SOFTWARE AND HARDWARE ASPECTS OF A MICROPROCESSOR CONTROLLED LATHE

by

JOHN HOPTON

A Thesis submitted towards the fulfilment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

to the

UNIVERSITY OF ASTON IN BIRMINGHAM

Department of
Production Technology and Production Management

July , 1983

D E C L A R A T I O N .

No part of this work described in the thesis has been submitted in support of an application , for another degree or qualification , of this or any other university , or institute of learning.

.....

(J HOPTON.)

Acknowledgements.

I wish to express my sincere thanks and appreciation to the following , who assisted me during the period of research.

Dr D A Milner. Ph.D; BSc.[Eng]; C.Eng.M.I.Mech E; M.I.Prod E; the project supervisor , for his valuable advice , guidance and encouragement.

Professor R H Thornley. Ph.D; MSc.Tech; DSc; AMCT.C.Eng; F.I.Mech E; F.I.Prod E; the head of the department , for the use of the departmental facilities.

Ebi Razavi; a Ph.D research student , for his friendship and advice and for the great companionship we had together.

Messrs Charles Partridge and Alf Exton; the departmental technicians , for their friendly cooperation in finding those "hidden" necessities.

Burton Upon Trent Technical College; my employers , who enabled me to visit Aston University.

Betty [Hopton]; my wife , for her devotion and duty during those very long hours of "enforced" silence and many other sacrifices, that made this work a possibility.

J HOPTON.

THE AUTHOR

The author was born in Carmarthenshire in West Wales and is now married and living at Burton Upon Trent, where he lectures in mathematics and microprocessors , at the local technical college.

He was trained by an engineering apprenticeship and gained further field experience during his period of duty in the National Service in the R.E.M.E , where he also gained a HNC in mechanical engineering with endorsements. When his interest in electronics developed further he retrained in the field of colour television and obtained an O.N.C in electrical engineering . He also operates a private radio transmitting station having acquired a Post Office Radio Amateur licence with two international call signs, G8 AXF and G3 WMP.

He then obtained a degree with the Open University by private home study and continued to gain an M.Sc at Aston University in mathematics with a thesis in The Rotational Rate of the Atmosphere by research work on a Russian Earth Satellite in conjunction with R.A.E Farnborough.

The author is now involved as a consultant to many local industries and attends regular seminars at a National Beer Brewing Company at Burton Upon Trent, a large car tyre and slipper manufacturer and the National Coal Board Research Establishment, where he trains both engineers and management in the field of microelectronics. He relies heavily on the experience gained at Aston University and at his private home laboratory where he is, at the moment, furthering the application of microprocessing to the control of tracking and receiving ground pictures from the weather satellite Meteostat, in conjunction with the European Space Agency at Darmstadt in Germany.

The Journal of the Society of Electronic and Radio Technicians have published several articles from the author who has also himself published two books on the topic of machine code programming of a microprocessor .

The University of Aston in Birmingham.

SOFTWARE AND HARDWARE ASPECTS OF A MICROPROCESSOR CONTROLLED LATHE.

BY

JOHN HOPTON

A thesis submitted for the degree of Doctor of Philosophy.

July , 1983.

SUMMARY

A small lathe has been modified to work under microprocessor control to enhance the facilities which the lathe offers and provide a wider operating range with relevant economic gains. The result of these modifications give better operating system characteristics.

A system of electronic circuits have been developed, utilising the latest technology, to replace the pegboard with the associated obsolete electrical components.

Software for the system includes control programmes for the implementation of the original pegboard operation and several sample machine code programmes are included, covering a wide spectrum of applications, including diagnostic testing of the control system.

It is concluded that it is possible to carry out a low cost retrofit on existing machine tools to enhance their range of capabilities.

KEYWORDS :

Microprocessor Control, Pegboard Replacement, Lathe.

CONTENTS. PAGE.

LIST OF FIGURES. PAGE.

INTRODUCTION.

CHAPTER 1.

CHAPTER 1.

INTRODUCTION.

The work is mainly concerned with the investigating the retrofitting of a microprocessor based controlling device to an existing , pegboard operated capstan lathe and to investigate new interfacing techniques, so as to enable the microprocessor to be coupled to various external devices.

The research work utilises a British made Ward capstan lathe, which is still used in industry and to update the control system by replacing the pegboard, the programming of which required a time consuming effort by the operator. Utilising existing machines minimises the cost, because to purchase and fit a small microprocessor controller is minimal, costing only in the region of a £1000 , which is far below the cost of totally replacing a working machine with a modern counterpart. Most working machines still have a very useful life span left and also existing tools could be used, reducing the cost further.

If further control capabilities were required then additional features could be controlled by simply updating the programmes and adding further microprocessor memory. The lathe was examined in detail and several points were noticed, these are detailed as follows.

The electronics involved gas filled thermionic tubes which are now obsolete and difficult to replace. This fact required that a replacement design be made, to modify the circuits with a solid state device, which could easily be obtained.

Normal operation of the lathe was controlled by an operator placing pegs into a matrix panel, but this was time consuming and prone to errors. The pegboard configuration was ideally suited to microprocessor control and so a self latching control interface was designed, to be simply connected to one of the control columns of the pegboard. This control being automatically taken over by the microprocessor , by pressing one operator control key. Control programmes could be pre-written and fed to the processor via a standard cassette tape recorder, or a plug in EPROM memory.

The next step was to investigate which processor chip to utilise in the design and the project contains a critical appraisal of the main types of microprocessors available during the early part of the research. A final choice was made to use the Zilog Z80 chip, which has since become an industrial standard quickly taking over the predecessor, the 8080, because of the powerful instruction set , ideally suited to numerical control.

The Z80 microprocessor chip design is intended to drive the interface components connected to the first column of the pegboard, with an electronic latch configured to return to the first column and then to read each new set of instructions presented by the microprocessor. The system should give excellent flexibility with a set task loaded directly from the keyboard or more efficiently from a standard cassette tape. The section on improvements also suggests a better method, by obtaining routines from a pre-programmed EPROM.

The work contains several programmes, proposals and designs for additional improvements to the system, these include a memory expansion unit, enabling better control programmes to be written, without the restricted memory space.

Most of the control programmes are written in machine code language, which has the important advantage of speed of operation, which is required in any control system and it allows a limited memory section to be utilised to the maximum efficiency. Many sample machine code programmes are given in the text and many more may be found in the author's own publication. (1)

A further design is included to programme UV EPROMS , with either a control programme , or an improved monitor system . These EPROMS give an excellent facility for simply plugging in a control programme for a complete task. This has cost saving advantages to industry and a design is included to reprogramme these EPROMS , thus saving replacement costs.

The modified lathe has been extensively tested and after initial modifications has been found to give excellent flexibility . Sample programmes are provided , to enable a new operator to gain confidence.

The project was slightly extended by interfacing the processor to a VDU graphics unit and sample programmes are included to enable simple graphics to be produced.

Provision has been made to store programmes externally , by the addition of two further interfaces , to a standard tape recorder and to any printer , operating on the EIA RS232 voltage sensing system , or the 20 mA current loop configuration.

A section is included involving a further microprocessor should it be deemed advantageous , to control other aspects of the lathe , this facility would open up a great many more possibilities for further research work. There then follows a discussion of the relatively new fibre optic technology which could be utilised to transmit video signals from the machine tool tip to a visual display unit , so tracking the tool contour at the actual work face. The theory of fibre optic technology is detailed and ends with a typical design circuit for fibre optic light transmission.

WARD CAPSTAN LATHE.

CHAPTER 2.

CHAPTER 2.

THE WARD CAPSTAN LATHE.

2.1. Retrofitting a Microprocessor Controller.

Several factors were taken into consideration when attempting to change the control mechanism of the capstan lathe.

a) Compatibility between the lathe and controller with respect to signal and voltage levels. Investigation of the existing electronics highlighted the fact that electronic technology had advanced in rapid leaps since the lathe was initially developed making some of the components obsolete. These could only be replaced by stocks held in a rapidly diminishing store, not only were they difficult to replace but the components also operated on voltages between 90 and 220 volts dc. This voltage range is incompatible with modern electronic standards which generally utilise a 5 volt TTL level.

It was decided to replace these units with a modern counterpart which could easily be obtained and which would operate on the required standard voltage level. The transfer signals in the lathe operated at approximately 90 volts with a pulse duration of 1 millisecond and a further modification was made to the pulse, by changing it to a 5 volt , 1 millisecond transfer pulse at TTL level with a latched input.

b) Removal of the pegboard.

The original pegboard consisted of columns of holes in which an operator pushed in diode pegs to complete a circuit, to control a sequence of operations. This type of configuration was ideally suited for microprocessor control and so an interface board was developed at standard 5 volt level to make the task of connecting the controller possible. The following photograph shows the original pegboard before modification. (Figure 2.1)

2.2. Control Details.

The microprocessor's address and data lines were used to control a data latching system allowing a preset sequence of events to occur, this was previously set up on the pegboard. When a sequence is completed a transfer pulse from the lathe is communicated from a latch to the peripheral port of the processor, the pulse is taken as a signal to output the next set of instructions through the address and data lines. A continuous sequence of events is thus possible by a pre-written programme entered from the keyboard, or EPROM memory, or a tape cassette. All these control signals are confirmed visually on the screen of the television monitor or VDU. Figure 2.2 shows an overall picture of the control layout.

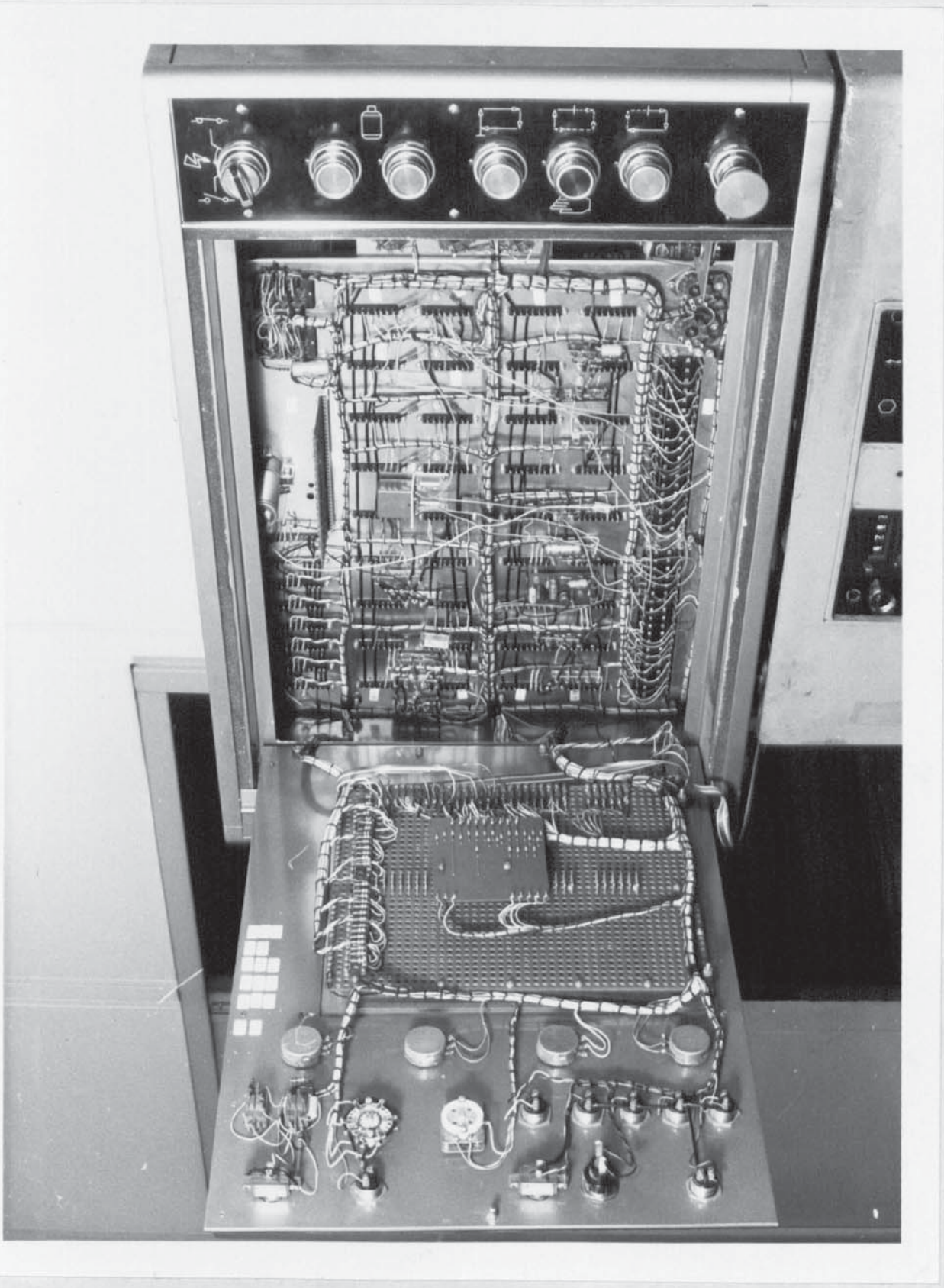


Fig. (2.1.) THE ORIGINAL PEGBOARD.

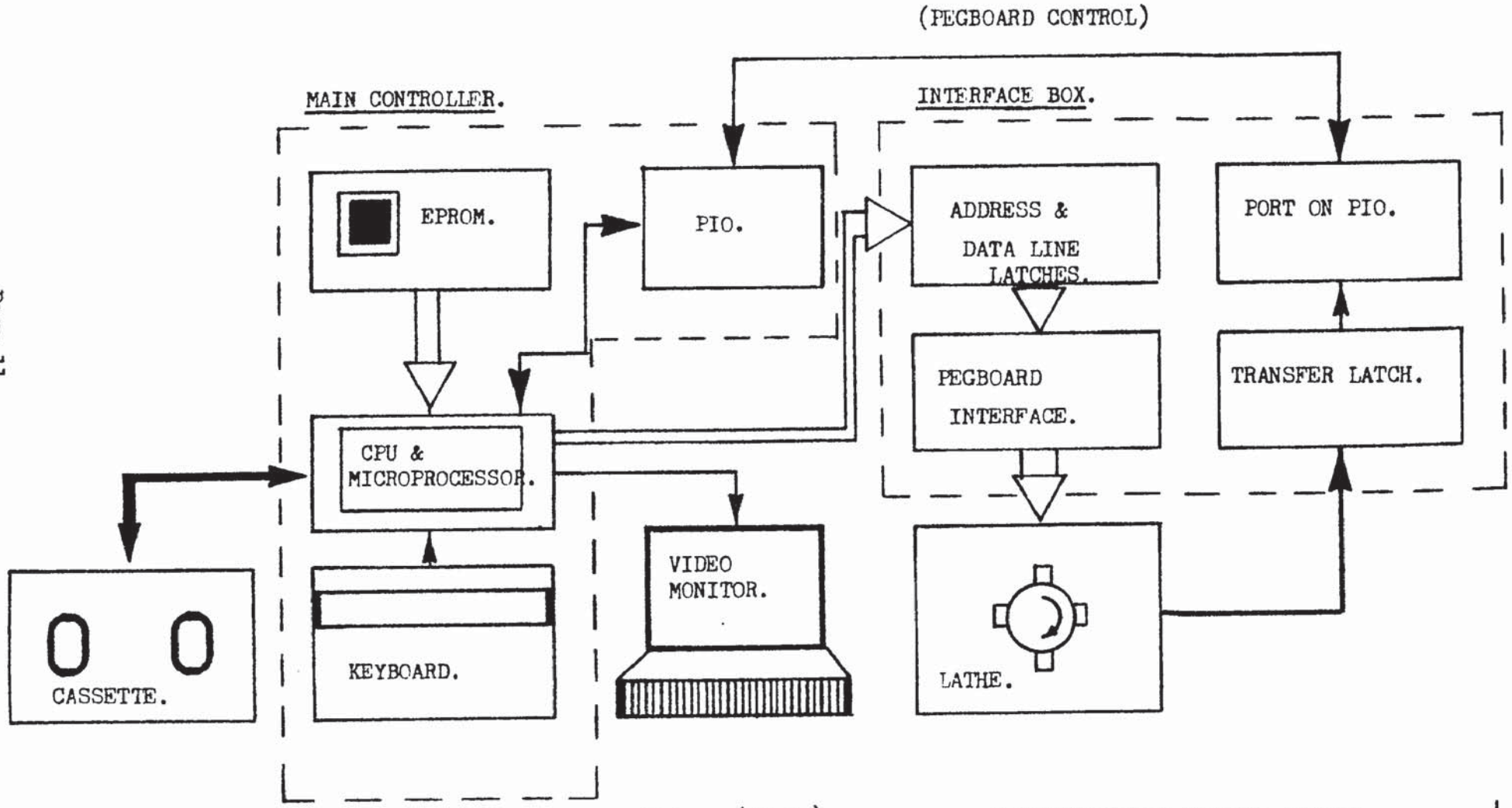


Fig. (2.2) MICROPROCESSOR CONTROL LAYOUT.

2.3. The Ward Capstan Lathe.

The lathe used for the research work is the 2DS AutoWard Capstan Lathe , manufactured by H W Ward of Birmingham and consists of two controlled sections (a) and (b) as follows:-

a) Originally the turret was controlled manually, but previous work was undertaken to operate this section electronically by using operational amplifiers with logic control [2].

b) Most of the other operations were controlled by the use of a diode pegboard arrangement consisting of a matrix of 31 rows and 24 columns as shown in figure 2.1. Lathe control was achieved by an operator using a set of pegboard diode insertions into this matrix, each diode controlled a circuit to move certain parts of the lathe. The main operational movements consisted of :-

i) A moving slide.

ii) Chuck speed and rotation.

iii) Turret movement , forward and backward.

iv) Chuck opening and closing.

This lathe has been used in British industry for many years but unfortunately, due to the advancement of technology, suffers from many disadvantages as highlighted below.

2.3.1. Lathe Disadvantages.

a) The time required to set up the pegboard is very expensive because today, labour is one of the most costly items in the manufacturing process.

b) Some of the electronic components are obsolete and once stocks are depleted their replacement will become very difficult.

c) Flexibility of manufacturing suffers due to the lack of precision control with no feedback between the lathe and the operator or controlling device.

These disadvantages were investigated and several changes were considered, such as to update the obsolete components with modern easily obtainable parts. Control would be passed to a Z80 microprocessor system which has a very powerful instruction set, allowing greater flexibility of manufacturing techniques. The performance could also be improved by fitting stepping motors to lead screws giving more flexibility when cutting components.

Further possibilities include tool monitoring by the use of microprocessor controlled fibre optic feedback. An important factor in any improvement is cost and with technology leaping ahead British industry has a difficult time with replacement of existing machines, retrofitting the proposed microprocessor controller gives an excellent compromise.

Existing equipment can be utilised thus reducing the costs to an absolute minimum. Any cost incurred would come mainly from the purchase of a microprocessor system and there are now available some small Z80 units.

These minimal systems could be pressed into service and would certainly be sufficient for many small factories utilising this type of lathe.

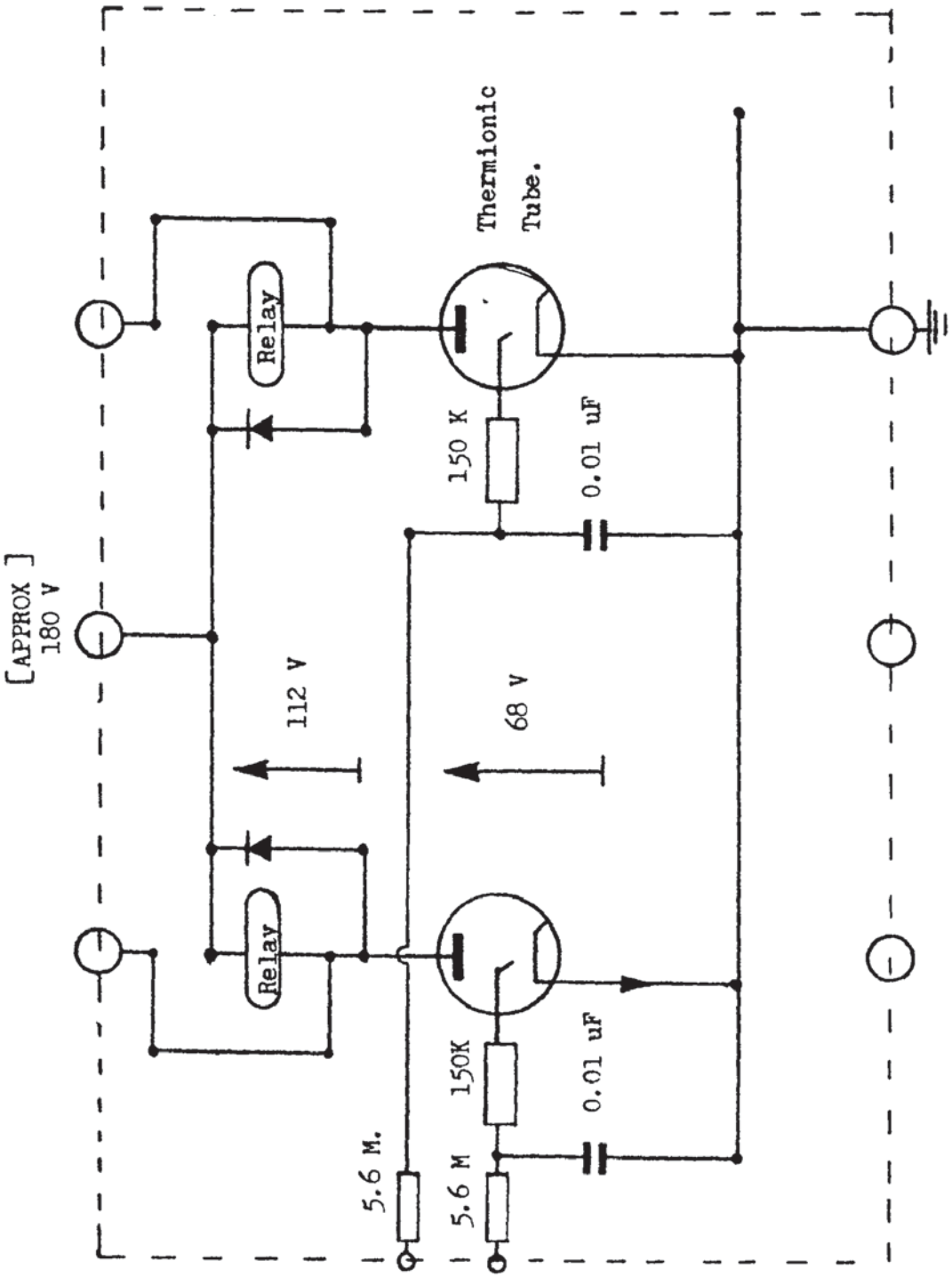
Costs would be in the region of £1000 which is a small outlay considering the advantages to be gained and a minute outlay if balanced against the cost of replacing the whole machine.

An added microprocessor controller does not of course solve all industrial problems, but it does offer a relatively cheap way of at least progressing in the correct direction with a minimum outlay.

2.4. Pegboard Control Modifications.

The original method of controlling the lathe was to enter some diode pegs into a column of holes , on an external pegboard as detailed in figure 2.1. As the diode peg was inserted a circuit was completed via the diode, connecting a voltage derived from the counter unit with two or more appropriate control flip flops. These flip flops utilised glass tubes.

The circuit shown in figure 2.3 details the original method of triggering the two thermionic glass tube devices, but unfortunately these tubes are now obsolete and future replacement even during fair wear and tear could become very difficult. The glass tubes used are of the Neon type, operating on a voltage source of 180 volts and direct current. It is also necessary for some of these devices to have a light source operating within the cabinet for reliable operation and if this light source is removed, by possibly just normal bulb failure, then the operation of the flip flops could be drastically changed unknown to the operator, because there is no feedback or warning device fitted to allow for this hazard.



