thus C4-6 refers to pin 6 of the I.C. which is the fourth down in the third column from the edge connector. In two cases during commissioning it was found to be necessary to introduce extra I.C.s between the standard columns, these devices are referred to by the prefix X.

There are forty edge contacts on each side of a printed circuit board and any individual contact is referred to by a letter, A for the non-component side of the board, B for the component side of the board, and a figure. Thus B14 is the fourteenth finger on the component side of the board. In all cases, save that of the main display store board, pins A40 and B40 are used for the 0v line, and pins A1 and B1 for the 5v line. On the display store board however, pin B1 is used for the extra supply necessary to run the MOS devices at the required speed.

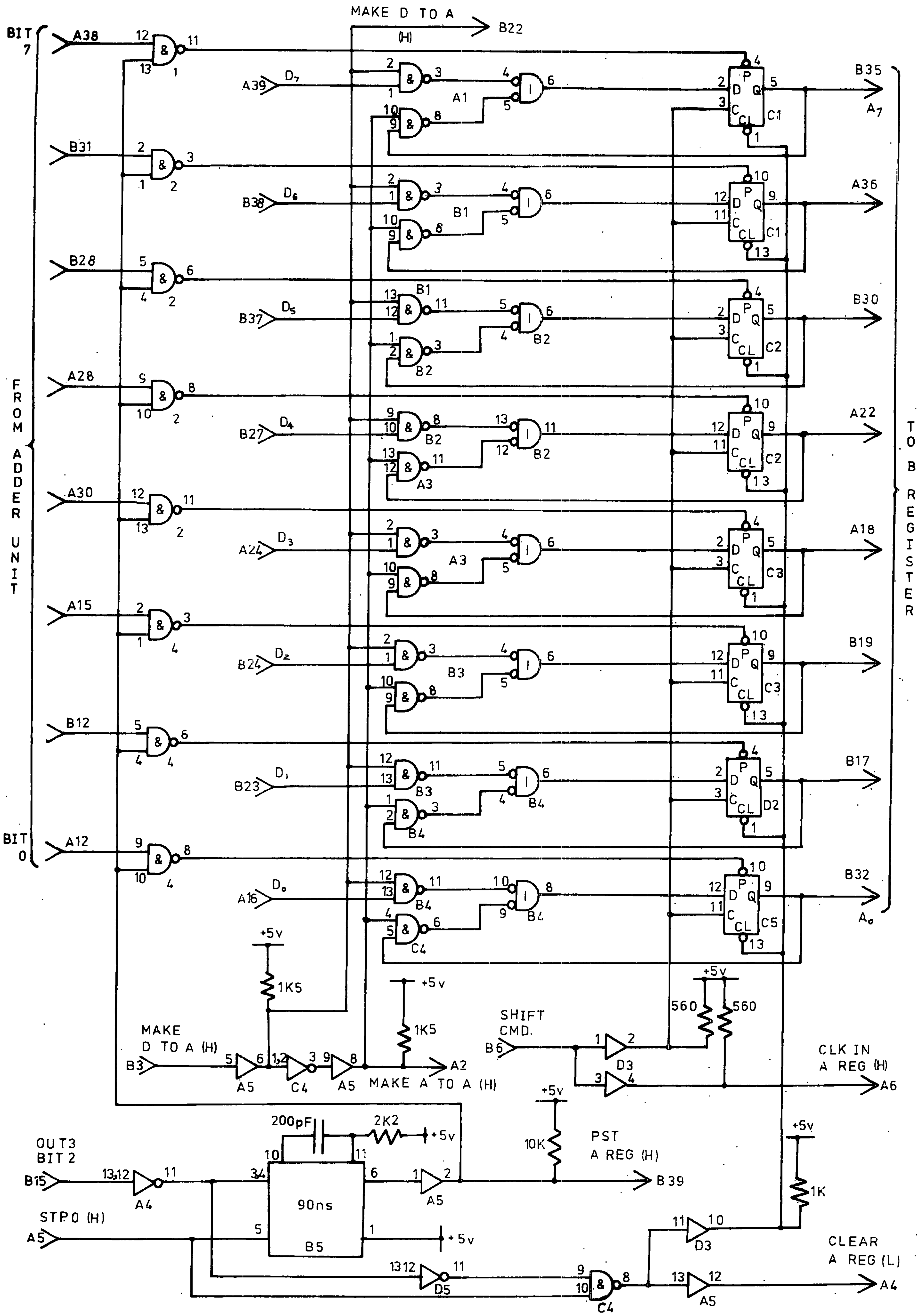
In order to minimise the effects of noise on the power supply lines, each board has a distributed capacitance of 20 μF between the 5v and 0v lines. However noise still caused problems during the commissioning of the equipment, this was considered to be due to the high packing density employed.