OLD WARD CIRCUIT.

Fig. (2.3.)

2.5. The New Trigger Unit.

Now a replacement device was required to roughly approximate the dynamic characteristics and operation of the old Neon trigger tubes. Albeit these new devices operate at low voltages it was decided to temporarily stay with the original voltage so that each trigger could be replaced and tested in turn, without interruption to the normal lathe operation. This method proved invaluable when some design problems were met when testing the system, the reason being that the Neon tubes were utilised somewhat differently in some of the control units and required a different approach.

The design was carefully contrived so that the system could be changed over to a lower working voltage when all the modifications were completed, the arrangement resulted in the necessity to simply remove load resistors in each module when the change over was required. Figure 2.4 shows the method of replacement with the circuit changes. Further details of the new component C106 may be found in (3).

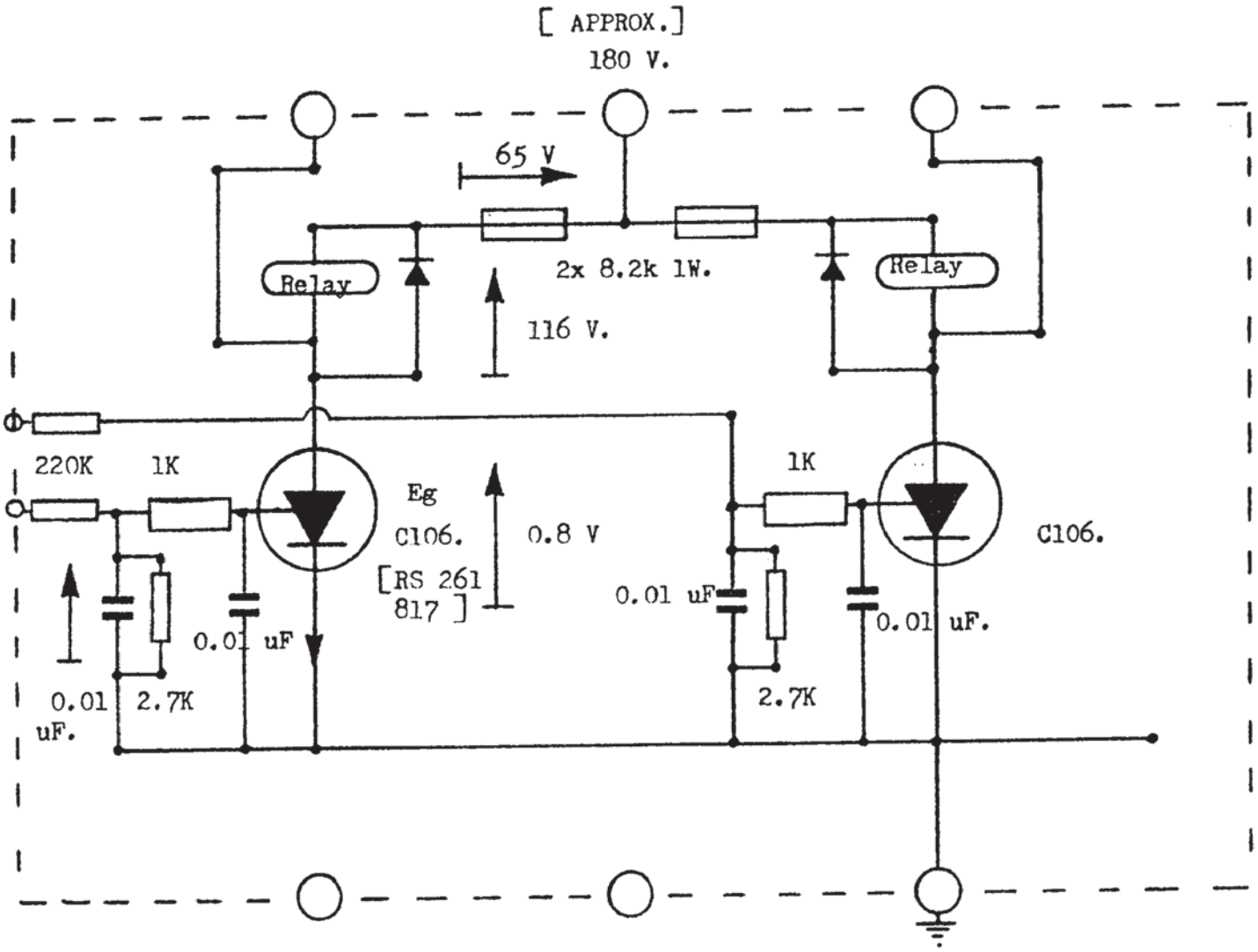


Fig. (2.4.)

NEW WARD CIRCUIT.

2.6. Transfer Unit.

When a lathe operation is completed a transfer pulse is provided by the system, to switch a transfer unit flip flop.

The unit derives a 90 volt pulse with a duration of between 1 and 3 mseconds. This pulse is sent to a counter unit, which steps control on from a previous column of diode pegs, to the next column in sequence. The new column contains the information required to perform the next lathe operation.

The following diagram shows the details of the electronics required to generate the 90 volt, 1 to 3 msecond transfer pulse. (See figure 2.5)

This pulse must also be reduced, before it can be utilised in a modern control circuit, due to the incompatibility of the two voltage systems.

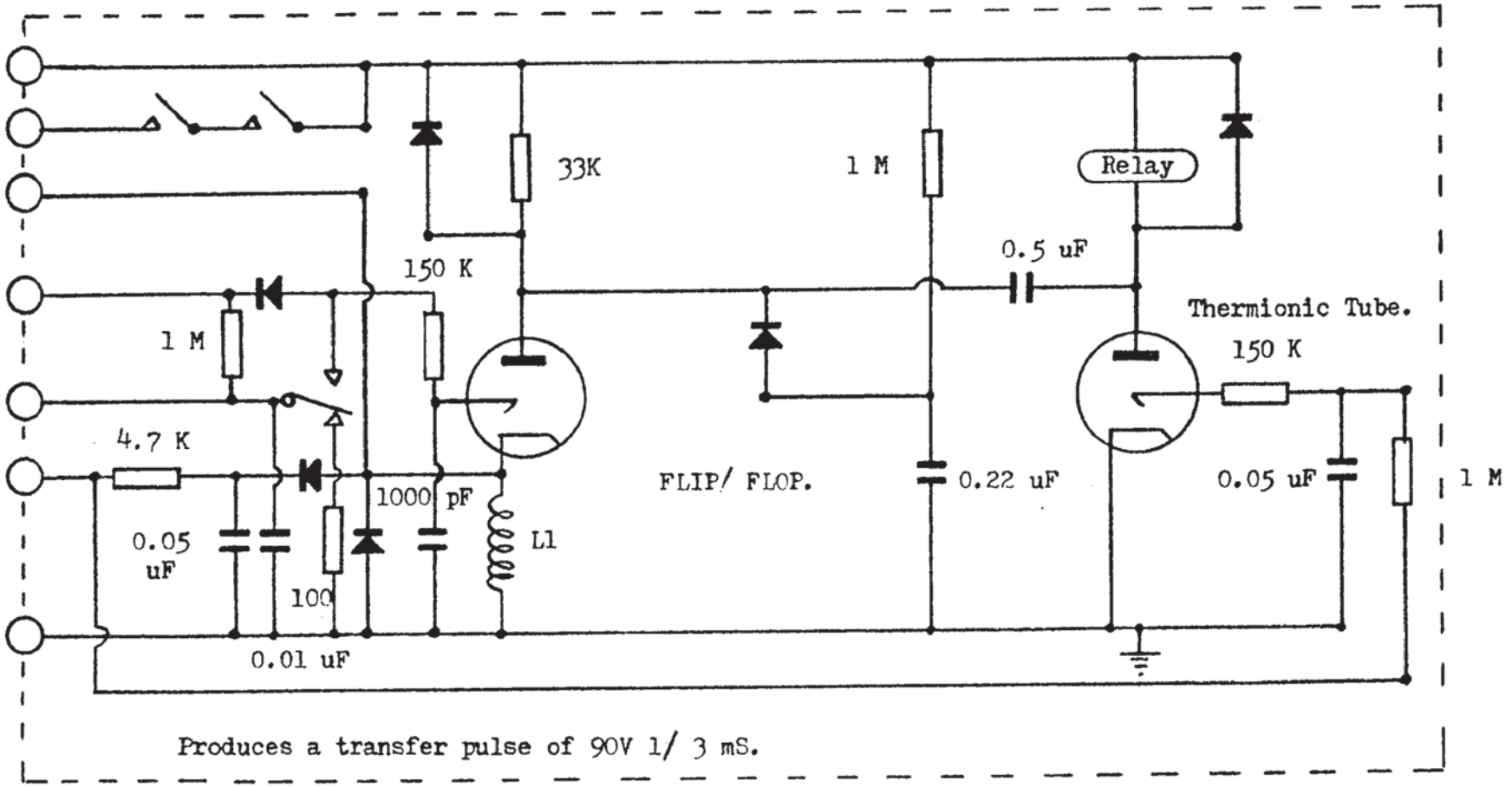
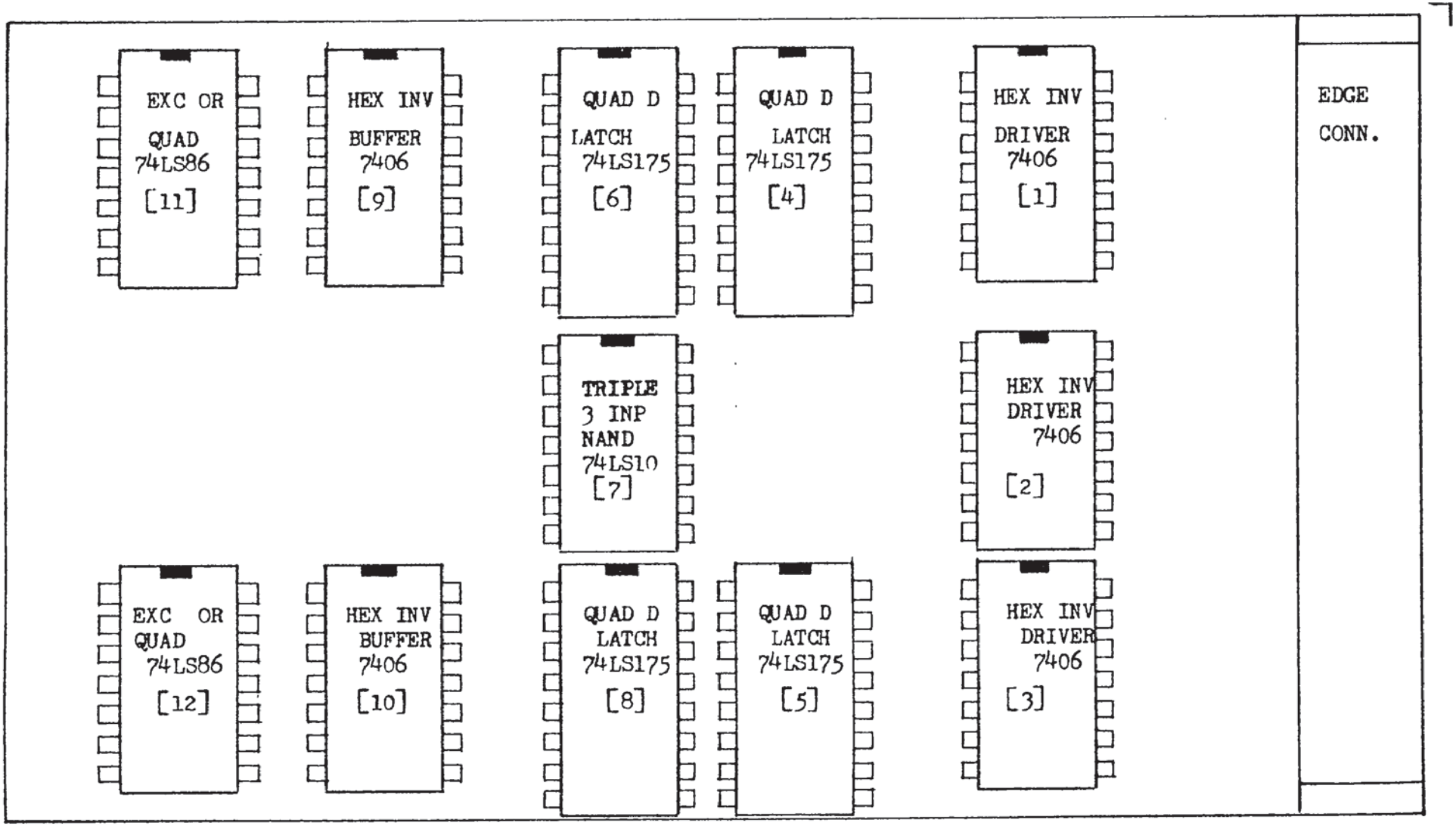


Fig. (2.5.) WARD TRANSFER UNIT.

2.7. The Pegboard Interface.

It was decided to replace the operation of the pegboard by a microprocessor as the next step towards controlling the AutoWard capstan lathe. This had been made possible by the changes made to the flip flop and transfer units to a lower operating voltage. The full advantages of this modification would become apparent later when all the lathe operations were microprocessor controlled.

The initial steps consisted of designing , building and testing an interface board between the microprocessor's peripheral ports , address and data lines and the pegboard diode assembly. The diode insertion action of the operator would be mimicked under microprocessor programme control from the keyboard directly or from a programme entered previously into an EPROM memory , or from data held on a tape cassette. Figure 2.6 shows the physical layout of the interface board and indicates the position of the buffers , drivers and latches utilised in the matching process.



RELAY INTERFACE BOARD. Fig. (2.6.)

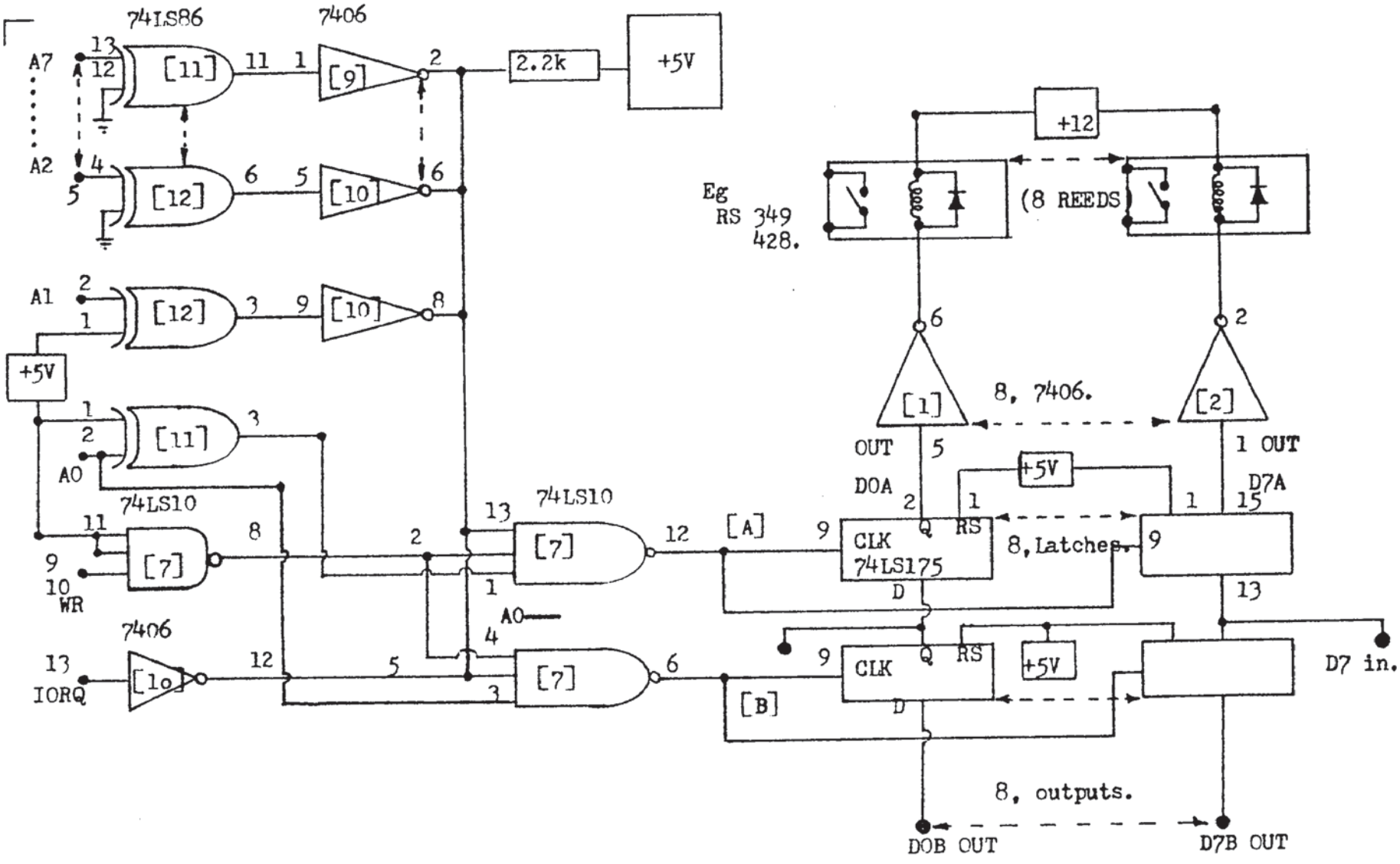
2.8. The Interface Board.

The interface board consists of a set of 74LS86 , see (3) for further details, these are quad exclusive OR logic integrated circuits that connect to the microprocessor's address bus. Their outputs are inverted and then buffered by 7406 type integrated circuits (3). All the address lines A_1 to A_7 act as outputs and are coupled together and held at the 5 volt logic level by a common resistor . This logic level is connected as one input to two triple input 74LS10 (3) NAND logic gates. The other two inputs consist of the A_0 address line together with the $\overline{WR}$ write pulse, from the CPU. The common address line for A_1 to A_7 is also controlled by the microprocessor's $\overline{IORQ}$ request line.

At any particular time only one of the 74LS10 gates give an output, because the A_0 address line is fed to one gate directly and inverted to the other. This facility allows a choice to be made, to activate either of the bank of two, eight 74LS175 latches. The sixteen latches are connected in two groups of eight, one group beeing selected by the appropriate 74LS10 gate. The latches are connected directly to the CPU data bus but only pass data when clocked, by their respective gate.

The output from the 74LS175 quad D type flip flops are fed to a bank of 7406 inverters utilised as buffers , to control a set of reed relays as shown in figure 2.7. The reed relays mimic the action of the operator inserting a peg into the diode matrix board. The relay outputs are connected in series with diodes and each output is then connected in parallel with the appropriate peg position on the matrix board. Figure 2.8 shows the reed interface board connected to the address lines and routed to an edge connector ready for coupling into the microprocessor output socket.

Only one column is utilised on the pegboard, by the microprocessor, because this allows the microprocessor to set up the instructions for each operation as though the pegs had been inserted by the original operator. When the lathe is activated, action is taken on each instruction by reading the same column continually , this column being controlled by the microprocessor software and the lathe electronics is fooled into returning to reread the same column continually.



REED INTERFACE CIRCUIT TO Z80.

Fig. (2.7.)

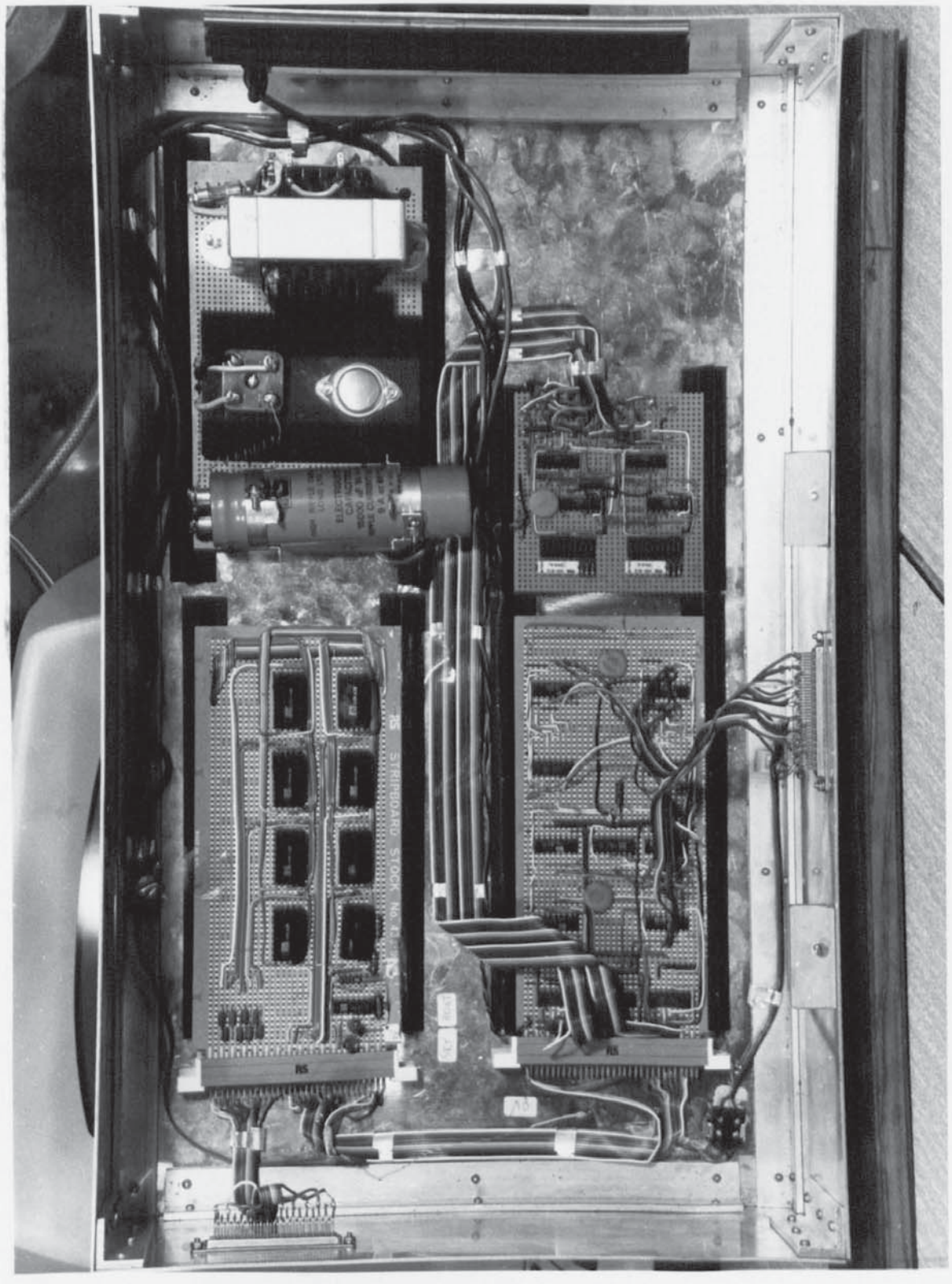


Fig.(2.8.) THE INTERFACE BOARD.

2.9. Pegboard Modifications.

Eight lines were connected to the pegboard enabling eight separate control functions to occur , however the full set of 16 latches were connected to the address lines , giving a further eight possibilities for future expansion and control of alternative devices. Each line is connected to the peg diode matrix board as detailed in figure 2.9. The diodes were required in the interconnections so as to steer the signals correctly to mimic the original action of each diode peg inserted by the operator.

When one column of instructions were completed by the lathe an automatically generated transfer pulse of 90 volts, with a width of between 1 and 3 milliseconds was obtained from the lathe's control system. This pulse was required by the microprocessor's system, but required some modification to tailor it to the standard 5 volt TTL level, before it could be utilised. Figure 2.1 shows where the control lines were connected on the lathe behind the original pegboard.

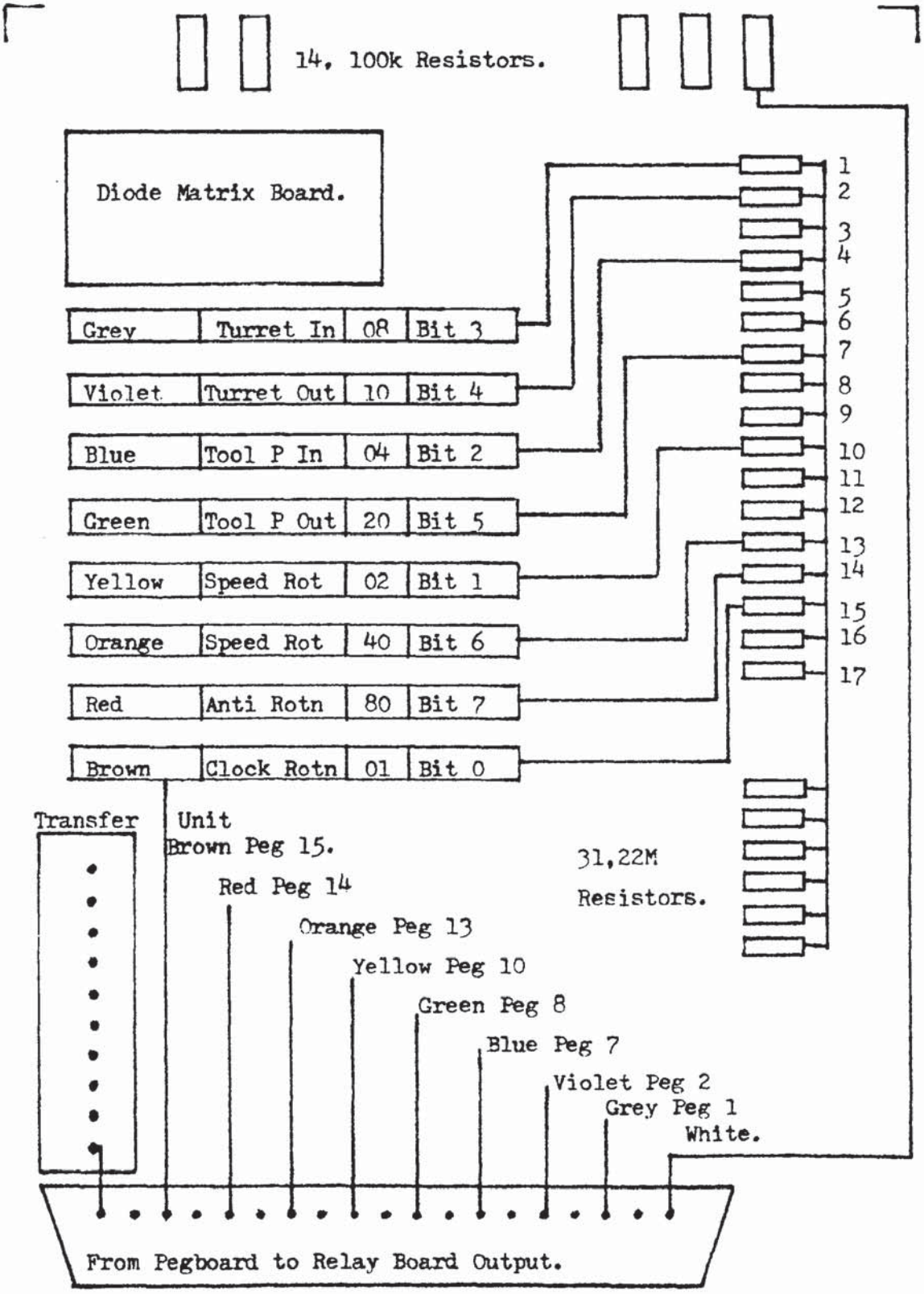
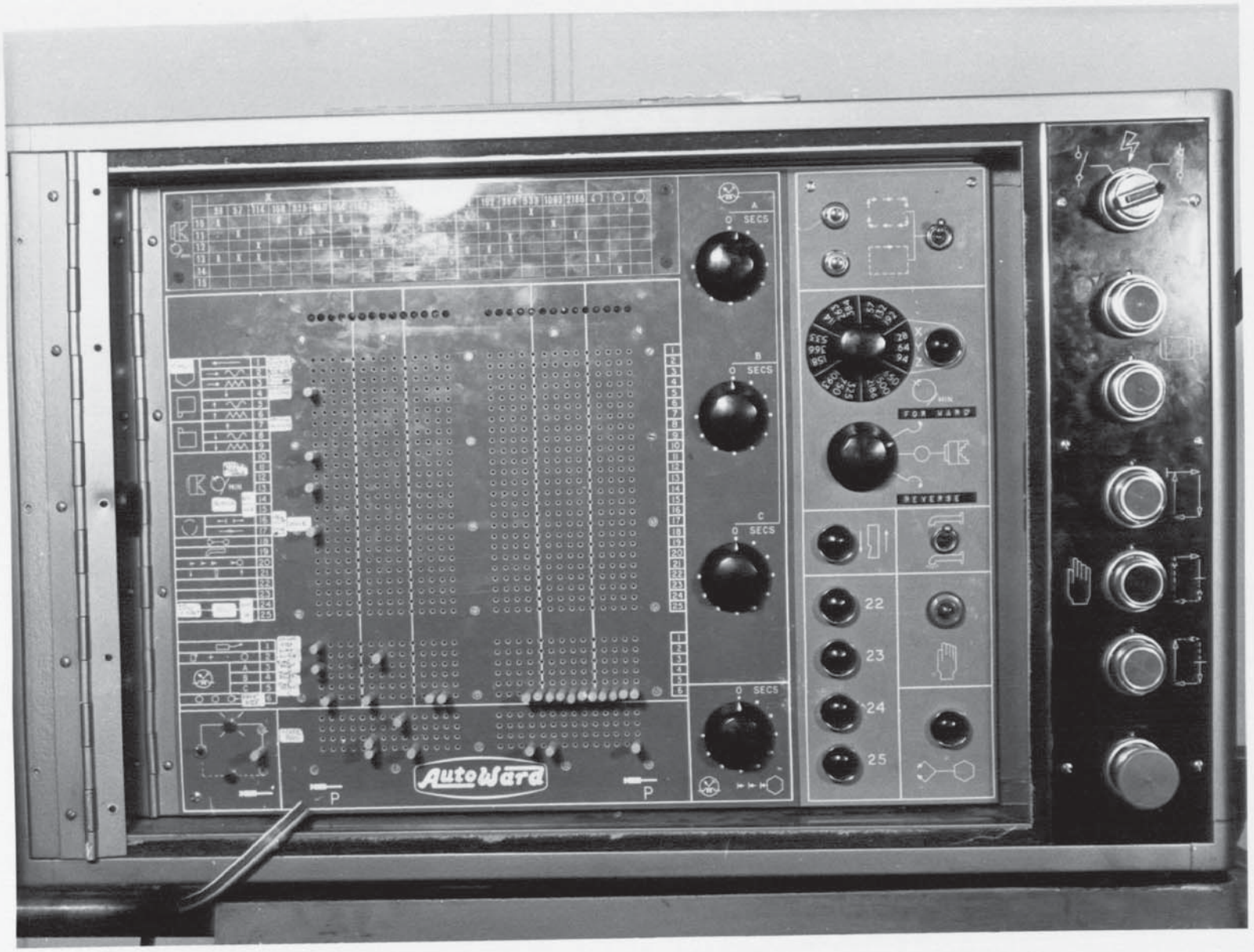


Fig. (2.9.) Behind the Pegboard.



Fig(2.10.) BEHIND THE PEGBOARD.

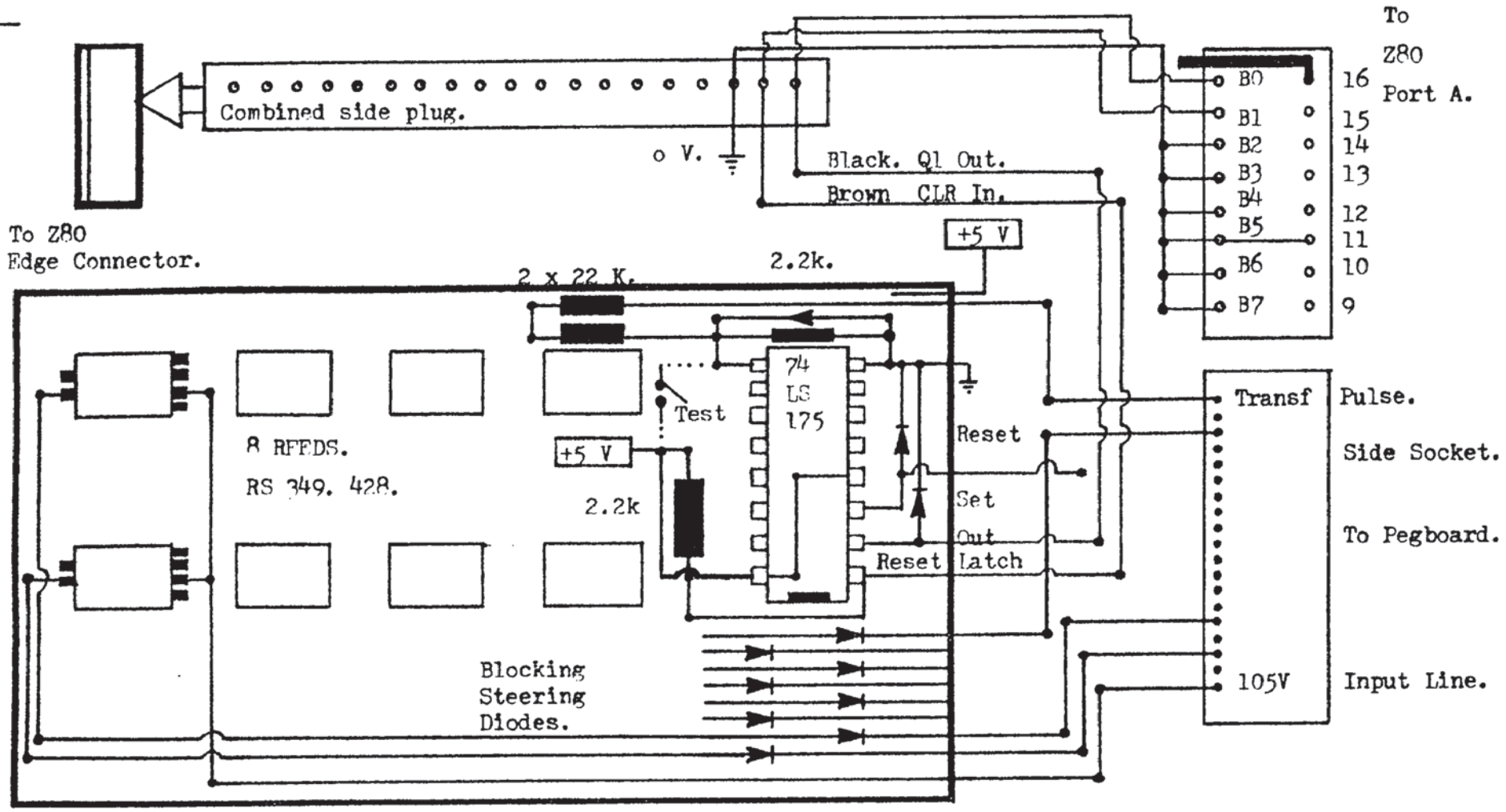
2.10. The Transfer Pulse.

The 90 volt transfer pulse required reducing to the standard TTL voltage level of 5 volts and a latch was used to hold the pulse, giving the system further flexibility, so that if required at a later date, the microprocessor need not be held up in an interrogating loop, until it had completed some other more important task.

When the transfer pulse is received by the processor, the next set of instructions is communicated to the pegboard column.

This pulse results in further action by the lathe and the process continues as before.

Details of the transfer pulse modifications and latching circuits are shown in the following diagram, figure 2.11.



Latch may be set by closing "Test" switch momentarily.

Fig. (2.11.) RELAY BOARD AND TRANSFER LATCH.

2.11. Counter Unit Modifications.

The original system utilised a string of "GDR 120W" counters in a ring loop configuration and as each counter was triggered in turn, the output was used to activate each of the peg columns individually.

Due to the nature of the microprocessor controller only one of the columns is used, hence a modification was required to the counter unit so that in effect it returned to the first column automatically, after each step.

The following diagram shows the details of how the counter is forced to return to continually activate the first column, see figure 2.12.

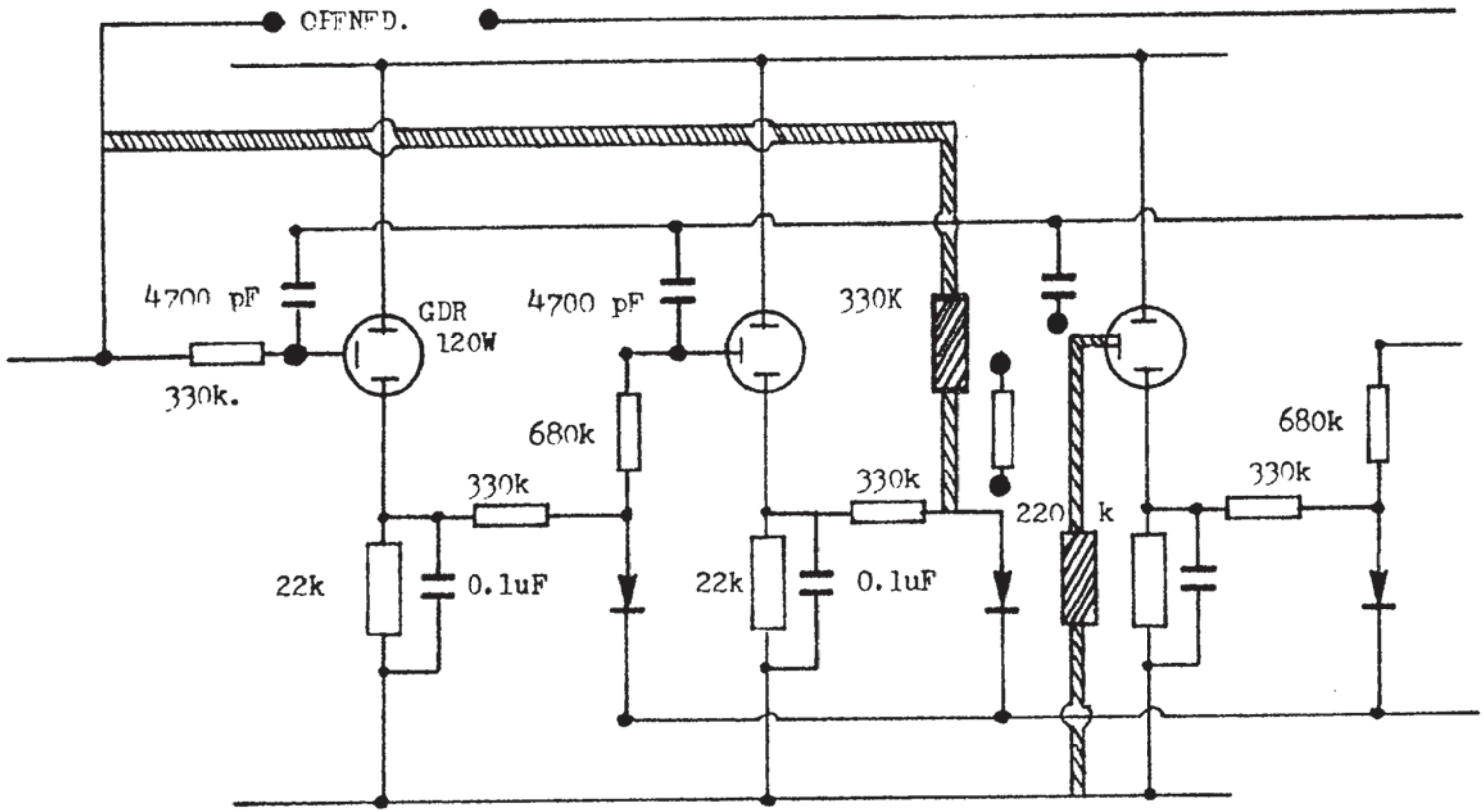


Fig. (2.12) Counter Unit Modifications.

2.12. The Microprocessor Controller.

The previous modifications enabled the microprocessor to communicate with the lathe via the interface and latch system. The following events are typical of a control sequence, when operating the lathe.

a) Switch on the lathe.

b) Switch on the microprocessor.

c) Type in, or feed in a prerecorded programme.

d) Initialise the programme sequence.

The events that now follow are directly under the control of the microprocessor :-

1) The transfer pulse latch is reset, ready for the next transfer pulse input.

2) An event sequence is sent out, on the address and data lines, into the interface latch system. A visual confirmation is then given on the monitor screen.

The event data is obtained from a prewritten event table, contained within the machine code programme.

3) The latched information is fed directly to the rear of the pegboard where the signal is interpreted as a lathe event.

4) A transfer pulse is accepted from the lathe and latched ready for reading by the microprocessor.

5) In fact , in the original sequence , the processor initialises the interrogation of the latch as soon as the pulse is received.

The processor performs four operations.

a) The transfer latch is reset , ready to accept the next transfer pulse.

b) A predetermined delay sequence is entered , between each event . This delay is adjustable under software control.

c) A new set of instructions is sent out to the data latch interface board and the instruction is confirmed by a visual display on the monitor screen.

d) The processor enters a further interrogation loop waiting for the next transfer pulse, the software could also be changed to generate an interrupt.

6) This sequence continues until the processor finds the code "FF" at the end of the event table, this stop code will then initiate a separate set of operations as follows:-

a) The transfer latch is reset,

b) The chuck rotation is stopped for additional safety,

c) The end of programme sequencing is indicated on the video monitor screen.

The complete cycle could then be repeated with a new work piece, or a new control programme entered from the keyboard, from EPROM memory or from a tape cassette.

The author has written some sample test programmes, but have not been included at this point because of the necessity for the operator to gain at least some familiarisation with the machine code language before using those programmes. A chapter is included later (chapter 8) as an introduction to the machine code language with sample test programmes inserted at the end of that section. For further programmes see (1), the authors' own book on machine code programmes for the Z80 microprocessor.

2.13. Choice of Controller.

During the period of design, construction and fitting, of the previously stated modifications, thoughts were obviously being given to the type of microprocessor to be utilised, when the modifications were completed.

Unfortunately, at that time, small inexpensive readily made available systems, were very few and far between and even the available ones were not found suitable for the control task in hand.

It was therefore necessary to build up a custom designed system, from the microprocessor chips and boards, available at that time.

Even then the choice was made difficult, by lack of availability and it was therefore found necessary to produce a critical appraisal of microprocessor chips and units, before progressing further.

As the time progressed and an understanding of the situation was obtained , more microprocessor units became available, so relieving the situation a little.

The following chapter will take the reader through the steps taken by the author, from searching for units to choosing the final microprocessor. The choice of microprocessor was found to be far more important than was first envisaged.

The experience gained was found to be invaluable as an aid in further industrial projects and designs . In fact the author now acts as a microprocessor consultant to several large industrial companies within his area and they are turning more and more to microprocessor control, but as yet do not have the nesessary background experience, when deciding on which microprocessor system to utilise for a specific task.

HISTORY OF THE MICROPROCESSOR.

CHAPTER 3.

CHAPTER 3.

THE HISTORY OF THE MICROPROCESSOR.

3.1. Electronic Changes.

Radical changes in electronics began as early as the 1960's with the first electronic calculators, which were products of the transistor era which eventually resulted in the integrated circuit.

The early 1970's brought a faster change due to the technical innovations as designers found a method of cramming more elements into tiny micro size electronic circuits, resulting in pocket size calculators at a substantially reduced price. The 1980's gave a microelectronics era which is a combination of new semiconductors and a totally new design technique , resulting in an unique technological progression, which was brought about by a system called " Large Scale Integration". The system enabled thousands of transistors to be placed on a tiny silicon wafer, enabling the computer to shrink both in size and cost.

In essence a microprocessor can replace some units that a mere decade ago would have cost many thousands of pounds. In the past logic design was mainly hardware oriented, system changes were difficult to implement , because a specific logic block would consist of a combination of many logic elements.

Changes were made either by redesign or by wiring in external components for entire circuits or even minor changes . Both were expensive in time and engineering resources.

For many years logic designers had hypothesised :-

"Why not replace gates and logic elements with a stored programme , thus performing all of the logic functions".

An interesting concept which was only feasible for large and expensive computers.

Eventually a convenient merging of "LSI" and the stored programme concept became a reality , resulting in the microprocessor . The microprocessor system stores the programme in a special LSI device called a memory chip . This chip is connected to the processor supplying control instructions, enabling conventional logic systems to be simulated.

Microprocessor based design is a totally new concept giving an unique mixture of hardware (ie with electronic blocks) and software (ie with programmes) , which promises to revolutionise the designer's task . Hardware is greatly simplified as all functions are centered around a general purpose processor.

Usually the hardware system design consists of interconnections and interfacing or matching of various LSI chips. These generally appear as a family of general purpose elements producing a chip marriage that significantly reduces the interfacing requirements.

In microprocessor based systems there are less circuit elements, so naturally less hardware design time is necessary. In essence the brunt of microprocessor design is the developement of software, consisting of flowcharts and programmes, these are written by the designer and then loaded into special memory chips, the programme then has the power to produce functions previously reserved for logical elements. Hence memory replaces random logic by storing programming sequences.

Generally one word of memory replaces one logic gate and many thousands can be stored in one memory chip, thus reducing the chip count drastically. This significant reduction in hardware complexity results in reduced design time and cost.Fewer parts means increased reliability giving a further overall efficiency to a cost effective design. In fact microprocessors used in industry have now become well known for their dependability.

A further advantage of the microprocessor system is that changes in environmental parameters can easily be implemented by a change in software.

3.2. Evaluation and Chip Selection.

There are various microprocessors available but the selection is far too important to be left to chance, or a hasty decision. The selection depends to a large degree on the application.

Processors are structured differently but they do have similarities or elements which are the same, their names may be different and they may have some unique factor. The subtle differences must be examined carefully during selection as they can sometimes indicate programming problems and solutions.

Microprocessor analysis often begins with the investigation of the manufacturer's data sheets, which can sometimes assist to remove misfits at an early stage.

The following items are generally recognised as an indication to the most important factors in the selection of a system.

3.3. Word Size.

This feature is a means of classifying processors into major groups. It is important from the standpoint of resolution, computational accuracy and ease of programming.

Eg : A 4 bit machine will act as an 8 bit device but requires two lots of data bursts. This is not usually time efficient and can complicate programming.

3.4. Addressing Modes.

The number of modes offered is very important and as a general rule more addressing modes mean more potential power, however some modes may be of little use, or be just a rehash of another inherent mode.

3.5. Address Capabilities.

This is the amount of memory that is directly addressable, eg :-

A 16 bit device may have a combination address:data bus to select 64 K** of memory, by applying three bursts of four bit multiplexed information. The disadvantage is that more time is required to address each location.

3.6. Input and Output Capabilities.

The input:output structure may be of paramount importance to a particular application, such as control applications, because the processor takes data from the outside world, processes it and sometimes returns a result to the sender.

** 1K of memory consists of 1024 bits of information, the number results from using the "Binary" system and 1024 is 2 raised to the power of 10.