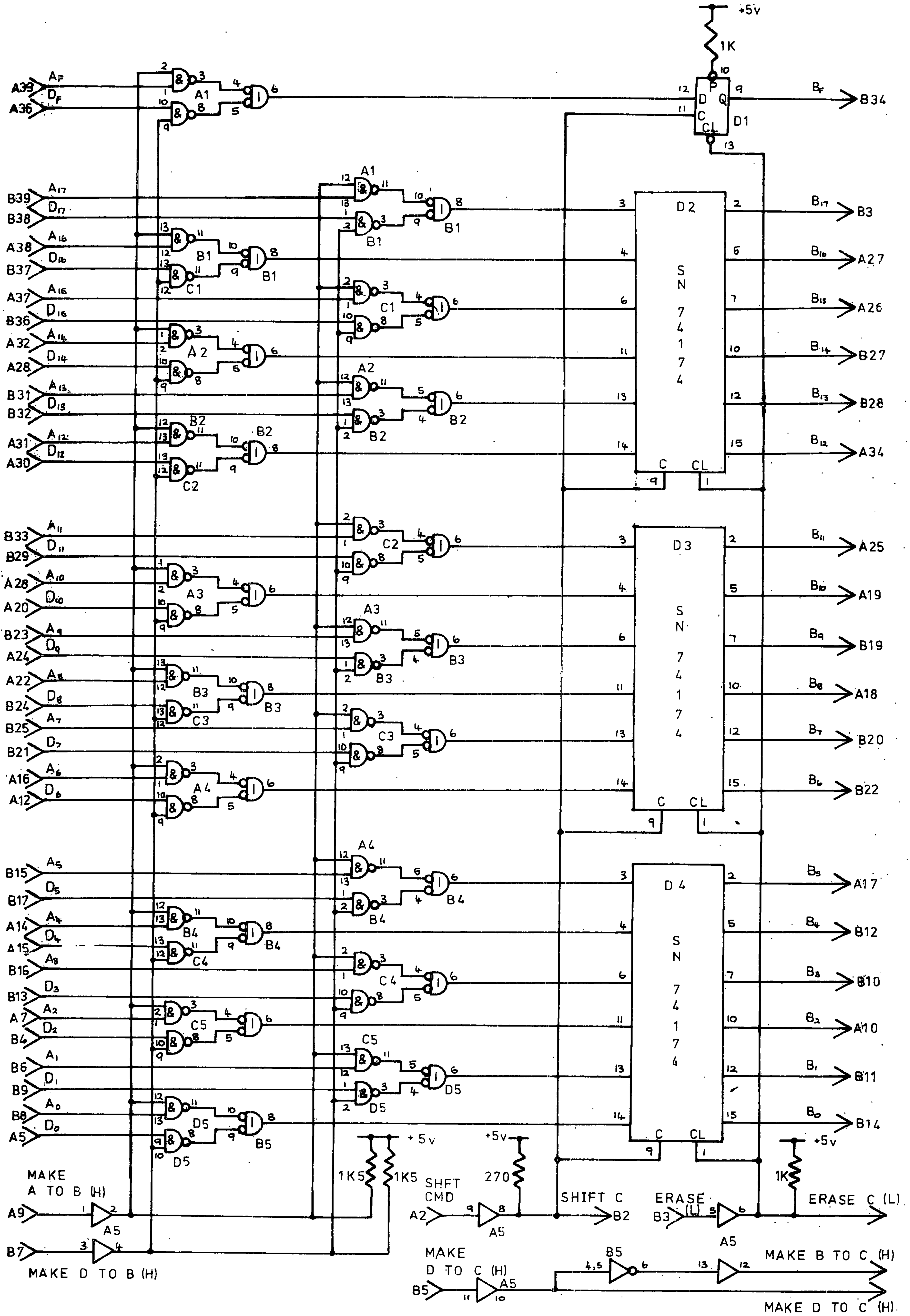
S.1 ADDER UNIT



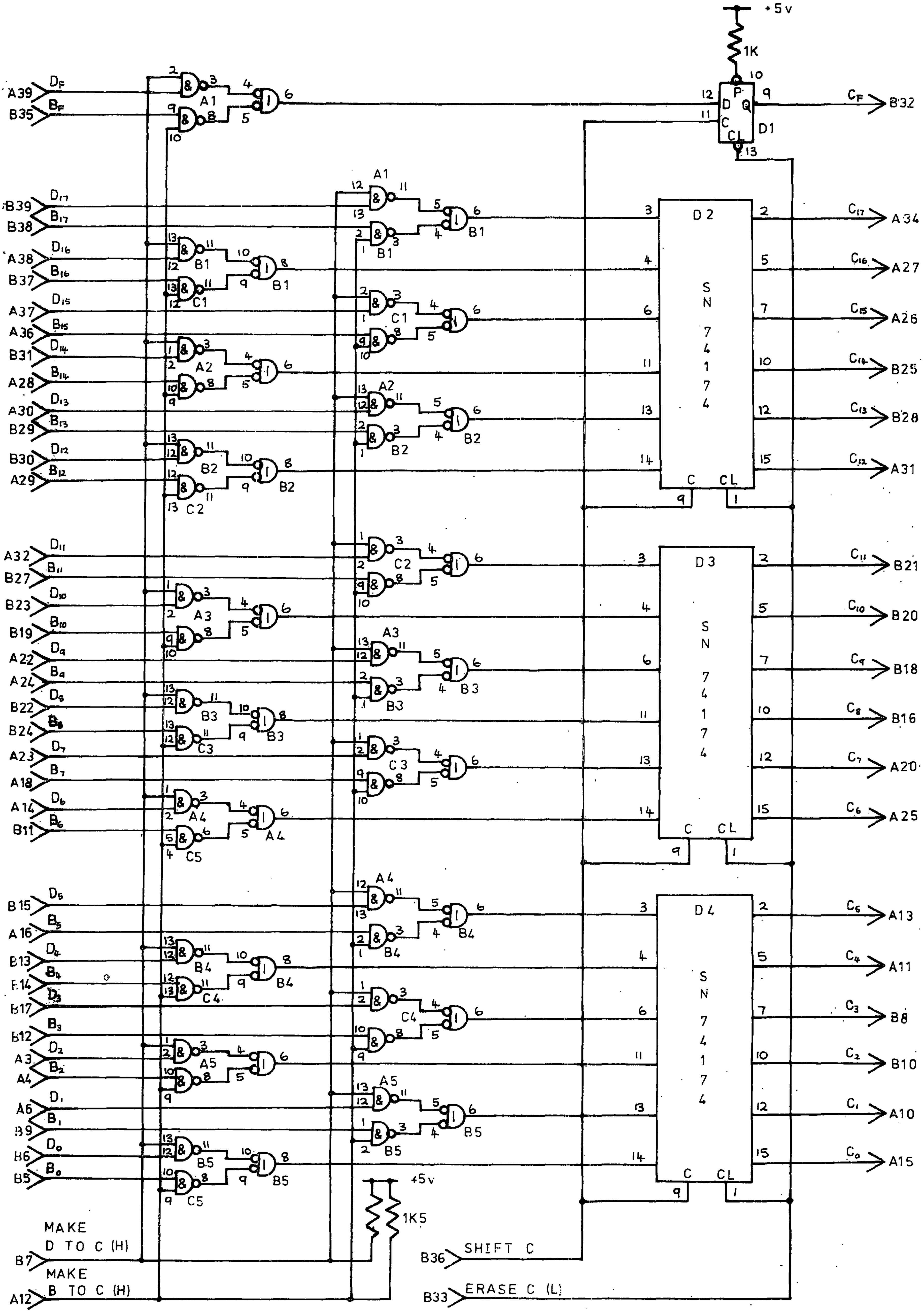
S.2 A REGISTER PART I



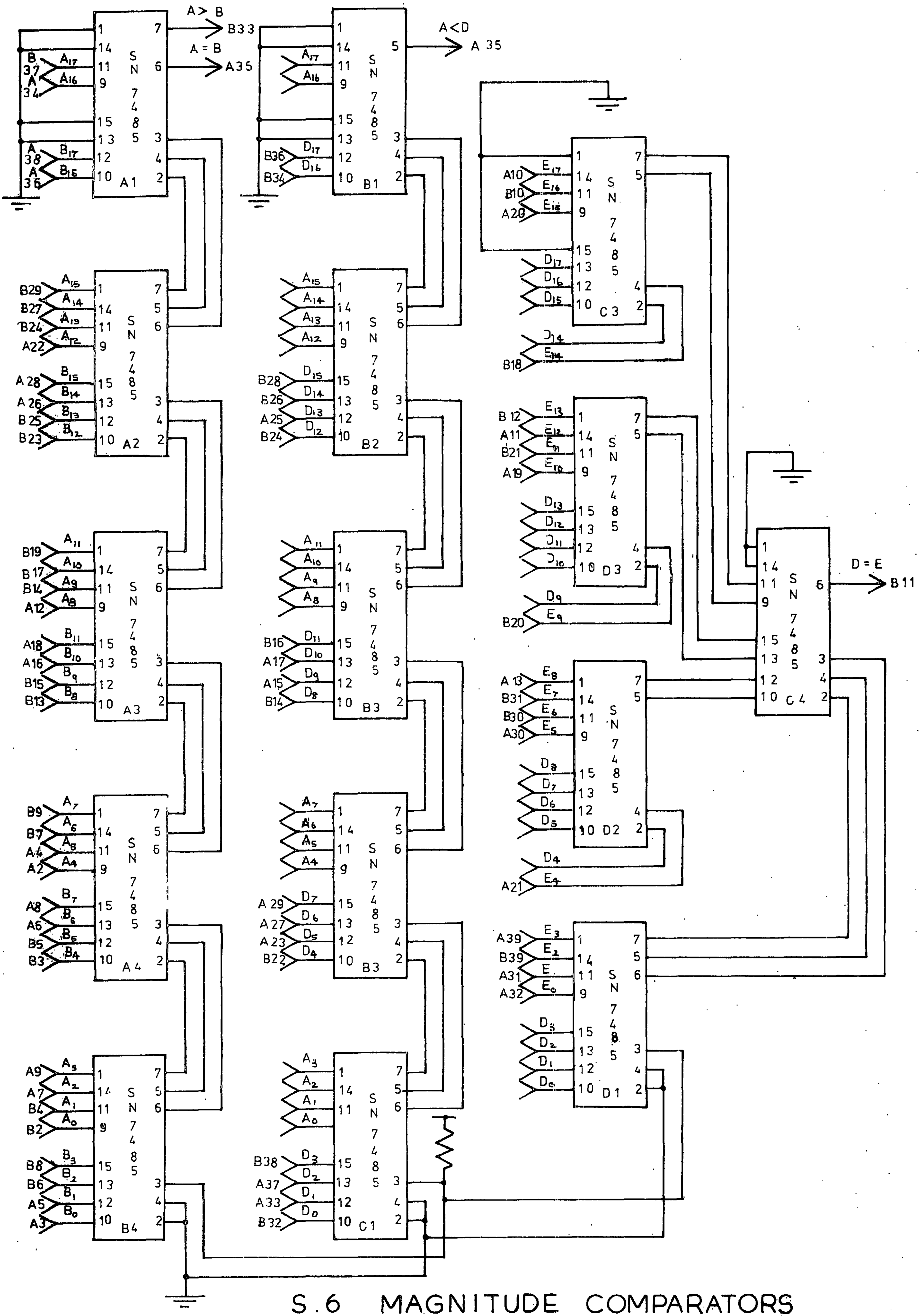
S.3 A REGISTER PART II



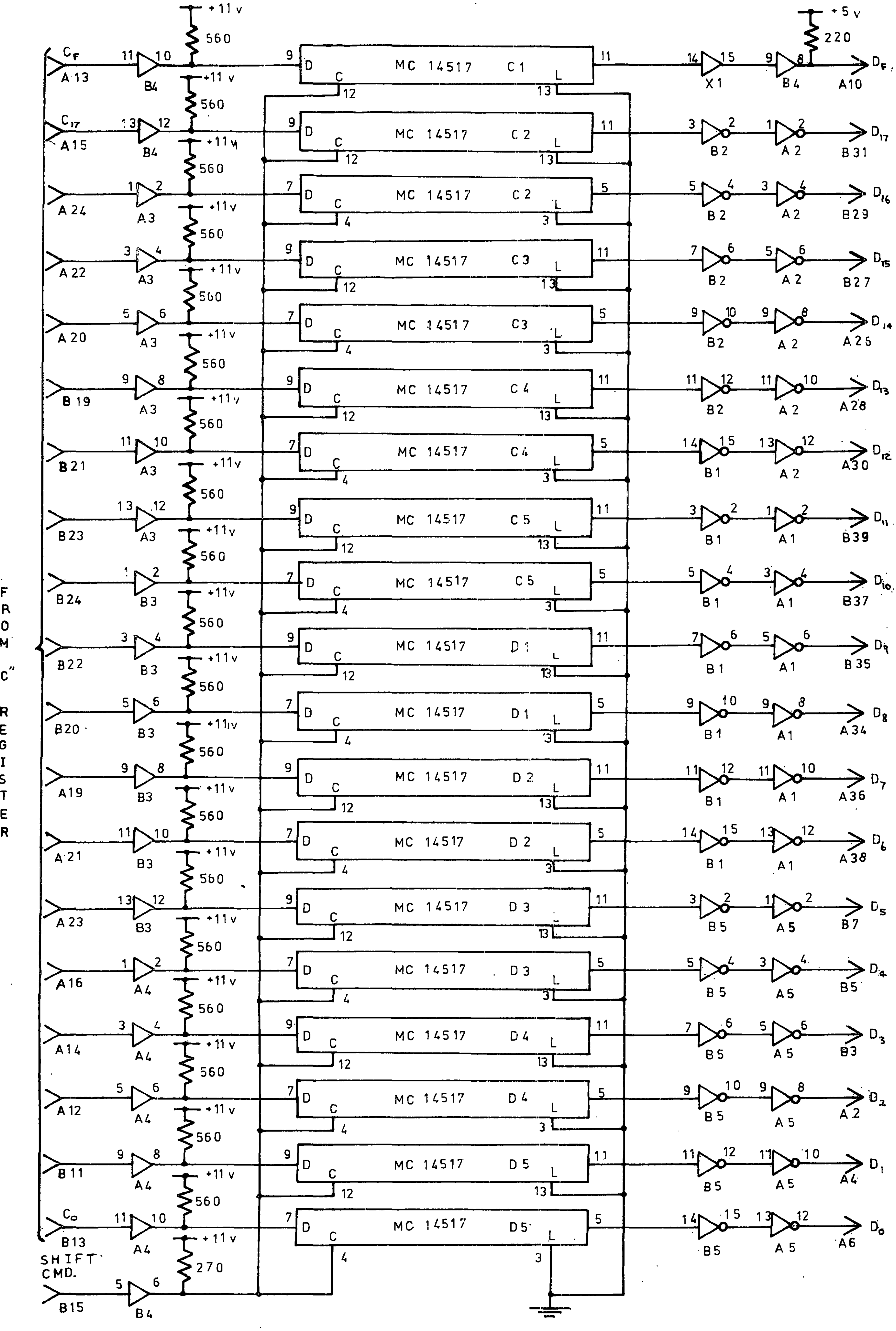
S.4 REGISTER B



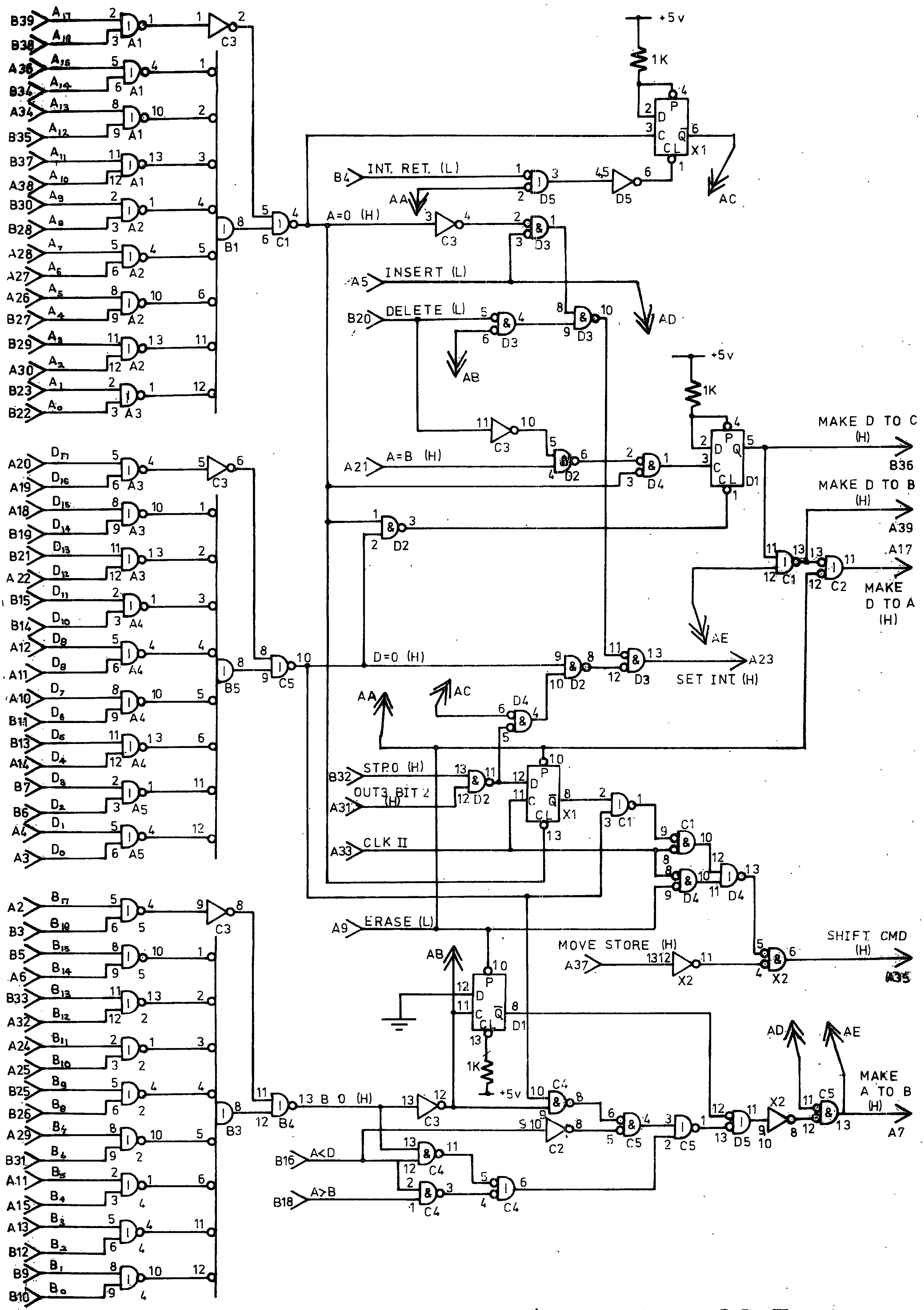
S 5 REGISTER C



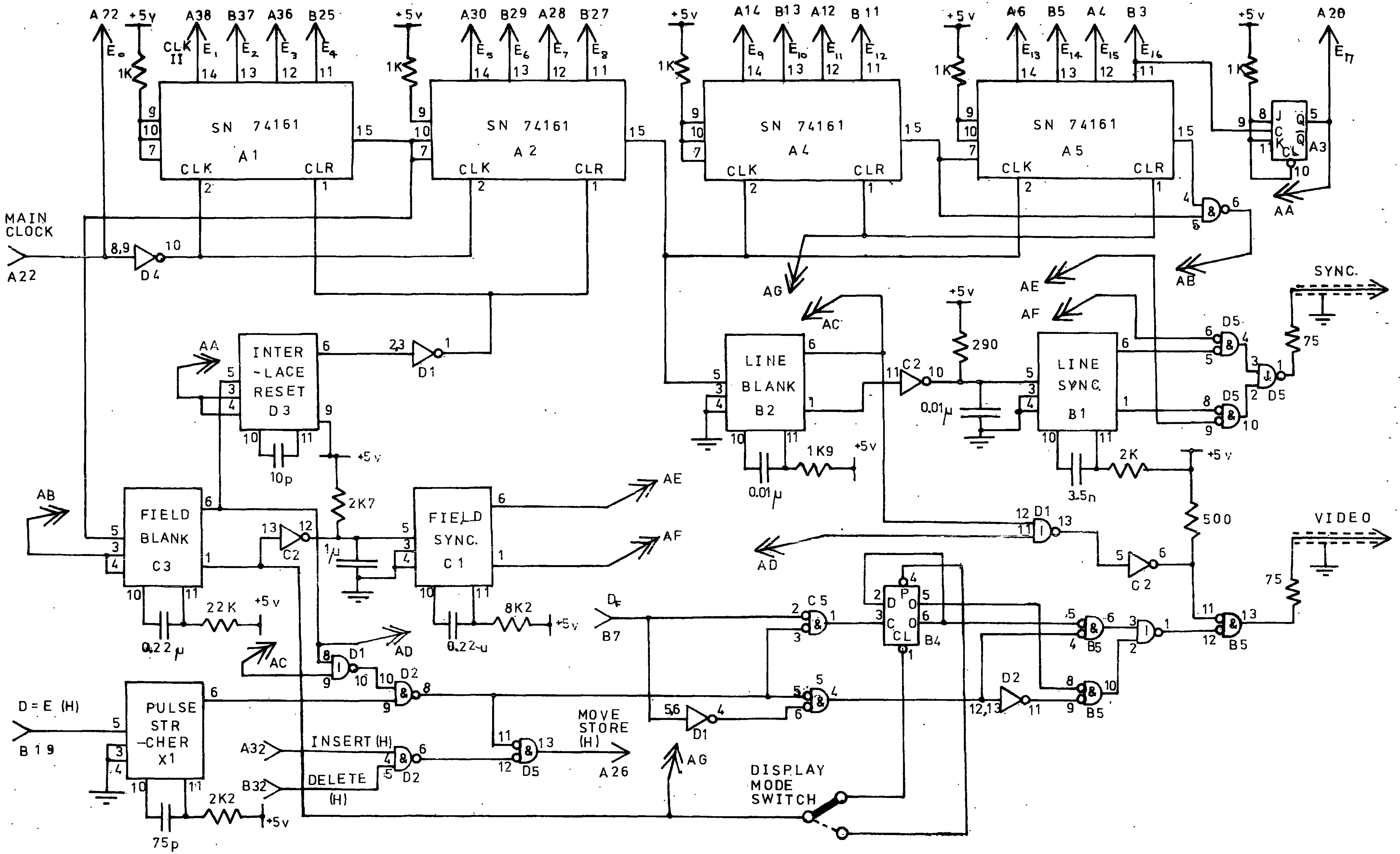
S.6 MAGNITUDE COMPARATORS



S.7 MAIN STORE



S.8 INSERTION / DELETION CONTROL



S.9 MONITOR CONTROL

REFERENCES

1. THEIS, D. J. & HOBBS, L. C., "Low-Cost Remote C.R.T. Terminals", Datamation, Vol. 14, No. 6, p. 22-29, June 1968.

2. HORMAN, A. M., "Designing a Machine Partner: Prospects and Problems", Datamation, Vol. 13, p. 29-33, February 1967.

3. MUMMA, F. W., "Progress with Displays - An Introductory Guide to Understanding and Selecting Digital Display Systems", Data Processing, Vol. 11, No. 10, p. 34-42, October 1969.

4. KAZAN, B. & KNOLL, M., "Electronic Image Storage", Chapter IV, Academic Press, New York, 1968.

5. WHARTON, W. & HOWARTH, D., "Principles of Television Reception", Chapter 3, Pitman, 1967.

6. MCS-8 Microcomputer Set. 8008 8-bit Parallel Central Processor Unit, Users Manual, Intel Corporation, November 1973.

7. BRATT, R., "C.R.T. Display Using T.V. Components", Australian Electrical Engineering, p. 30-32, October 1973.

8. Standard Telephones and Cables, I.T.T. 3210 Visual Display Unit, Engineers Handbook.

9. WALKER, B.S., GURD, J. R. & DRAWNEEK, E. A., "Interactive Computer Graphics", p. 49-50, Edward Arnold, London, 1975.

10. MACAULY, M., "A Low Cost Computer Graphic Terminal", Proc. Fall Joint Computer Conference, 1968, p. 777-785.