Basically there are two types of input:output structure :-

a) The Dedicated Input and Output.

This system has a limited number of ports and requires special instructions before it can be implemented.

b) Memory Mapped.

This system treats the input : output data just like another memory and is usually a more versatile method. A further advantage is that any portion of the memory may be allocated to input and output.

3.7. Internal Registers.

All microprocessors have the standard registers, such as a programme counter and accumulator.

The important aspects to look for are extra user accessible internal registers, like extra accumulators, index or general purpose registers.

These additional registers can mean programming economy and as a general rule, more registers lead to a more or less flexible system.

3.8. The Instruction Set.

The instruction set should be carefully examined to find out just how flexible the registers are. This set is one of the most important selection criteria and should be well organised, easy to learn and powerful, so that fewer instructions can perform the more complex tasks. It must also be memory efficient and this efficiency will be reflected by the number of memory cycles required to complete a task.

3.9. Execution Speed.

A very important criteria if real time executions are envisaged, in the past the clock frequency was used as a standard, but that led to misleading results. Today the most popular standard is the register to register addition time.

3.10. Interrupt Structures.

a) The simple single line interrupt.
   This method ties all interrupts to a common line and the processor must poll to locate the initiating device.

b) The multilevel interrupt.
   This system has more than one hardware interrupt line and the device is immediately identified without the necessity for polling.

c) Vectored Interrupt.

A very fast system that requests service and then branches automatically to the interrupt service routine.
It is often found that various interrupt methods are combined one microprocessor system.

3.11. Interfacing Complexibility.

Interfacing is a major problem for all processors, especially buffering, however there is an inbuilt strength in the microprocessor family, because a family chip can usually be connected directly without buffering. TTL compatibility and voltage levels are also important.

3.12. Power Supply Requirements.

Single power supplies give good economy with reduced design time, it is important to use family device peripherals because they are always voltage compatible.

3.13. Manufacturer's Hardware Support.

This is a direct measure of difficulty in obtaining information and the manufacturer's commitment to servicing the user's needs. It is important when choosing to look for simulators or emulators in development systems.

3.14. Software Support.

This includes documentation such as programming manuals, application notes and other technical literature. Also look for monitor programmes for assembly, simulation, diagnostics and debugging.

If the microprocessor data sheets have been carefully scanned with the above features in mind then the field of choice should be more or less defined and narrowed down considerably.

The final choice is to choose from the microprocessor chips that do fulfil most of the designer's requirements. It is not possible to satisfy them all and this is where careful thought must be given to the advantages and disadvantages of the final choices.

There now follows a critical appraisal of the microprocessor chips and literature available at the time of initial research.

CRITICAL APPRAISAL & LITERARY SURVEY.

CHAPTER 4.

CHAPTER 4.

A CRITICAL APPRAISAL OF MICROPROCESSOR CHIPS AND A LITERARY SURVEY.

4.1. The 4 Bit Group of Microprocessors.

There were many 4 bit microprocessor devices , but the appraisal is restricted to two of the most popular types, it became immediately evident that the others were unsuitable to the task in hand.

4.2. The 4040 Type of Microprocessor.

The important hardware features include an address stack that has designated seven levels of subroutine nesting, with each level being 12 bits wide. The system also contains index registers that can be used as either twelve 8 bit address registers , or as twenty four 4 bit storage locations.

This processor can directly address 4 K words and the typical register to register addition time is 10.8 μ seconds. There is also a vectored interrupt capability and an instruction set with decimal arithmetic capability, having sixty operating instructions.

4.3. The Rockwell PPS 4 CPU.

This processor has an 8 bit instruction data bus and a separate 12 bit address bus.

An internal stack is provided which can also be expanded to RAM.

Conditional branching is provided by two control flip flops.

An x register is provided for temporary storage of either the accumulator , or the BM register.

Other important features include a 5μsec instruction time, directly addressing sixteen input output circuits, with an instruction set having decimal arithmetic capability with 50 instructions.

The single level interrupt has the capability of being expanded to 15 levels.

4.4. The 8 Bit Microprocessor Group.

The 8 bit microprocessors are by far the most popular at the moment. Seven various types are appraised and the final choice was in fact taken from this important and very resourceful group.

4.4.1. The Motorola 6800 CPU.

This processor has six addressing modes, an excellent 2 μsec register to register addition time with 72 instructions.

It has a maskable vectored interrupt structure and a non maskable interrupt capability.

The system also has decimal arithmetic capabilities and the processor can directly access 64k bytes of memory.

The 6800 system requires only one 5 volt power supply and can drive up to ten family support chips without the need for bus extension.

4.4.2. The Intel 8080A CPU.

The 8080 group and the descendants are very widely used in British industry because it is admirably suited to the field of numerical control.

The 8080 A architecture consists of only one accumulator, but has six 8 bit scratch pad registers and may be used as such, or as three 16 bit registers.

There are also two other register pairs not accessable to the programmer. Other important features include four addressing modes, by an instruction set having decimal arithmetic capabilities with 78 instructions. It boasts an excellent 2 μsec register to register addition time and can address a massive set of 256 input output ports, with a multilevel vectored interrupt structure.

The processor can directly address 64k bytes of memory and these advantages made the 8080A a strong contender for the project in hand.

4.4.3. The Zilog Z80 CPU.

This chip is an enhancement of the 8080 and the major hardware differences consist of the large number of inbuilt internal processor registers. The main registers are all duplicated and the processor has four additional special purpose registers, an interrupt vector register, a dynamic RAM refresh register and two 16 bit index registers, which together with three other pairs makes this so called 8 bit system into virtually a 16 bit microprocessor.

The processor can directly address 64k bytes of memory with ten addressing modes.

The system is controlled by a massive 158 instructions, which if the alternatives were counted singly, as done by some manufacturers, then the number is near to 700 very powerful instructions, which are also upward compatible with the 8080 and only take 1.6 μsec per register to register instruction.

Other important features are the single phase clock, TTL compatibility and the single 5 volt power supply.

4.4.4. The Fairchild F8 CPU.

This is an input output oriented processor and has two 8 bit ports with an onboard clock circuit. The chip contains 64 internal registers which serve as a workspace, making RAM unnecessary for simple applications.

The programme counter and stack registers are unfortunately not on the microprocessor chip and must be provided by a special external ROM. The system has eight addressing modes, with a 2 μsec register to register addition time with 60 instructions. Decimal arithmetic capabilities are provided and a single level vectored interrupt system.

4.4.5. The Signetics 2650 CPU.

This special processor has eight level stack registers with an address adder that calculates relative and indexed addresses.

The most unique feature of this processor is the set of seven general purpose registers. Register zero is thought of as an accumulator and the remaining six secondary registers form two, three register banks.

The system can only address 32k bytes because the programme counter is only 15 bits wide. There are six addressing modes with 72 instructions having a 4.8 μsec register to register addition time. The interrupt system is vectored and the processor has decimal arithmetic capability.

4.4.6. The MOS Technology Inc 6502 CPU.

There are many models of the 6500 series, some contain onboard clocks such as the 6502 and the chip comes in two package sizes, 28 pin and 40 pin. Albeit the 6500 series is frequently thought of as an enhanced 6800, careful investigation of the architecture soon disproves this supposition.

The system has only one accumulator , the 6800 has two, the index register has to be made up of two 8 bit registers and the stack pointer is also only 8 bits wide. The reduced number of instructions, 56 as compared to 72 in the 6800, is complemented by an extra seven addressing modes, making 13 in total. In high speed applications the short 1.0 μsec register to register time can be used to advantage.

4.4.7. The RCA CDP Cosmac 1802 CPU.

The hardware appointments are 16 general purpose 16 bit registers, all can be designated as the programme counter by using the "P" register . The N, P and X point to a register using a 4 bit select line, the "D" register functions as the primary accumulator. This system supports one interrupt level and a distinctive feature is the fabrication by CMOS technology, with high noise immunity. The processor has a single phase clock and 4 addressing modes with 91 instructions , but a long 6 μsec register to register add time.

4.5. The 12 Bit Microprocessor Group.

When investigating there were very few available 12 bit processors, information was scarce and this consideration itself made the group unsatisfactory.

4.5.1. The Intersil 6100 CPU.

A popular processor because it is software compatible with the "PDP 8E" minicomputer . The architecture is clear as all the main registers have 12 bits which can address 4K words , each 12 bits wide . There are 69 instructions with 4 addressing modes , a single level vectored interrupt with a register to register time of 2.5 μsec . The fabrication is by CMOS technology.

4.6. The 16 Bit Microprocessor Group.

4.6.1. The National Semiconductor IPC 16 Pace CPU.

A system with four 16 bit accumulators with zero as the primary, one as the secondary and accumulators 2 and 3 double as secondaries and the index registers. An onboard stack is 16 bits wide and 10 words deep addressing 65k words, each 16 bits.

The processor has 45 instructions with five addressing modes and the register to register time is 8 μsec, there is a multilevel interrupt structure.

4.6.2. The Texas Instruments TMS 9900 CPU.

The 16 bit processor approaches the level of performance of a minicomputer but has only 3 user accessible registers. The workspace is a most unique and powerful programming feature and identifies the first of 16 general purpose registers located in RAM, they can be used as either accumulators, index registers, temporary stores or for buffers, however the drawback is that it cannot operate without a supporting RAM chunk.

This architecture gives the unique feature of variable hardware and software with 64 instructions, which can directly address 32k words each 16 bits. There is a built in multiply divide capability with a register to register time of 4.7 μsec with 15 levels of external interrupts and also a vectored interrupt system.

4.7. Availability and the Final Choice of Processor.

The final choice after removing the obviously unsuitable and the unavailable, depends mainly on the application and also on a good general knowledge of the particular microprocessor. The designer selects to suit his own requirements and also if possible to simplify the design.

Innovations take place daily and a choice can only be made at the actual time into the types readily available, it is not possible to wait too long for further developements, as this produces a vicious circle, leading to time wasting frustration. There were unfortunately not many good systems available, when the choice had to be made and the following were the considerations taken into account when making the difficult decision.

4.7.1. Industrial Standards.

The most widely used processor in British industry at the time was the 8080 family, making this family compatibility almost a must, but future developement was also strongly considered. Further research along these lines showed that some industrial thoughts were being given, but not yet implemented, to the Z80 a far more powerful tool, not only superior but completely software compatible with the 8080 industrial standard. Hence the Z80 was considered to be short listed, from the point of view of industrial standards.

4.7.2. Processing Speed and Power Requirements.

The main processor application was for machine control making execution speed and sampling rate important factors. The speeds ranged from the slow Cosmac at 6 μsec to the fast 6502 at 1 μsec, however the 6502 was sadly lacking in the necessary machine control instructions and the 8 bit index register facility finally placed this chip out of the running.

The Z80 was the next fastest at 1.6 μsec with a massive and powerful set of 158, or 700 expanded, instructions with the majority ideally suited to machine control, with a 16 bit index register pair. The 6800 was next at 2 μsec and a clear set of 72 instructions, giving a choice of two processors in this group.

4.7.3. The Interrupt Structure.

Any serious work in machine control must rely heavily on the interrupt structure, unless the processor is to be completely tied up with servicing. Processors should be free to perform other tasks without waiting for a signal, such as an emergency stop, yet obey it when it occurs, the interrupt routine takes care of both of these conditions.

Some applications require various interrupts of varying importance, answering each in the correct order, this is called a priority queueing system. Investigation of the various processors regarding interrupts was found to be very disappointing.

The Cosmac and F8 types had only a single level of interrupt structure, the 2650 had only a vectored interrupt, the 8080 had only a multi level vectored interrupt, but the 6800 and the 6502 both had a vector interrupt and a non maskable system, all of these processors were sadly lacking, from the point of view of interrupt structure, the only processor having all the required interrupt attributes was the Z80.

The Z80 processor has the non maskable interrupt of the 6800 and 6502, the vector interrupts of the 6800, 6502 and 8080 and also an unique vector system for polling many interrupts of varying importance.

4.7.4. Indexing and Registers.

By now the choice had been made to use an 8 bit processor, due to cost, availability, documentation and other factors, hence a careful investigation was required of the 8 bit types, to ensure that at least some of the advantages of the 16 bit machines could be utilised.

A pleasant surprise was found when investigating one of the 8 bit processors. The 6502 was the worst having only an 8 bit stack pointer and even the index registers were made up from two 8 bit registers. On the other hand the Cosmac had an unique hardware appointment of 16 general purpose 16 bit registers , hence it was ideally suited for this task. Unfortunately the Cosmac failed on too many other requirements , such as interrupts and time.

The next best was the Z80 which scored heavily in this section due to the unique advantages , which included two 16 bit index registers , a 16 bit stack pointer and a very unusual arrangement of 8 bit register pairing. Three pairs of registers could be combined to produce three 16 bit registers. These six registers have individual opcode instructions , allowing them to be fully utilised as six 8 bit registers and also another set of instructions to control them in the 16 bit mode. There are also a full set of duplicate registers with equal control.

This large amount of 16 bit facilities make this 8 bit processor virtually into a 16 bit machine , for many applications.

4.8. Literary Survey.

When this research work began, it was soon realised that the amount of literature on the open market, was very limited indeed and because the development had originated in the United States of America, obtaining literature required contacting importers or United Kingdom representatives, who at that time were very few and far between. There was also a scarcity of books on the subject and in fact the book situation has only improved during the 1980's. All the books stated in the bibliography were obtained during the very late stages of this research work. Luckily any one now following this work will find a good selection of quality books on the subject.

The poor state of available literature greatly affected the choice of microprocessor chip and is also reflected by the author's own attempt to slightly rectify the situation by even writing articles and publishing his own book on the subject (1), this venture being very successful and resulting in an international second edition due to many requests from Germany and Holland for copies of the publication.

Details now follow of the Z80 microprocessor which was the author's final choice of controlling device.

Z80 MICROPROCESSOR

CHAPTER 5.

CHAPTER 5.

THE Z 80 MICROPROCESSOR SYSTEM.

5.1. The Final Choice.

After appraising the advantages and disadvantages of the various types of microprocessors , the correct system to use became quite clear and the final choice was made to utilise the Zilog Z 80 microprocessor chip , this system consists of three main components :-

A) The Central Processing Unit.

B) The RAM and ROM Memories.

C) The Interface circuits to the Peripheral devices.

5.1.1. The Central Processing Unit.

This is the heart of the microprocessor based system, the function of which is to obtain instructions from a memory section and perform the preprogrammed operations. The memory is utilised to hold the data and instructions until they are processed.

Figure 5.1. indicates the internal structure of the Z 80 central processing unit and shows the major elements and registers , these registers will be explained in turn.

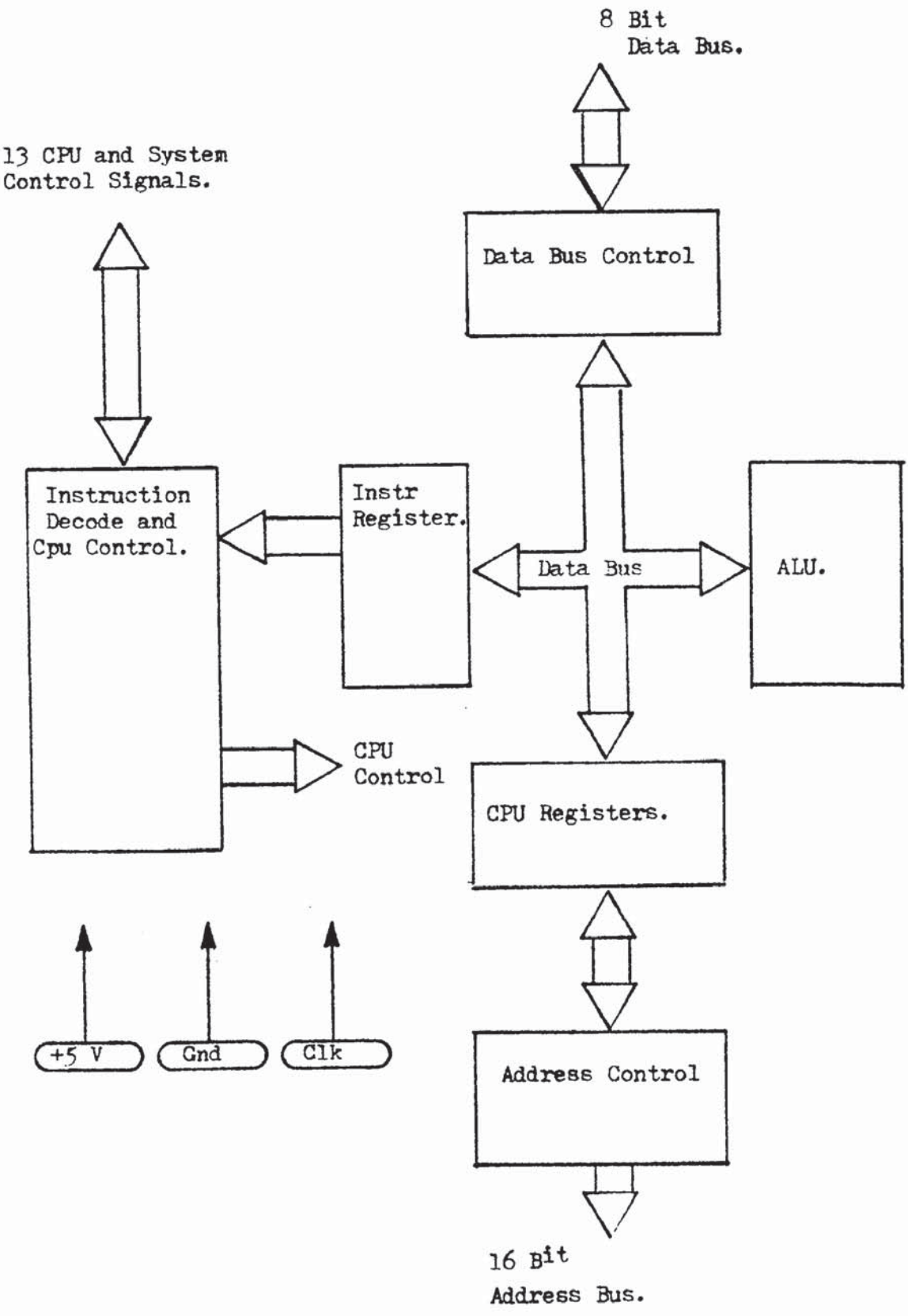


Fig. (5.1.) Z80 CPU Architecture.

5.2. The Z80 Special purpose Registers.

5.2.1. The Programme Counter.

The programme counter holds the 16 bit address of the current instruction being fetched from the memory , it is automatically incremented after the contents have been transferred to the appropriate address lines, however a programme jump will overwrite the value stored in this counter and execution then continues from the new value stored in the counter.

5.2.2. The Stack Pointer.

The stack pointer holds the 16 bit address of the current position of the top of the stack memory , this location is within the system's own Random Access Memory. This external stack memory is organised as a last in and first out or LIFO data file. The data can be pushed onto the stack memory from a specific central processor register and popped off again from the stack back into the register when required for further use. The data popped off is always the last data that was pushed on.

The stack allows simple implementation of multi level interrupts with unlimited subroutine nesting and the simplification of many types of data manipulation.

5.2.3. Two Index Registers. IX and IY.

These are two independent index registers that hold a 16 bit address, used for indexed addressing modes.

The index register is used as a base to point to a region in memory, from which data can be retrieved.

An additional byte is included in the indexed instruction to specify the displacement from this base.

5.2.4. Interrupt Page Address Register. I.

The Z80 CPU can be operated in a mode, where an indirect call to any memory location can be achieved in response to an interrupt.

The I register is used to this end, it stores the high order 8 bits of the indirect address, while the interrupting device provides the lower 8 bits of the address.

5.2.5. Memory Refresh Register R.

The Z80 CPU contains a memory refresh counter, to enable dynamic memories to be used, with the same ease as static type memories. This 7 bit register is automatically incremented after each instruction fetch.

The data in the refresh counter is sent out on the lower portion of the address bus together with a refresh control signal, while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation.

5.2.6. Accumulator and Flag Registers.

The CPU includes two independent 8 bit accumulators, with their associated 8 bit flag registers.

The accumulator holds the result of 8 bit arithmetic or logical operations, while the flag register indicates specific conditions for 8 or 16 bit operations, such as indicating if the result of an operation was equal to zero.

5.3. CPU Internal Registers.

5.3.1. General Purpose Registers.

There are two sets of general purpose registers, BC, HL, and DE with their complementaries BC', HL', and DE'. Each set contain six 8 bit registers that may be used individually as 8 bit registers, or as three 16 bit register pairs.

At any one time the programmer can select either set of registers to work with, through single exchange commands, which transfers the entire set.

In systems where fast interrupt response is required, one set of general purpose registers and an accumulator with a flag register may be reserved for handling the fast routine and another set may be utilised in the main programme. Only one simple exchange command need be executed to exchange all the registers.

This facility greatly reduces the interrupt service time by eliminating the requirement for saving and retrieving the register contents in the external stack during the interrupt or subroutine processing.

5.3.2. Arithmetic and Logic Unit. (ALU).

The arithmetic and logical instructions of the CPU are executed in the ALU.

Internally the ALU communicates with the registers and the internal or external data bus. The functions performed by the ALU include :-

ADD	SUBTRACT	LOGICAL AND
INCREMENT	COMPARE	LOGICAL OR
DECREMENT	SET BIT	LOGICAL EXC OR
ROTATES	RESET BIT	TEST BIT
LEFT SHIFT	RIGHT SHIFT	

5.3.3. Instruction Register and CPU Control.

As each instruction is fetched from the memory it is placed in the instruction register and then decoded. The control section performs this function and then generates and supplies all the necessary control signals to read or write data to and from the registers.

5.4. Z80 CPU Pin Description.

The Z80 CPU [4] is packaged in the industrial standard 40 pin dual in line package. Figure 5.2 shows the input and output pin configuration and the function of each pin is described below :-

5.4.1. The Address Bus. (A_0 to A_{15}).

The pins are designated by A_0 to A_{15} and the chip constitutes a 16 bit tristate active high output, making up the 16 bit address bus. This bus provides addresses for memory up to 64 Kbytes (where 1Kbyte = 1024) . The bus also provides for data exchange within the system and for input and output data exchange with an external peripheral.

The input , output addresses use the lower eight address bits to allow the user to directly select up to 256 input or output ports. A_0 is the least significant address bit and during the refresh time the seven lower bits contain a valid refresh address.

IC 37 MK 3880 Z80 CPU.

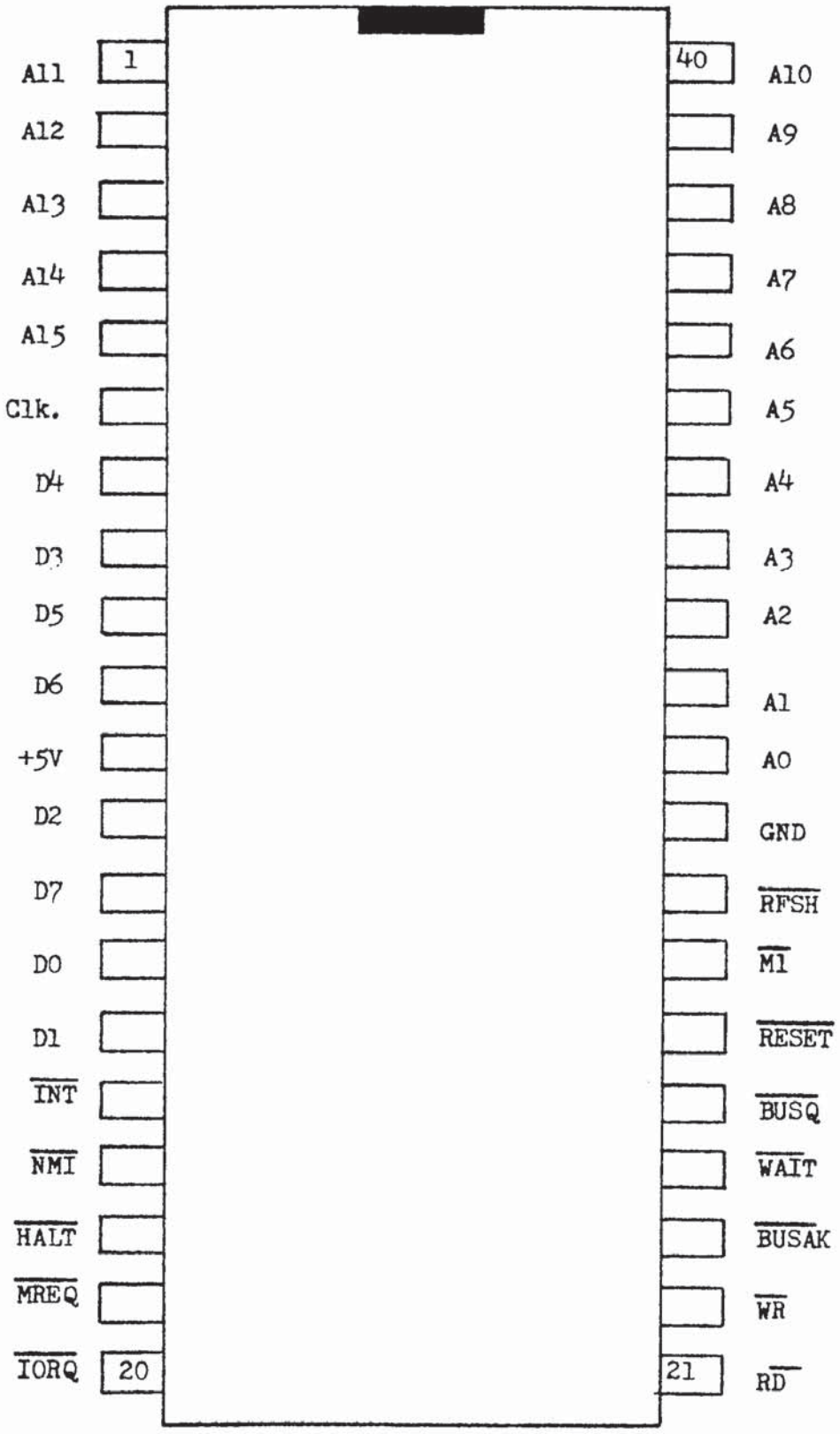


Fig. (5.2.)

5.4.2. The Data Bus. (D_0 to D_7)

The data bus is used for data exchanges with the memory or any input output device.

It constitutes of an 8 bit bidirectional tristate set of input output lines, which are active high.

5.4.3. Machine Cycle. ($\overline{M_1}$)

This is an output line which is active low and indicates that the current machine cycle is the "Op Code Fetch Cycle", of an instruction execution.

The M_1 signal is generated for a two byte opcode as each opcode is fetched.

M_1 also occurs with the " $\overline{IORQ}$ " signal to indicate an interrupt acknowledge cycle.

5.4.4. Memory Request. ($\overline{MREQ}$)

The memory request signal "MREQ" is a tristate active low output, which indicates that the address bus holds a valid address for a memory read or write operation.

5.4.5. Input , Output Request. ($\overline{MREQ}$)

The IORQ signal is a tristate output, active low, which indicates that the lower half of the address bus holds a valid I/O address for an input/output read or write operation.

An IORQ signal is also generated with an M1 signal, when an interrupt is being acknowledged, to indicate that an interrupt response vector can be placed on the data bus.

Interrupt acknowledge operations occur during M1 time, while the input/output operations never occur during the M1 time.

5.4.6. Memory Write. ($\overline{WR}$)

The signal is a tristate, active low output, indicating that the CPU data bus holds valid data, to be stored in the addressed memory, or input/output device.

5.4.7. Memory Read. ($\overline{RD}$)

The signal is a tristate active low indicating a CPU request for data from memory or an input output device, it is used to gate data on to the CPU data bus.

5.4.8. Refresh. ($\overline{RFSH}$)

The signal is an active low output indicating that the lower 7 bits of the address bus contains a refresh address for dynamic memories, the "MREQ" signal is used to refresh read the memories.

5.4.9. Halt State. ($\overline{HALT}$)

The signal is an active low output indicating that the CPU has executed a software "HALT" and is awaiting either a non-maskable or a maskable interrupt with mask enabled. While halted the CPU executes "NOPS" to maintain the refresh activity.

5.4.10. Wait. ($\overline{WAIT}$)

This active low input signal indicates that the addressed memory or input output devices are not ready for data transfer. The CPU continues to enter "Wait" states as long as this signal is active, to allow devices of any speed to be synchronised. No refreshing occurs.

5.4.11. Interrupt Request. ($\overline{INT}$)

This is an active low input generated by an external device, the request is honoured at the end of a current instruction if the internal software controlled interrupt enable flip flop [IFF] is enabled and if the "BUSRQ" is not active.

When the CPU accepts the interrupt, an acknowledge signal "IORQ during M_1 time", is sent out at the beginning of the next instruction cycle. The CPU can respond to an interrupt in three different modes.

5.4.12. Non Maskable Interrupt. ($\overline{NMI}$)

The signal is a negative edge triggered input. The "NMI" request line has a higher priority than the "INT" signal and is always recognised at the end of a current instruction independent of the status of the interrupt enable flip flop. The "NMI" is received and automatically forces the Z80 CPU to restart to memory location 0066 hexadecimal.

The programme counter is also automatically saved in the external stack so that the user can return to the programme that was interrupted. Continuous "WAIT" cycles can prevent the current intruction from ending and a "BUSRQ" signal will override the "NMI" signal.

5.4.13. Reset. ($\overline{\text{RESET}}$)

This signal is an active low input which forces the programme counter to zero and initialises the CPU by:-

a) Disabling the interrupt enable flip flop.

b) Setting the I register to address 0000.

c) Setting the R register to 0000.

d) Setting the "Mode 0" interrupt.

Note however that the PIO is NOT reset. During the reset time the address and data buses go to a high impedance state and all output control signals go to the inactive state with no memory refreshing.

5.4.14. Bus Request. ($\overline{\text{BUSRQ}}$)

The bus request signal is an active low input and is used to request the CPU address bus, data bus and the tristate output control signals, to go to a high impedance state so that other devices can control these buses.

When "BUSRQ" is activated the CPU will set these buses to a high impedance state, as soon as the current CPU machine cycle is terminated.

5.4.15. Bus Acknowledge. ($\overline{\text{BUSAK}}$)

The bus acknowledge signal is an active low output used to indicate to the requesting device that the CPU address, data and tristate control bus signals have all been set to their high impedance state and that the external device can be used to control those signals. No memory refreshing occurs and the CPU requires communicating devices to assist in this operation and control, it also requires an oscillator or clock to keep all the operations under strict time control.

Figure 5.3 shows the oscillator that is utilised to clock the central processor unit at the frequency of 2 MHz and this oscillator consists of two invertors in a 74S04 (3) hex inverter integrated circuit, driven by and controlled with a 16 MHz quartz crystal for frequency stability. The output drives a 74LS163 (3) synchronous binary counter chip that provides outputs at 1 MHz , 2 MHz and 4 MHz which may be selected to drive the central processor at the desired frequency. In this design the 2 MHz oscillator output was selected because this matched the speed capabilities of the available support memory devices.

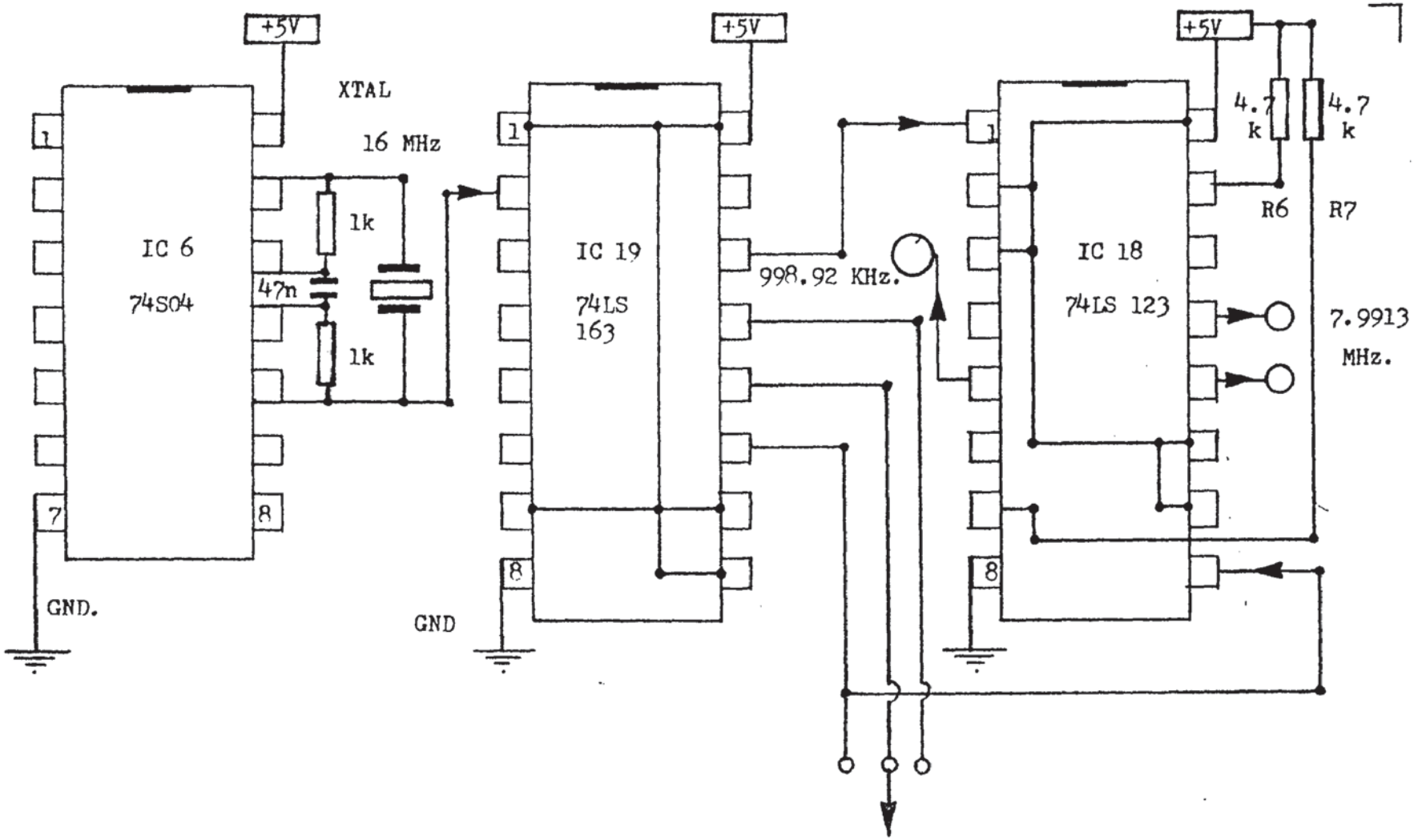


Fig. (5.3.) CPU CLOCK OSCILLATOR.

5.5. The CPU Peripherals.

Immediately after being reset the Z80 central processor fetches the first instruction from memory location 0000 , hence there must be some Read Only Memory at this location so that the contents can be preprogrammed but not corrupted by any error on the part of the operator. It is thus usual to position the monitor or administration programme at locations 0000 , 000I etc . This system has the firmware located in two 1 K EPROMS at addresses 0000 to 03FF hexadecimal and at 0400 to 07FF hexadecimal.

Figure 5.4 shows the circuit which is connected directly external to the central processing unit , this circuit controls the data, address and all other lines.

Figure 5.5 shows the monitor EPROMS connected to the address and data bus. These EPROMS [5] [6] are selected by a combined signal derived from the $\overline{RD}$ and $\overline{MREQ}$ signals and the address lines of A_0 and A_{11} .

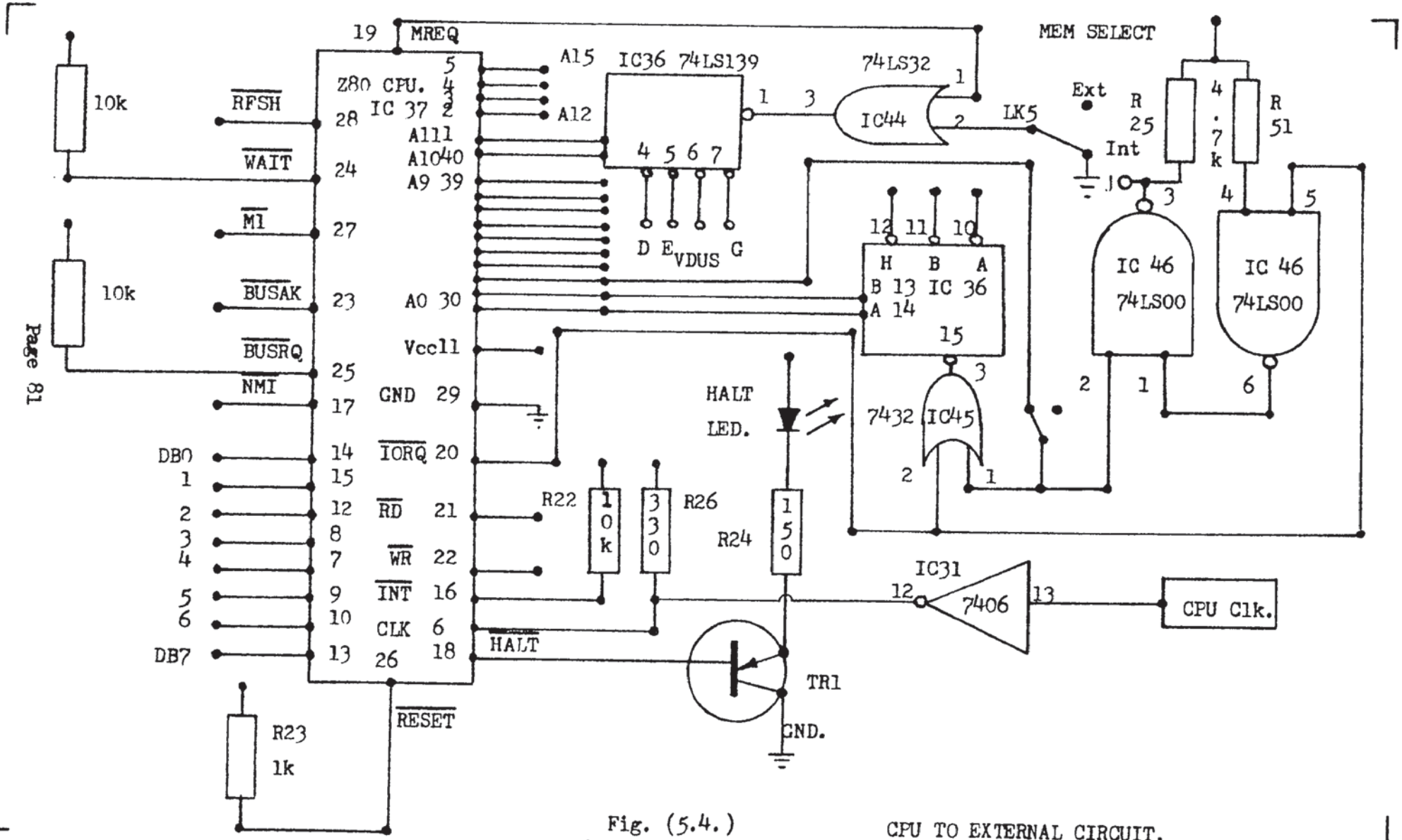


Fig. (5.4.) CPU TO EXTERNAL CIRCUIT.

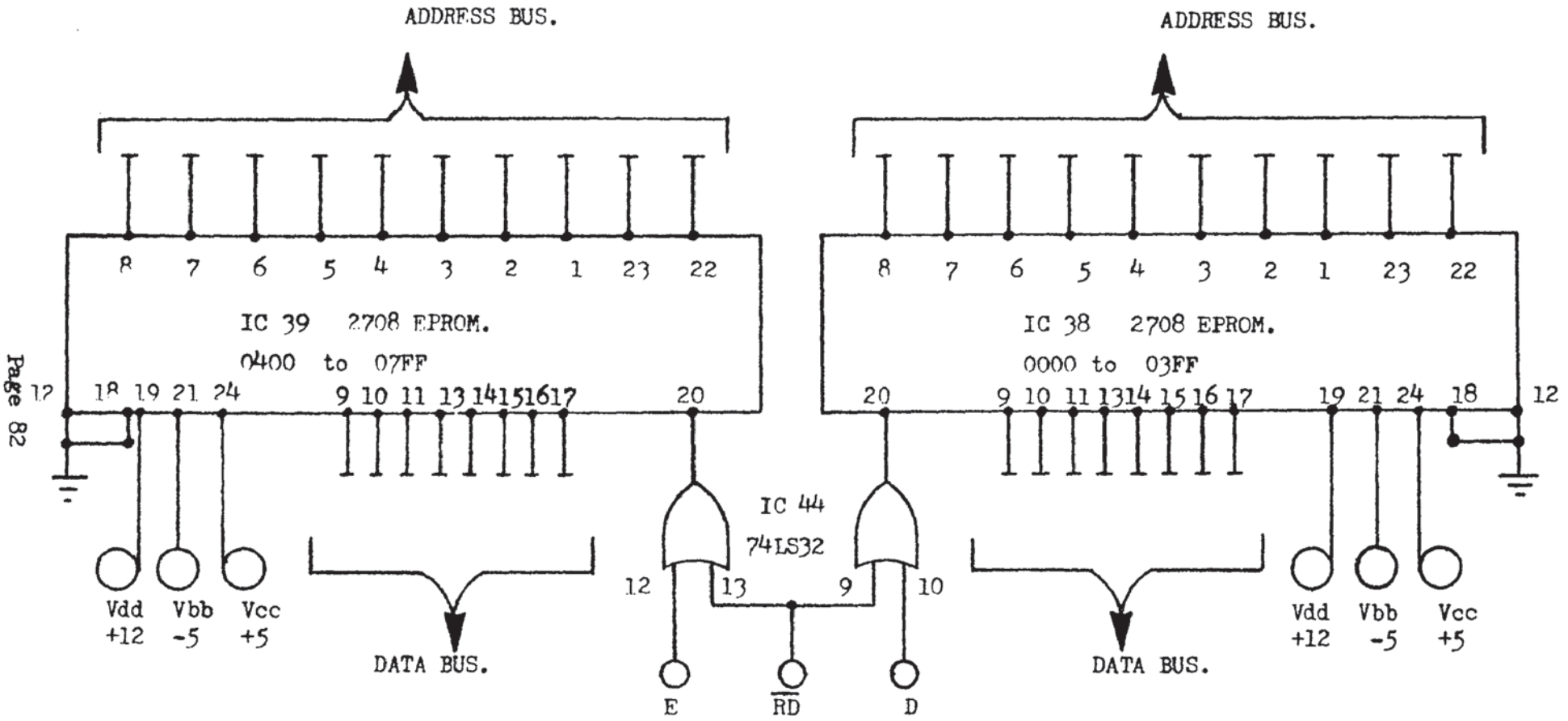


Fig. (5.5.)

EPROM TO ADDRESS & DATA BUS.

5.6. Visual Display.

A microprocessor system of this type requires that the operator be presented with visual data feedback as programme developement may require the presentation of several thousands of characters, also the cost of displaying these alpha numeric characters should be minimized. Hard paper copies are not always required but only confirmation of correct typing, or the verification of a particular memory location.

The method chosen for this research work is a "Memory Plane Peripheral" and is not sited at the ports as in conventional input output methods. It consists of integrated circuit logic blocks which share a section of the system memory, this block is designed to present a radio frequency modulated composite video signal to a standard domestic television set, in such a way that the contents of the memory section, is presented and interpreted as standard characters. The CPU is given absolute priority and this prevents any possible access conflict . The screen is blanked off during the CPU access time.

The system operates by mapping a section of memory onto the visible screen plane, the position of the symbol is a function of the address in memory. The symbol itself is a function of the least significant 7 bits of data at that particular memory location.

This section of screen memory is called the "Video RAM" and is operated by switching the memory chip address lines between the CPU address bus and a counter divider chain so that the hardware continuously cycles the address lines to the memory. Jamming is prevented by placing a transmission gate between the data bus and the output pins.

Being able to utilise a standard domestic television set reduces the cost from at least £600 for a VDU, down to about £80 for a small screen television set. Figure 5.6 indicates the layout of a typical RAM and gate connection utilised in this design to control the flow of video data.

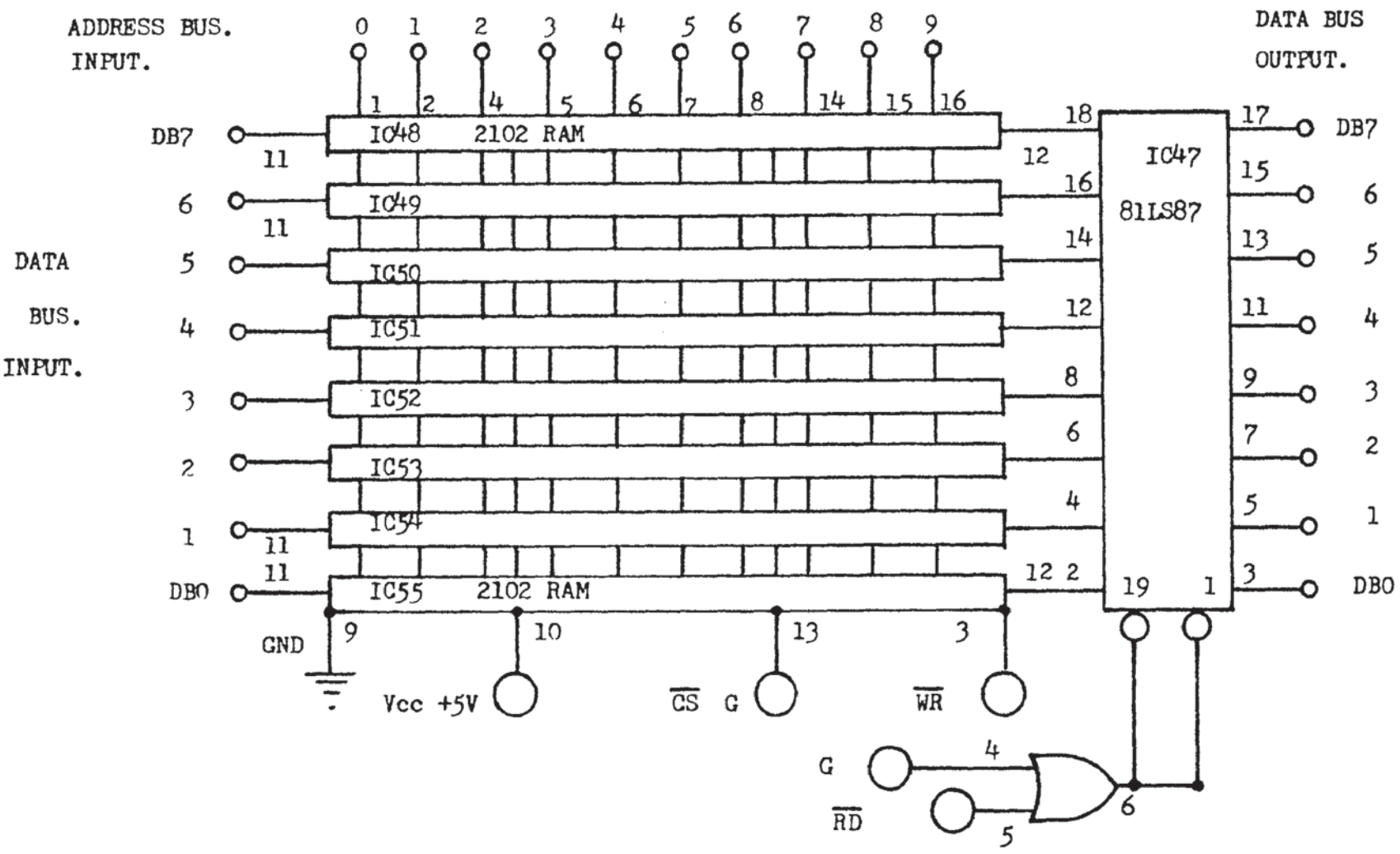


Fig. (5.6.) 2102 RAM DECODING.