11. ROSE, G. A., "Intergraphic: A Microprogrammed Graphical Interface Computer", I.E.E.E. Trans. on Electronic Computers, EC-16, No. 6, p. 773-784, December 1967.

12. BRYDEN, J. E., "Addressing the Elements of a Visual Display", Computer Design, Vol. 13, p. 85-93, November 1974.

13. CHEEK, T. B., "A Graphic Display System Using Raster-Scan Monitors and Real Time Conversion", S.I.D. Int. Symposium, Digest of Tech. Papers, p. 56-57, New York, May 1973.

14. THORNHILL, D. E. & CHEEK, T. B., "Raster-Scan Tube Adds to Flexibility and Lower Cost of Graphic Terminal", Electronics (U.S.A.), Vol. 47, Pt. 3, p. 95-101, February 1974.

15. SHURTLEFF, D. A., "Studies in Television Legibility - A Review of the Literature", Information Display, Vol. 4, Pt. 1, p. 40-45, February 1967.

16. Intel Corp. P2125-2 High Speed Static 1024-Bit Three State Output R.A.M. - Data Sheet.

17. Motorola Semiconductor Products Inc., MC 14517 AL/CL/CP - Data Sheet.

18. BALDWIN, J., STALLEY, A. D. & KITCHIN, H., "A Standards Converter Using Digital Techniques", I.B.A. Technical Review, p. 15-29, June 1973.

19. Texas Instruments, T.T.L. Data Book, p. 120, 1973.