5.7. The Character Generator.

Each address for the video RAM is latched and used to address a large ROM called the "Character Generator".

The output of this ROM has been pre-programmed to provide the video dot pattern, of part of a character, depending on the particular television raster row, or the character.

A composite video output is available consisting of video blanking pulses, frame and line synchronising pulses and the data from the character generator.

Details of the electronic input and output to the character generator are given in the following diagram, figure 5.7.

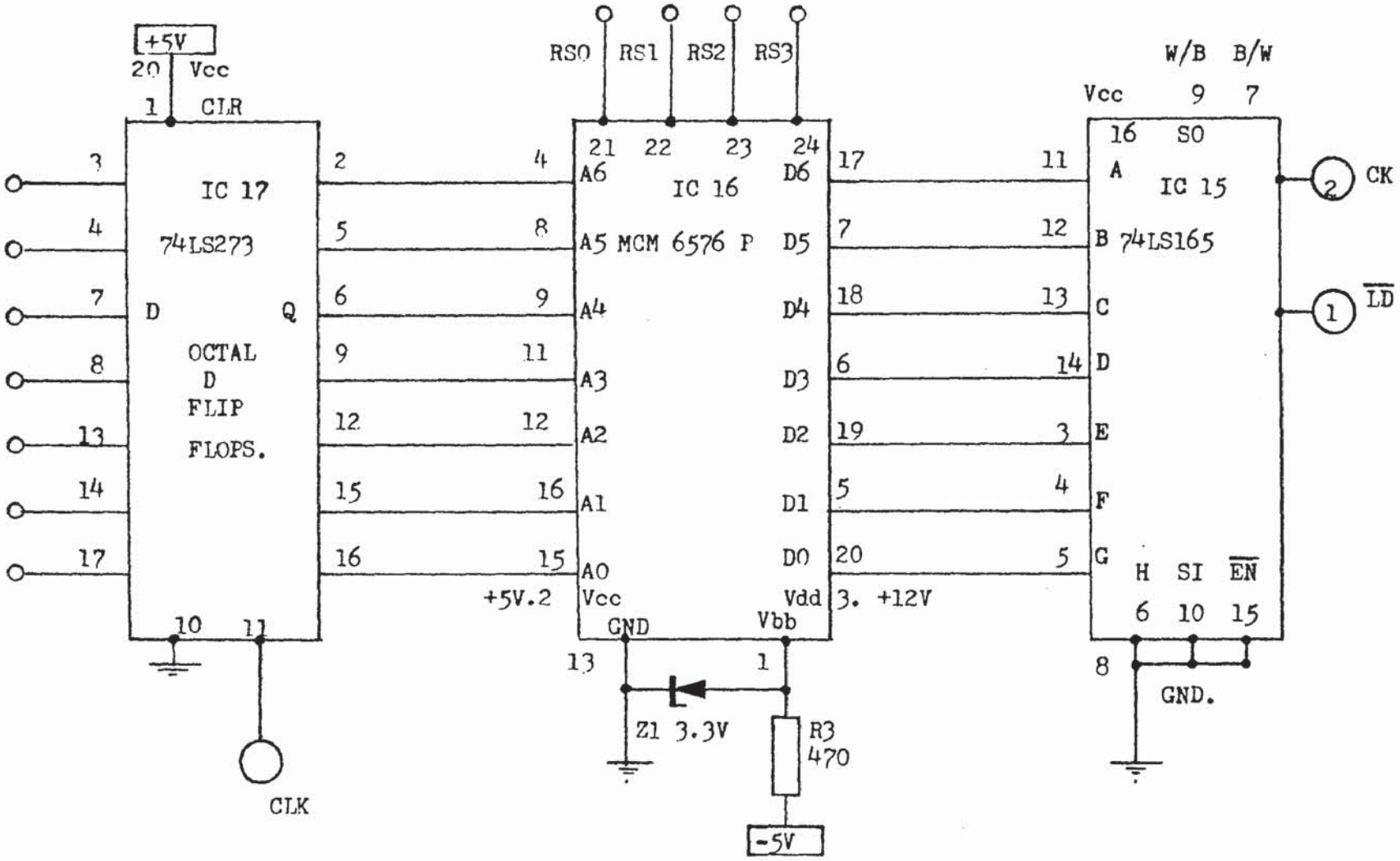


Fig. (5.7.) CHARACTER GENERATOR & GATES.

5.8. The Keyboard.

The keyboard is arranged to look like a standard set of "QWERTY" characters, however no switches are used to obtain key closures. The action occurs by using the magnetic saturation of a pulse transformer in each key, thus avoiding mechanical contacts, giving virtually unlimited life without contact bounce.

Electronically the keyboard is arranged as a single port peripheral with a port address of "0". Hardware realisation is by utilising two integrated circuit packages to obtain latched outputs with gated inputs, thus 14 lines are available to port "0". The 14 lines consist of 8 input and 6 output lines.

Further use has been made of the output lines by choosing a 6 bit latch and using only two of them for the keyboard, these two lines drive the clock and reset inputs of a counter decode package, whose outputs are connected to columns of keys.

Keyboard details follow in the next diagram, figure 5.8. and figure 5.9 shows the reverse side of the electronics of the keyboard and microprocessor controller.

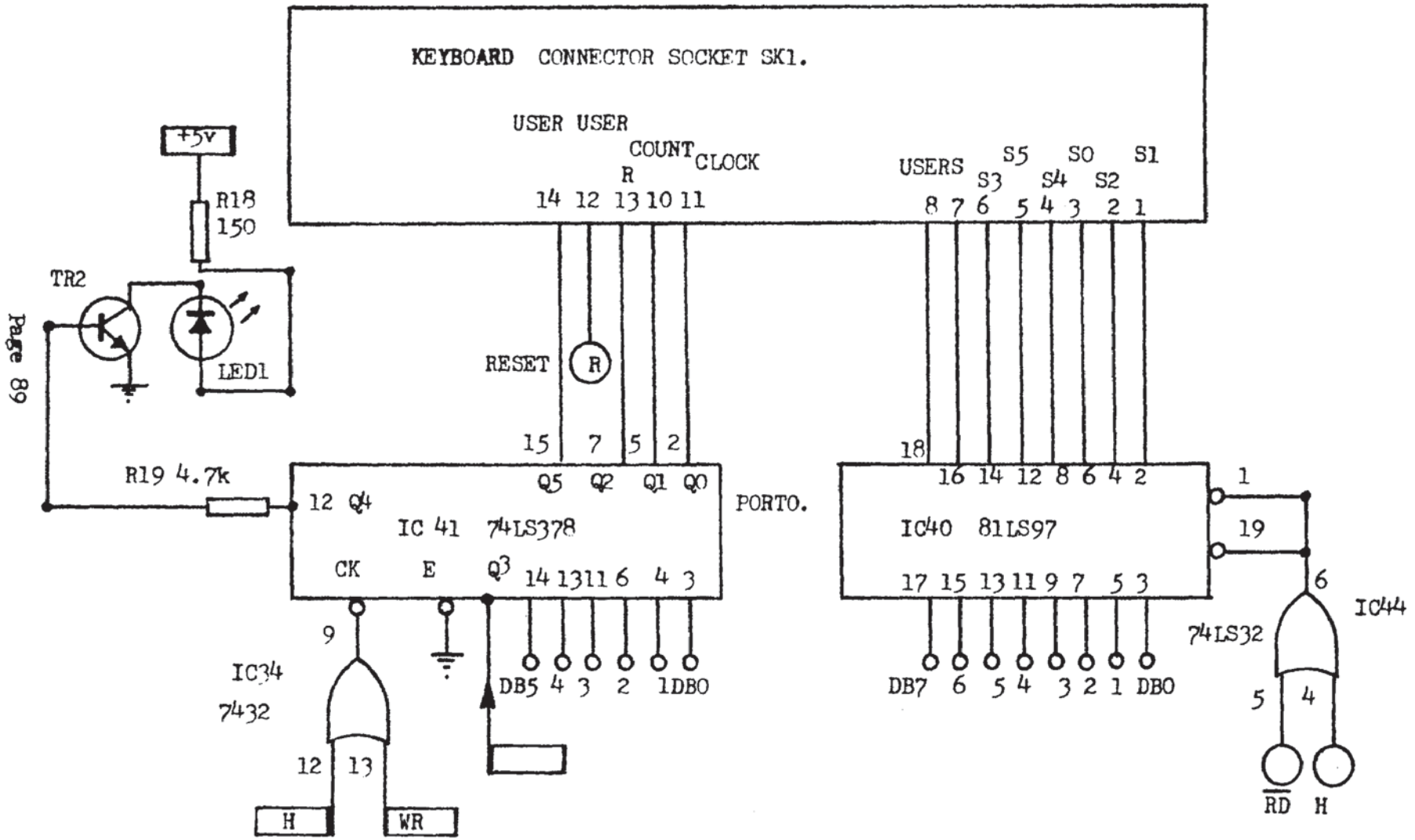


Fig. (5.8.) KEYBOARD PORT INPUT.

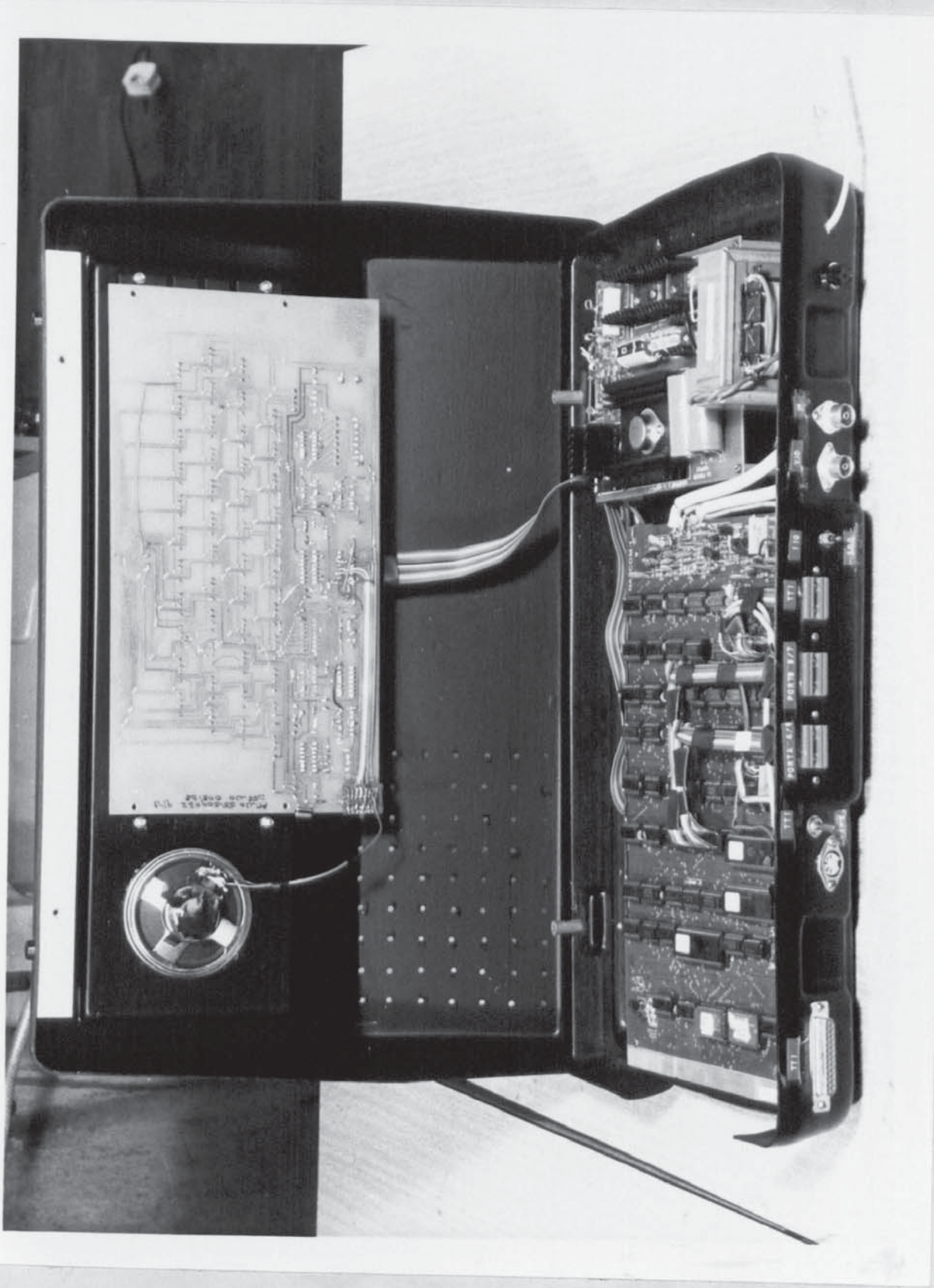


Fig.(5.9.) BEHIND THE KEYBOARD.

The columns of keyboard keys form a matrix, which is completed by the 6 row pulse sensing lines, each driving a transistor amplifier, which is connected via the output flip flops and the keyboard cable to the port input transmission gate.

The output command signals to port "0" cause the data bus to be latched, while the input commands, cause the keyboard row lines to drive the data bus.

The CPU thus has the opportunity to determine the key that was pressed and the software in the monitor programme performs the function of contact bounce elimination, albeit with this particular keyboard, the problem is already eliminated. The CPU also has the task of determining the change of state and the hexadecimal code assignment of each key, according to the position in the effective 8x6 matrix, which it simulates.

The next two diagrams give details of the keyboard decode and timing in figures 5.10, 5.11 and details of the monitoring characteristics are now considered.

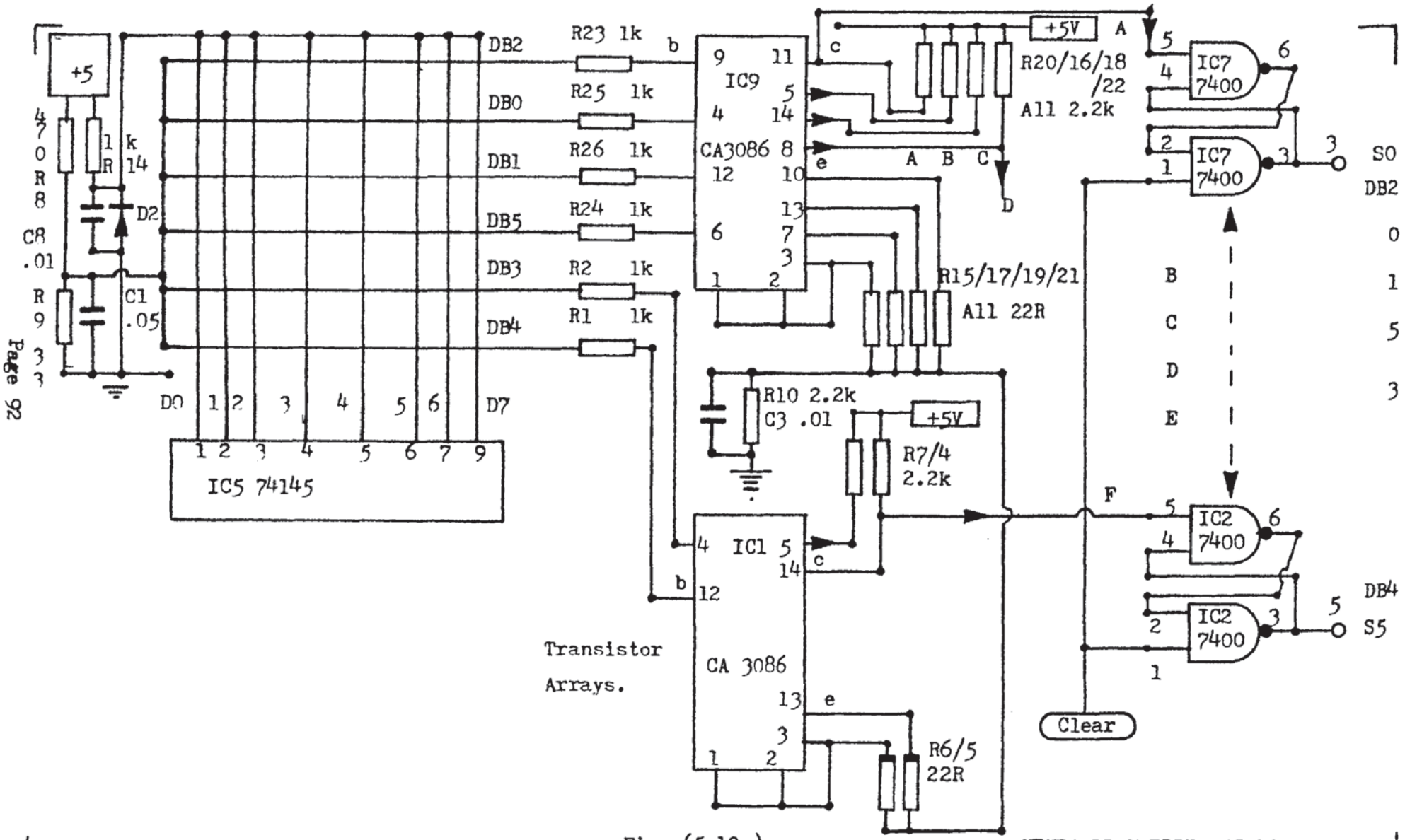


Fig. (5.10.) KEYBOARD MATRIX DECODE .

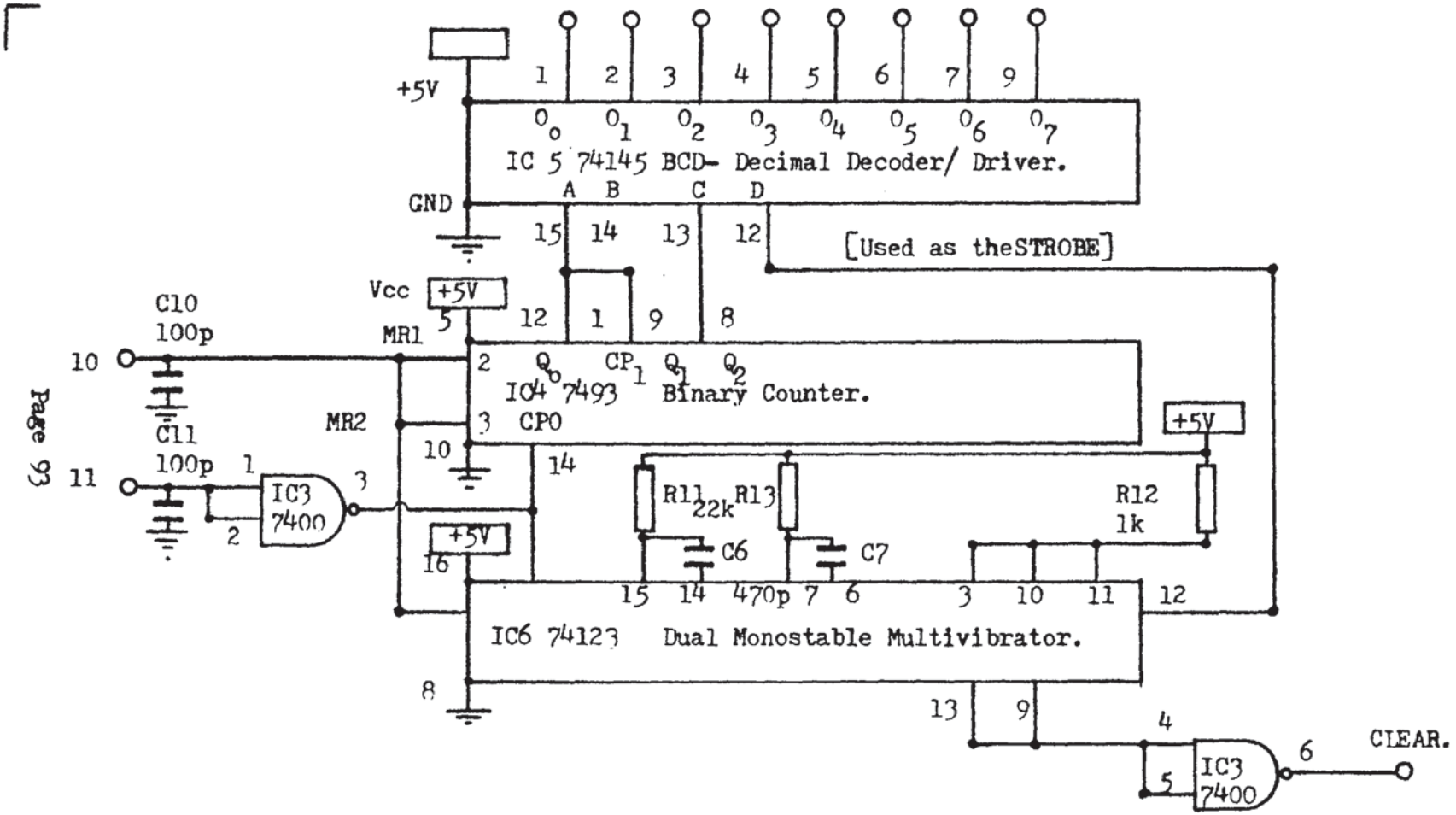


Fig. (5.11.) KEYBOARD TIMING CIRCUIT.

THE MONITOR SYSTEM.

CHAPTER 6.

CHAPTER 6.

THE MONITOR SYSTEM.

6.1. The Memory Search or Modify Key. (M).

The monitor system or administration programme has been inserted into two 1K byte EPROMS of the 2708 type (4) (5) which reside at memory locations ØØØØ to 07FF hexadecimal. Their programme execution is controlled from the keyboard as follows :-

To operate the microprocessor :-

1. First press the RESET key, this initialises the system. Note however that if the VDU is used to replace the TV monitor then extra instructions are required, these may be found in section 10.6.3
2. Next press the M key and the spacebar which is the large bar across the bottom of the keyboard.
3. Now type ØC8Ø ensuring that ZERO is used, it lies on the top right of the keyboard and has a stroke through the Ø.
4. Next press the special NEWLINE key on the middle right.

The Screen should now show :- ØC8Ø 43 (or some other pair)

The ØC8Ø is the memory location being M (Modified) and the 43, or some other pair, is the contents of that location, which may be any pair of hexadecimal digits. From now on every "O" found in the programme should be understood to mean " Ø " on the keyboard.

If two hexadecimal digits such as "2A" are typed followed by the "Newline" key, then the system will automatically insert the "2A" into memory "0C80" and overwrite the previous contents. The screen should then show:-

0C81 A3 (The A3 may be any pair of hex digits)

if another pair of hexadecimal digits such as "B2" are typed followed by the "Newline" key , then again the contents of memory location "0C81" will be changed to "B2".

This procedure may be continued until all the required memories have been changed, to terminate the procedure a ● (Full stop) must be pressed followed by the "Newline" key, the system would respond and terminate the memory changing procedure. It is then possible to check on the memory entries as follows:-

Press RESET (The hidden key)

Press M space bar 0C80 Newline

The screen shows:-

0C80 2A Press Newline

0C81 B2 Press Newline

0C82 -- Press Newline

0C83 .. Press Newline

Where -- and .. are the previously entered hexadecimal digits.

To terminate press ● Newline

T.___ (Tabulate or Type command.)

The previous block of memory may be listed in one operation by using the "T" command, however the memories can only be looked at with this command, they cannot be changed.

Press RESET

Press T spacebar 0C80 spacebar 0C90 Newline

The screen should show:-

0C80 2A B2 CC 34 5D AA 76 98

0C88 34 F5 66 23 87 53 CB 5A

0C90 EA E4 56 78 64 24 DF FA

The first four numbers of each line are the memory locations and the hexadecimal pairs following are the contents of the memories, eg 0C80 contains 2A, 0C81 contains B2, 0C82 contains CC.

The numbers on the screen may differ because these are only examples.

A final reminder, the "T" key can only look at the memories, the "M" key must be used to modify them.

E.__ (Execute the programme command.)

An inserted programme may be executed [RUN] as follows, a sample programme is included in the chapter on machine code.

Press E spacebar 0C80 Newline

The programme will now execute starting with the first instruction written at memory "0C80", or any other location following the "E" command.

On this particular machine it is possible to start from any address in the range "0C80 to about 0FE0", the last address depends on how much of the system stack has been used. The programme must also be terminated correctly, here are some examples:-

a)	76	Halt.
b)	DF 5B	Return to the system monitor.
c)	C7	Reset.
d)	E7	Breakpoint.

The best method is to end with "DF 5B", this gives a safe return to the monitor system.

B. (Breakpoint command)

If a programme is executed by the "E" command , it may be stopped by one of the correct terminations or by the "B", breakpoint command as follows.

Press B spacebar 0D20 Newline

Press E spacebar 0C80 Newline

If a correct programme has been inserted starting at location 0C80 then the system will execute that programme, but will stop automatically at location 0D20.

Note that the breakpoint address MUST be the start address of an instruction.

Eg 0D20 C3 40 0F

It would NOT be possible to stop at 0D21 or 0D22, because they are both in the middle of the three part instruction "C3 40 0F".

The screen would then show the contents of all the registers, when execution stops, of the instructions from 0C80 up to but not including the step at 0D20.

The meaning of the screen display can be found under the details for the "S" command.

S. The Single Step Command.

The "S" or single step command allows the operator to step through a programme by one instruction at a time.

To operate the command :-

Press S Spacebar 0C80 Newline

The processor would execute only the instructions written at location 0C80, which could be 1, 2, 3 or 4 opcodes long. It would then stop automatically at either 0C81, 0C82, 0C83 or 0C84, depending on how many opcodes there were in that particular instruction.

The display would show the contents of the CPU registers at that point in the programme and further executions would be obtained by simply pressing the "Newline" key, for every required operation.

The screen display is an important monitor feature because it allows the user to examine the contents of the CPU registers after each step and if an error occurs, then it becomes apparent at that stage before continuing. The error is corrected by using the memory modify "M" key as explained previously.

Single Step Visual Display.

All the numbers shown are examples and can differ each time, but the layout and meaning are identical.

Stack Pointer	Program Counter	A . F	H . L	D . E	B . C	I	I Index	Y Index	Flags
		Registers							
1000	0C80	D3 42	2C 16	AB B5	DD A4	56	AD F6	34 5E	ZC

Fig (6.1.) Single Step Visual Display.

6.2. Monitor Routines.

The monitor system EPROMS have been programmed to allow the user access to the inbuilt system commands. These subroutines can save the user writing many standard programmes of his own.

The command system subroutines fall into two main categories :-

6.2.1. Restart Instructions.

The "Restart" instructions may be obtained by calling a routine with one of the Z80 one byte "RST" codes, as shown on the following page. To use the subroutines it is simply required to enter:-

Eg: C7 as a programme instruction, the system will recognise this code and call the correct subroutine automatically.

The monitor would interpret this particular coded instruction " C7 " as a restart to location " 0000 ", in the monitor EPROM and would thus reset the CPU to reinitialise the command system.

There now follows a list of available " RST " restart command codes.

Monitor Restart Instructions.

Opcode.	Assembler.	Name.	Function.
C7	RST 0H	START	To reset the CPU.
CF	RST 8H	RIN	Wait for a keyboard input.
D7	RST 10H [Eg D7 25]	RCAL	A relative subroutine call, which will call the subroutine at the displacement following the D7.
DF	RST 18H [Eg DF 65]	SCAL	A direct subroutine call, which must be followed by the code number of the required subroutine.
E7	RST 20H	BRKPT	If E7 is placed at a programme location, then a breakpoint occurs at that programme step [It stops]
EF	RST 28H [Eg EF 54 48 45 00] = the	PRS	All characters following the EF will be printed on the screen, a "00" will terminate the instruction.
F7	RST 30H	ROUT	The character in the accumulator will be displayed on the screen.
FF	RST 38H	RDEL	A wait period will occur, depending on the contents of A.

6.2.2. Subroutines.

The routines may be called by inserting the appropriate machine code following the "DF" call instruction.

The actual monitor system commands such as " E " for " Execute " or " T " for " Tabulate ", may also be called, by placing the equivalent ASCII code figures following the "DF" call.

Eg: To call the execute "E" command DF 45. The ASCII code for "E" is 45 and when the programme comes to the command " DF " , it will automatically look up the meaning of the command letter code " 45 " and obey the command " E " to execute.

The following pages list the other calls that can be used without writing the routine, simply by utilising one of the automatic bank of calls.

Monitor Subroutines.

Code.	Name.	Function.
DF 5B	MRET	Used to end a programme.
DF 5C	SCALJ	To call a routine, the address must be at location 0C0A.
DF 5D	TDEL	Wait for about 1 second.
DF 5E	FFLP	Flip bits in port 0.
DF 5F	MFLP	Turn the tape drive on or off.
DF 60	ARGS	Load the registers. HL = ARG 1 DE = ARG 2 BC = ARG 3
DF 62	IN	Scans the keyboard for an input, but does not wait, sets the carry flag if there was an input.
DF 63	INLIN	Obtain an input line and "Blink".
DF 64	NUM	Examine the input line and convert a hexadecimal value to a binary value.
DF 66	TBCD3	Output the value in the HL registers in ASCII followed by a space.
DF 67	TBCD2	Output the value in accumulator A in ASCII and add A into the register C.
DF 68	B2HEX	Output the value in the accumulator in ASCII.

Monitor Subroutines.

Code.	Name.	Function.
DF 69	SPACE	Output one space.
DF 6A	CRLF	Output a Carriage return:Line feed.
DF 6B	ERRM	Output an "Error" message.
DF 6C	TXI	Output the contents of the HL registers in ASCII, then a space, then the contents of the DE registers, then a space.
DF 6D	SOUT	Send a string of characters directly to the RS 232 serial port.
DF 79	RLIN	Examine an input line and convert up to ten hexadecimal values, from ASCII to Binary.
DF 7A	BlHEX	Output the least half of the contents of accumulator A in ASCII.
DF 7B	BLINK	Blink the "Cursor" and wait for an input from the keyboard.
DF 7C	CPOS	If HL are set to a position on the screen, then HL will be set to the address of the first character, on that line.

Table Changing Commands.

Code.	Name.	Function.
DF 71	NOM	If the HL registers are set to the address of the new output table, then the table is automatically changed for output routines.
DF 72	NIM	If the HL registers are set to the address of the new input table, then the table is automatically changed for input routines.
DF 77	NNOM	This call resets the output table back to the normal call address.
DF 78	NNIM	This call resets the input table back to the normal call address.

It is also necessary to know the routine numbers to place at these tables.

All routine numbers must be placed after the code "DF".

The input and output routine numbers follow.

6.3. Input Routine Numbers.

CODE.	NAME	FUNCTION.
DF 61	KBD	Scans the input keyboard.
DF 70	SRLIN	Scans the serial input port.
DF 74	XKBD	Scans the external ASCII keyboard. See also the "X" commands.
DF 76	UIN	User specified input routine.

The above can be placed in your own table, or used after the DF code.

6.4. Output Routine Numbers.

Code.	Name.	Function.
DF 65	CRT	To display on the monitor screen.
DF 6F	SRLX	Gives an output to the serial port.
DF 6E	XOUT	Gives an output to an external ASCII device.
DF 75	UCUT	An output to the user's specified output routine.

It is also possible to call routines that are actually command routines themselves.

For example, the "W" command is to "Write" to the cassette receiver tape. This command can be called within a programme by calling the "ASCII" equivalent code for "W", which is "57", hence to call the "W" command in a programme, the instruction "DF 57" is used.

6.4. Output Routine Numbers.

Code.	Name.	Function.
DF 65	CRT	To display on the monitor screen.
DF 6F	SRLX	Gives an output to the serial port.
DF 6E	XOUT	Gives an output to an external ASCII device.
DF 75	UOUT	An output to the user's specified output routine.

It is also possible to call routines that are actually command routines themselves.

For example, the "W" command is to "Write" to the cassette receiver tape. This command can be called within a programme by calling the "ASCII" equivalent code for "W", which is "57", hence to call the "W" command in a programme, the instruction "DF 57" is used.

6.5. System Workspace.

The area of memory from " 0C00 to 0C7F " is utilised by the system monitor, as a workspace area, thus giving location " 0C80 " as the first free available RAM location for user operation.

The memory locations " 0C00 to 0C6A " are initialised by the system monitor to contain " 00 " , after the reset command. The second section from " 0C6B to 0C7D " is initialised from a table of values kept within the system monitor EPROM.

Locations " 0C7E and 0C7D " are also initialised and the following tables indicate the function of each location, within the workspace area.

The user may choose to make use of, or even alter certain values within the workspace table, to good advantage.

System Workspace.

Address.	Length.	Name.	Function.
0C00	1	Port 0	Copy of the current port 0.
0C01	9	KMAP	Map of the keyboard state.
0C0A	1	ARG C	Last processed command.
0C0B	1	ARG N	Number of values in input line.
0C0C	2	ARG 1	First entered value.
0C0E	2	ARG 2	Second entered value.
0C10	2	ARG 3	Third Value.
0C12	12	ARG 4-9	Fourth to ninth values.
0C1E	2	ARG 10	Tenth value.
0C20	1	NUM N	Number of characters examined.
0C21	2	NUM V	Value returned by NUM.
0C23	2	BRK ADR	Breakpoint address.
0C25	1	BRK VAL	Stored value from BRK ADR.
0C26	1	CON FLG	Zero, or -1 for "E" command.
0C27	1	SK OPT	Keyboard option.
0C28	1	SX OPT	X option.
0C29	2	CURSOR	Position of the cursor.
0C2B	1	ARG X	Last command letter.
0C2C	53	MON STK	Monitor stack.
0C61	2	R BC	Restore area for BC registers.

System Workspace. [Continued.]

Address.	Length.	Name.	Function.
0C63	2	R DE	Register save area for DE.
0C65	2	R HL	Register save area for HL.
0C67	2	R AF	Register save area for AF.
0C69	2	R PC	Programme counter save area.

The workspace is table initialised from here on.

0C6B	2	R SP	Stack pointer area.
0C6D	2	SK TABL	Length of keyboard table.
0C6F	2	SK TAB	Start of keyboard table.
0C71	2	SS Tab	Start of routine addresses. [They actually start at 82H beyond this point.]
0C73	2	S OUT	Start of the output routines.
0C75	2	S IN	Start of the input routines.
0C77	3	SU OUT	Jump to user's specified input routine.
0C7D	3	S NMI	Jump to the NMI routine.

6.6. "X" or External Command.

This command provides comprehensive capabilities for communicating with an external device, through the serial input and output ports.

Options are chosen simply by typing in "X" followed by one of the code letters shown below. The option is then automatically configured by the monitor system. It is also possible to configure "Odd or Even Parity" with the same command.

If the number after "X" ends in zero, the parity is even and if the number ends in one, then the parity is odd.

The Options :-

Even.	Odd.	Function.
X 0	X 1	Support a terminal in full duplex with every typed character sent back. Line feed is automatic after carriage return.
X 10	X 11	The same but with no line feed.
X 20	X 21	Support a terminal in half duplex, but no characters sent back. Line feed is automatic after carriage return.
X 30	X 31	The same but with no line feed, giving the option of a half duplex terminal.

6.7. The Z80 Controller.

The interface boards must be directly accessed by the CPU which is under the operator's control.However it is first necessary to understand the architecture of this particular design, before contemplating writing any of the control programmes.

The processor must be operated in "Machine Code" only, the assembly language mnemonics are only used for convenience,to help to explain the system, but can not be used by this machine, due to the limited memory.

The assembly mnemonics used are those of "Zilog UK", who are manufacturers of the Z80 chip. The full list of mnemonics and machine codes may be found in the appendix and are reproduced by kind permission of Zilog.

6.8. Memory Map of the System.

It is important for the operator to know exactly where the various memories of the operating system lie and which is RAM, ROM or EPROM. The following diagram shows the structure of the memory. (Figure 6.2.)

Programmes may be written into the hexadecimal memory locations of RAM, from " 0C80 to about 0FFF ", but care must be taken at the top end, because the monitor system uses some of the memories from " 0FFF " downwards, for the operating stack memory.

Under mormal conditions it is safe to use the locations " 0C80 to 0FE0 " , leaving " 0FE1 to 0FFF " free for the stack. However a stricter check must be kept if the programme is used with many nested subroutines, or if the instruction " PUSH " is used many times.

Nested subroutines and " PUSH " will place more data onto the stack and it is possible for the stack area to come down below " 0FE0 " and thus run into the programme.

THE SYSTEM MEMORY MAP.

SIZE	LOCATION	DETAILS	UTILISATION
1k 1k	0000 to 03FF 0400 to 07FF	Contained in the 2708 EPROM and used by the system monitor control.	MONITOR ROM.
1k	0800 to 0BFF	Used to display the system monitor commands and screen programme output.	VIDEO RAM.
1k	0C00 to 0C7F 0C80 to 0FFF	Shared by the system for tables and reflections. Shared memory to be used for programmes tables. The top part is used by the system for the stack.	SYSTEM RAM. AND SYSTEM STACK.
60k	1000 to FFFF	The locations may be used for memory expansion. Memory is NOT FITTED in this location, in the original unit.	MEMORY EXPANSION. NOT YET FITTED.

Fig. (6.2.)

6.9. ASCII Characters.

All the screen characters used in this system are formed by the standard international "ASCII" set, as shown in the next diagram, there is also a set of special symbols.

ASCII is the acronym for :-

A American.

S Standard.

C Code. (for)

I Information.

I Interchange.

The system has been interfaced to operate :-

a) A standard cassette recorder, as a cheap method of storage.

b) A standard UHF domestic TV, for visual display of data.

c) An ASR 33 printer, for a hard paper copy.

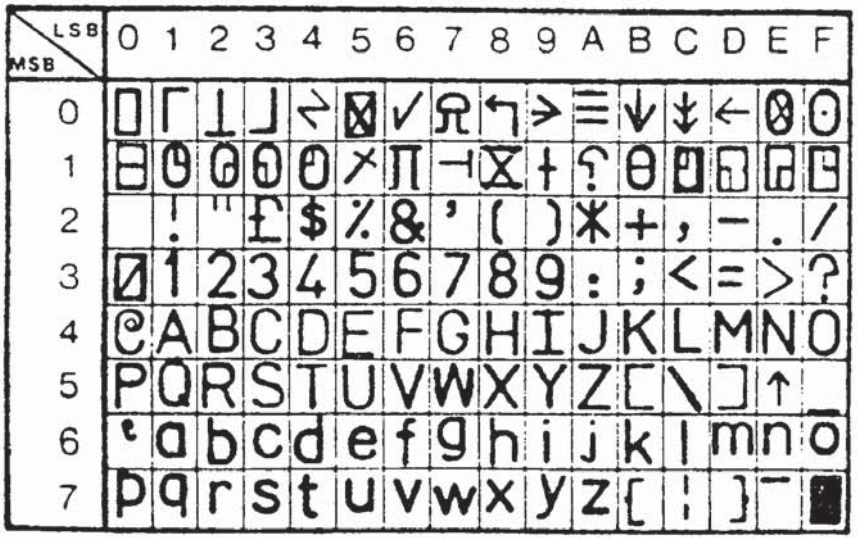
d) A VDU, for better displays and graphics.

e) The RS 232 voltage level serial input and output system.

f) The 20 mA current loop system.

All the above systems utilise the following ASCII set.

The following diagram is taken from (1), figure 6.3.



LSB / MSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2		!	"	£	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	↑	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	¯	■

Eg A = ASCII 41 [4 Down and 1 Across.]
9 = ASCII 39
d = ASCII 64
? = ASCII 3F

CHARACTER SET. [ASCII CHARACTER SET.]

ASCII *

A American
S Standard
C Code (for)
I Information
I Interchange.

Fig. (6.3.) The International ASCII Set.

6.10. The Memory Mapped Television Screen Display.

The system has been interfaced to operate a standard unmodified UHF domestic television set, to enable the monitor commands to be displayed, it may also produce simple graphics.

The monitor will display one unscrolled title line with 15 lines of 48 characters below the title. This has been achieved by utilising the television screen as a mapping of a configured memory plane peripheral.

The position of each symbol on the screen is a function of the address contained in the video RAM, the symbol itself is formed by the least significant 7 bits of data, placed into that particular memory location.

The screen locations are controlled from video RAM at memory locations "080A to 0BF9" , as shown by the following diagram. Further details may be found in the chapter on "Sample Test Routines", where examples may be found on how to display the screen characters.

The following diagram is taken from (1), figure 6.4.

0	0BCA	cb	cc	cd	ce	cf	d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	da	----------	f6	f7	f8	0BF9
1	080A	0b	0c	0d	0e	0f	10	11	12								1a		36	37	38	0839
2	084A	4b	4c	4d	4e	4f	50	51									5a			77	78	087
3	088A	8b	8c														9a				b8	08B
4	08CA	cb															da					08F
5	090A																1a					093
6	094A																5a					097
7	098A																9a					09B
8	09CA	cb	cc	cd	ce	cf	d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	da	----------	f6	f7	f8	09F
9	0A0A																1a					0A3
10	0A4A																5a					0A7
11	0A8A																9a					0AB
12	0ACA	cb															da					0AF
13	0B0A	0b	0c														1a		-		38	0B3
14	0B4A	4b	4c	4d	4e	4f	50	51									5a			77	78	0B79
15	0B8A	8b	8c	8d	8e	8f	90	91	92								9a	----------	b6	b7	b8	0BB9

Fig. (6.4.) MEMORY PLANE SCREEN LOCATIONS.

6.11. Interfaces to the VDU and Television Set.

The Z80 system as designed and built allows for the connection of many peripherals, but each must be operated under the existing standards of communication. The standards used in the design are:-

a) RS 232 voltage level serial output and input data streams.

b) 20 mA current loop serial output and input data streams.

c) A composite video output at 1 volt level.

d) The UHF modulated output to British TV standards.

e) The UART device operating on 244.14 bits per second.

f) A cassette output consisting of a 1.95 KHz modulated tone, conveying serial data.

g) A cassette input, tone detector.

h) A parallel input and output PIO device.

The microprocessor control of the Ward lathe was envisaged as a part of a Direct Numerical Control system and the availability of existing units are now given.

MICROPROCESSORS IN A DNC CELL.

CHAPTER 7.

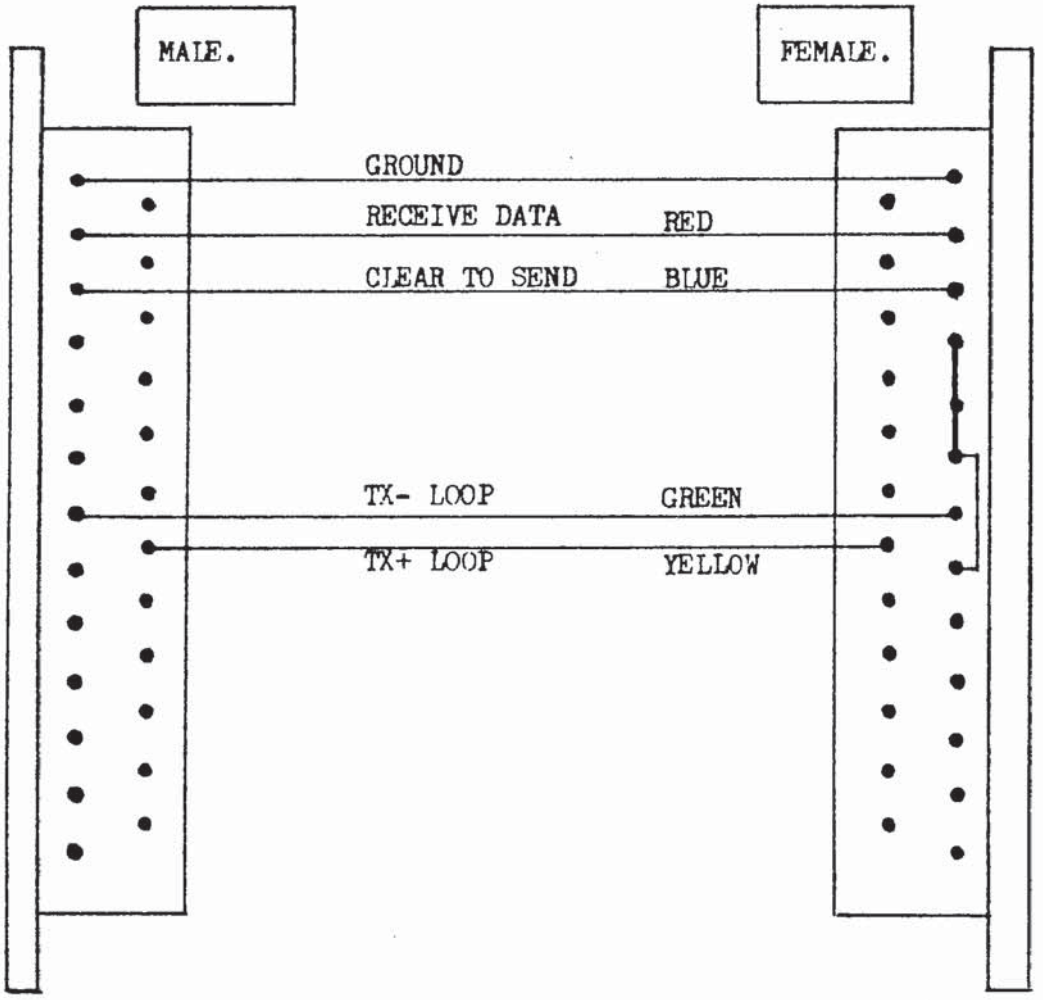
CHAPTER 7.

THE MICROPROCESSOR IN A DNC CELL.

7.1. Equipment Standards.

When starting the work on the microprocessor control system it was decided to investigate the availability of laboratory electronic equipment and their standards with a view to building up a Direct Numerical Control Cell with interconnected units, the new devices could then be built to match these standards.

The investigation showed that each subunit within the laboratory had been constructed as a complete system, but as yet no attempt had been made to interface one piece of equipment to another, hence no common standard existed. The first task was to decide on the electronic standard to utilise and obtain interface cable sets to this format. The next diagram, figure 7.1 shows the cable sets thus obtained with their connections to the double standard of the RS 232 and the current loop systems. Where possible each interconnection between two types of equipment was made utilising these cable sets. There then follows a list of available equipment in the laboratory for interconnecting into the proposed cell.



CANON GB 7926 [SOURIAU 8630]
DB 25P [93C 25

MCME 0607 .

Fig. (7.1.) CABLE PAIR.

7.1.1. Availability of Equipment.

a) ASR 33 teletypes.

b) Type 43 teletype.

c) A Mini Nova computer.

d) A micro Nova computer.

e) A standard domestic television set.

f) A standard cassette recorder.

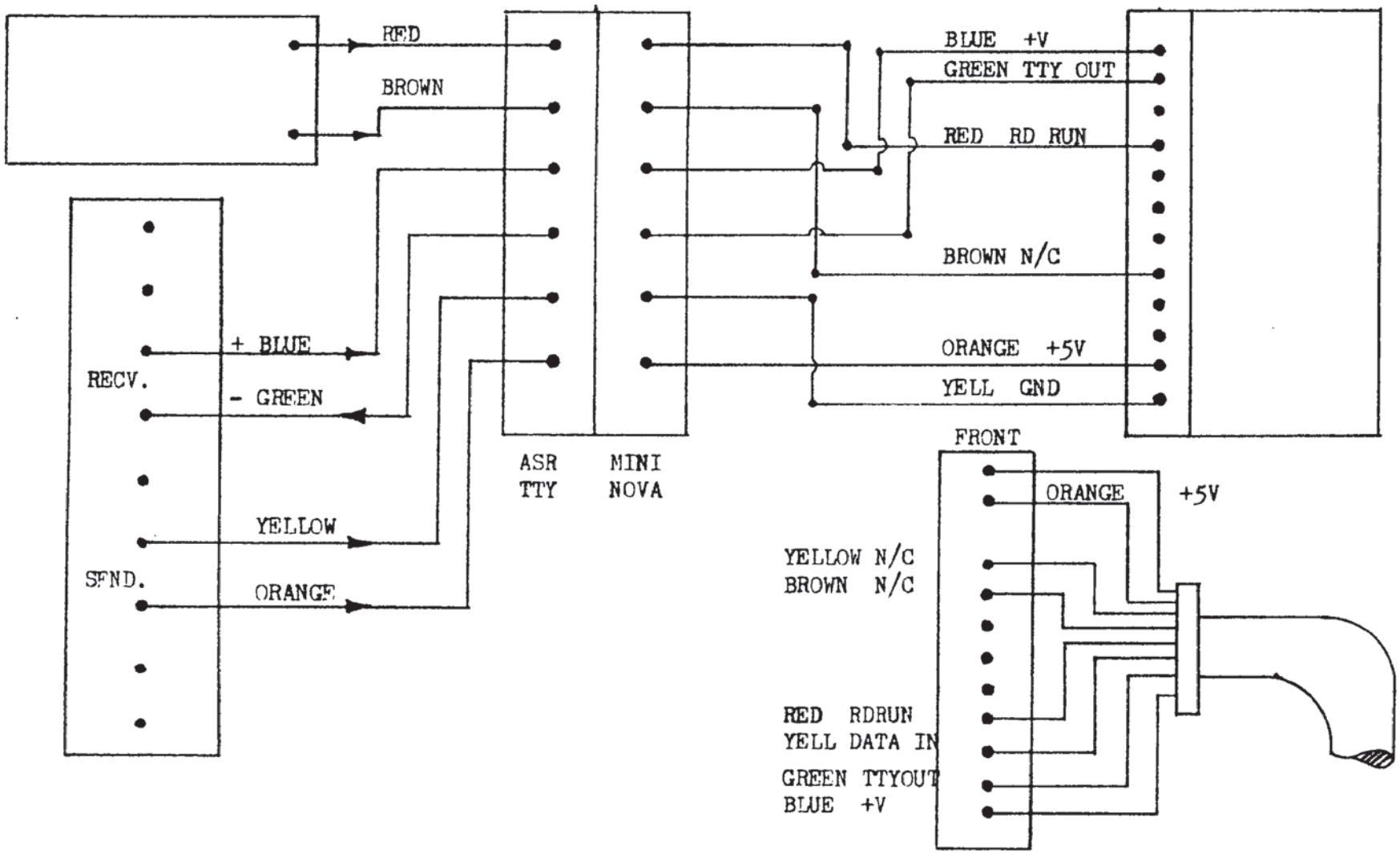
g) An experimental Z80 constructed microprocessor.

Some of the above equipment had been purchased and built specifically for the project in hand.

A start was made to interface each system into an integral layout, by connecting two of the ASR 33 terminals as follows.

a) ASR 33 terminal to the Mini Nova computer.

This interface consisted of simply wiring two of the standard connectors into the output block of the ASR 33 teletype and into the backpanel wiring of the Mini Nova. The following diagram, figure 7.2, shows details of the interconnection.



ASR TELETYPE TO MINI NOVA.

Fig. (7.2.)

b) ASR 33 Terminal to the Z80 Microprocessor.

The standard cable set was used to connect the output block of the ASR 33 terminal , however the Z80 microprocessor side required an interface circuit to be constructed so as to match the serial output from the Z80 device to the ASR 33 terminal.

The circuit shown in figure 7.3 ensures that all inputs and outputs conform to the RS 232 voltage operating standard. The input side consists of a voltage clipper , two diodes, with a 7406 integrated circuit buffer inverter that gives the correct level of serial output to the UART . The UART 's serial output is passed through a 7406 buffer inverter which drives two dc coupled transistors giving the necessary RS 232 output levels. Figure 7.3 illustrates these points and shows the electrical connections between the Z80 device and the ASR 33 terminal.

7.1.2. Teleterminal to the Micro Nova.

This interface consisted of interconnecting cables between the teletype output socket and the backpanel wiring slot of the Micro Nova computer. The first three interfaces were not directly connected to the Z80 microprocessor work in the control system but enabled valuable experience to be obtained, regarding voltage levels and various system requirements. This knowledge was found to be invaluable later and enabled the laboratory equipment to be compatible with a larger cell of control devices . Figure 7.4 shows the connections for the teleterminal and Micro Nova computer.

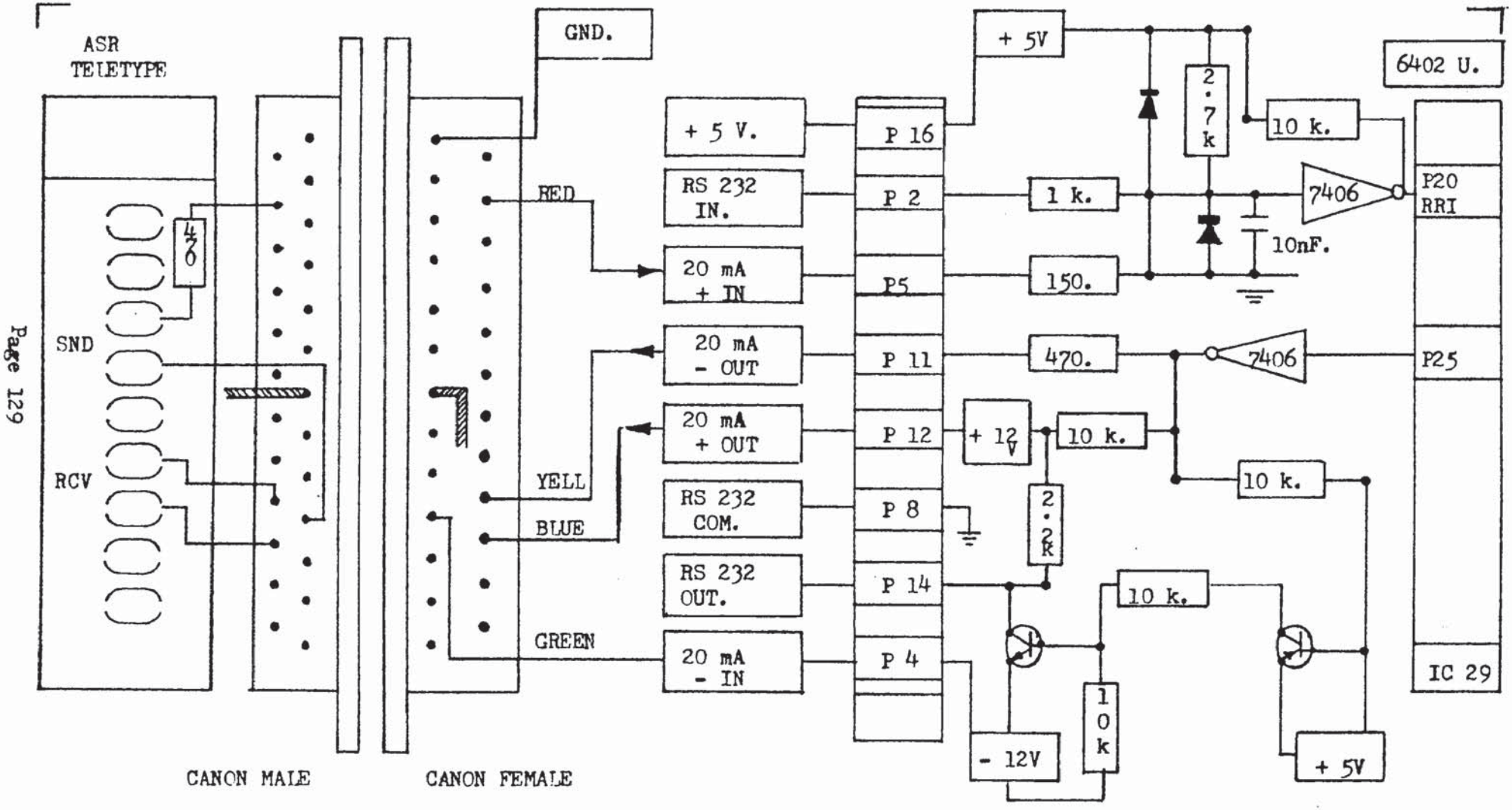
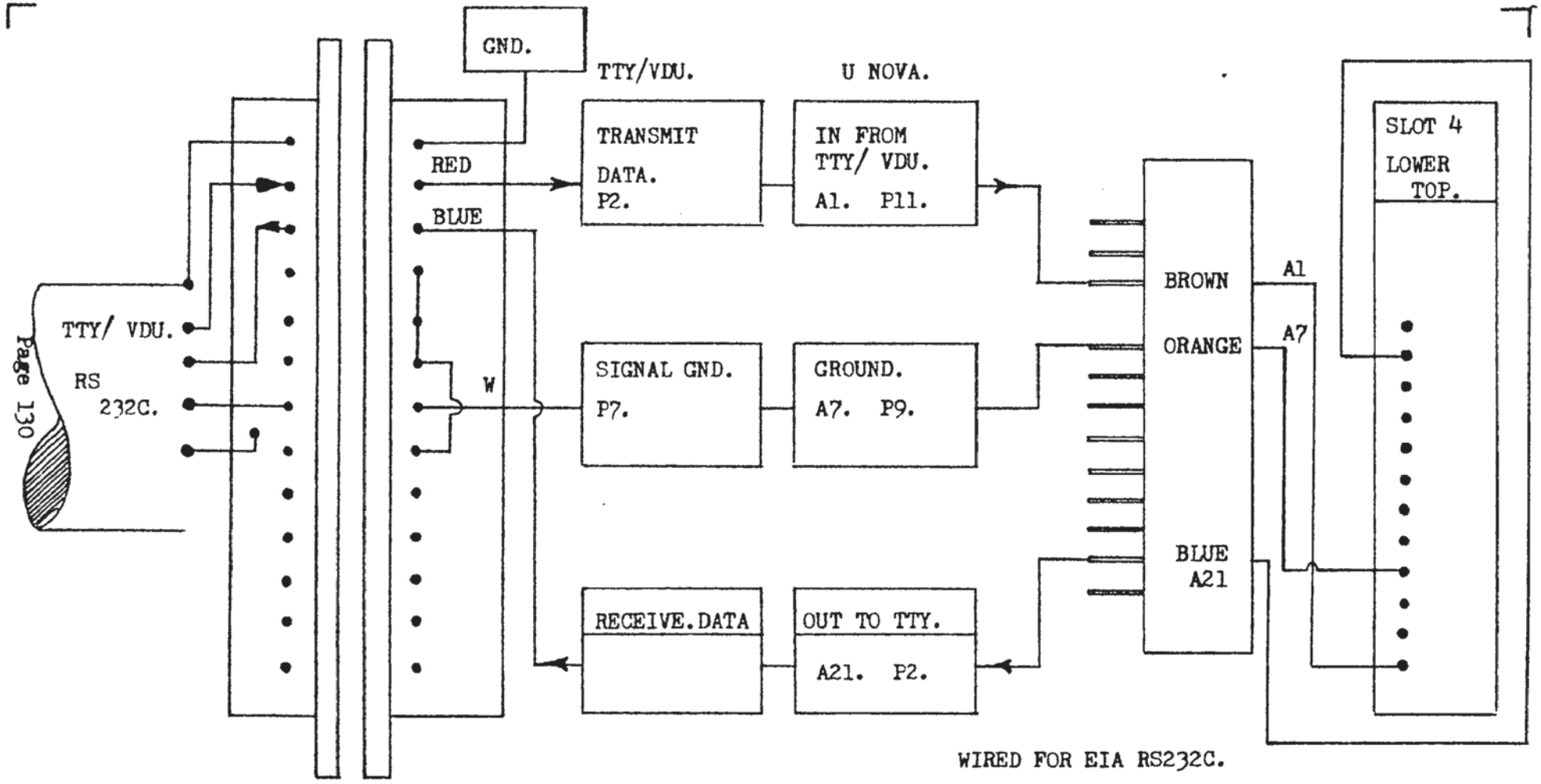


Fig. (7.3.) Z80 TO ASR TELETYPE.



Page 130

Fig. (7.4.) TTY/ VDU TO U NOVA.

7.2. Z80 Microprocessor to the VDU.

A later part of the developement of the proposed system requires that a device is connected for a better graphic display and so it was decided to interface the Z80 microprocessor to an existing VDU , which could either be utilised for monitoring the programme development or for true graphic displays. The VDU had the facility to be set to the RS 232 standard at various baud rates and advantage was taken of the interface already developed for the ASR 33 terminal , this interface was within the Z80 microprocessor system .

Utilising the existing interface, originally designed for the ASR 33 terminal enabled a large reduction in component cost to be made. The resulting interconnecting system with the system wires are detailed in figure 7.5.

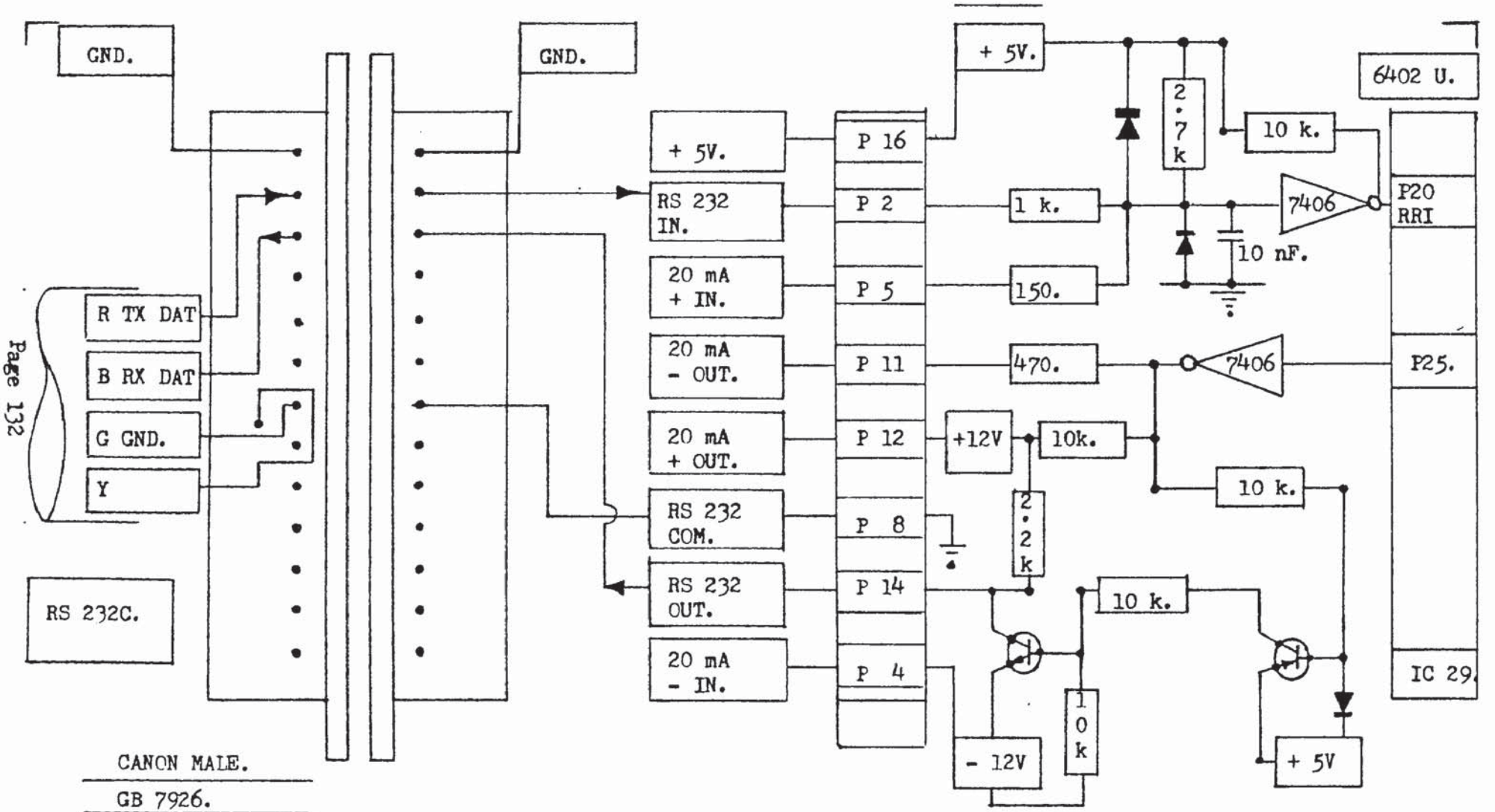


Fig. (7.5.)

Z80 to VDU.

7.3. Z80 to a Domestic Television Set.

Due to the fact that the developed Z80 system had a memory mapped plane display capability, it was decided to interface the unit to a domestic UHF television set. This gave the system several important advantages, as detailed below:-

a) The VDU could be released for more important tasks, when the system only required simple monitoring facilities.

b) The cost of the UHF television was only in the region of £70, as compared to about £700 for another VDU.

c) The monitor system could utilise the memory plane to display the contents of the accumulators or registers, when debugging a programme.

d) A video monitor could be used later, by using the standard 1 volt video signal available from the system.

Figure 7.6 gives electronic details of the Video Modulator.

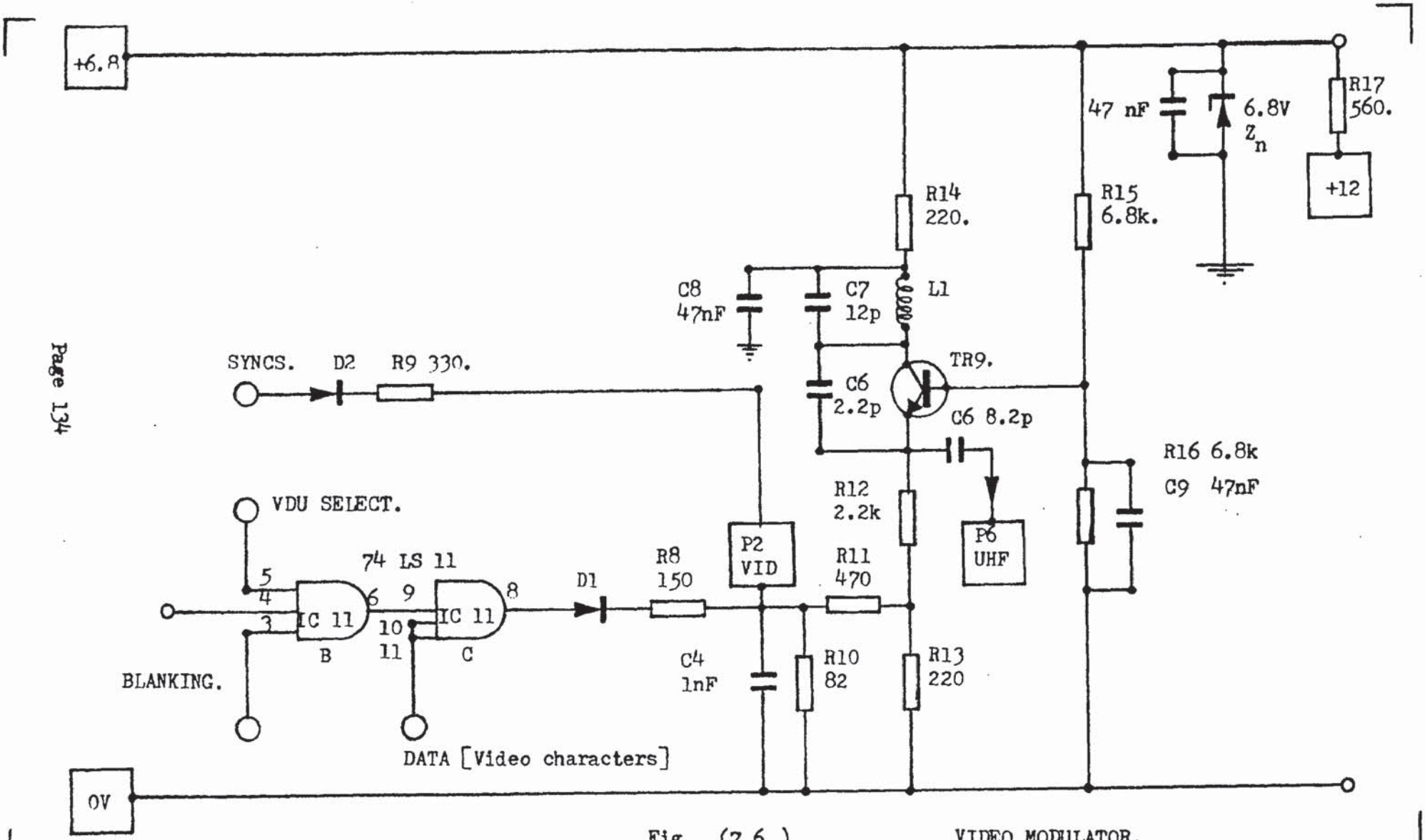


Fig. (7.6.) VIDEO MODULATOR.

7.3.1. The Video Modulator.

Figure 7.6 showed details of the video and UHF modulation system. The clock frequency of 16 MHz was divided down to provide signals for the correct cycling of the memory address lines and to control the video shifting . The resulting waveforms which generate the video blanking , frame synchronising and line frequency pulses are all combined by the two logical AND gates , 74 LS 11 .

Compatibility with a domestic television set is further assured by utilising the UHF oscillator built around the transistor . The 82 Ohm resistor is utilised to develop a composite video signal at the emitter of the transistor and the signal lies within the frequency range of any UHF domestic television set, which may be used without modification.

7.4. Cassette and PIO Interfaces.

Albeit possible to provide internal memories within any microprocessor system , it is always good practice to keep programme copies in some form external to the machine. The cassette recorder gives an excellent cheap , yet efficient way of storing these programmes , which can be reloaded into the internal memory when they are required. Cassette storage also gives the advantage of reutilising the internal memory for many programmes , without the cost associated with an EPROM only system . It also gives the flexibility for an EPROM to be dedicated to regularly used programmes such as the system monitor .

A communicating device exists that enables a domestic cassette recorder to be utilised for this task without modification . The device is called the UART or universal assynchronous receiver and transmitter and figure 7.7 shows the physical layout of a typical IM 6402 UART (7).

IC 29 UART IM 6402.

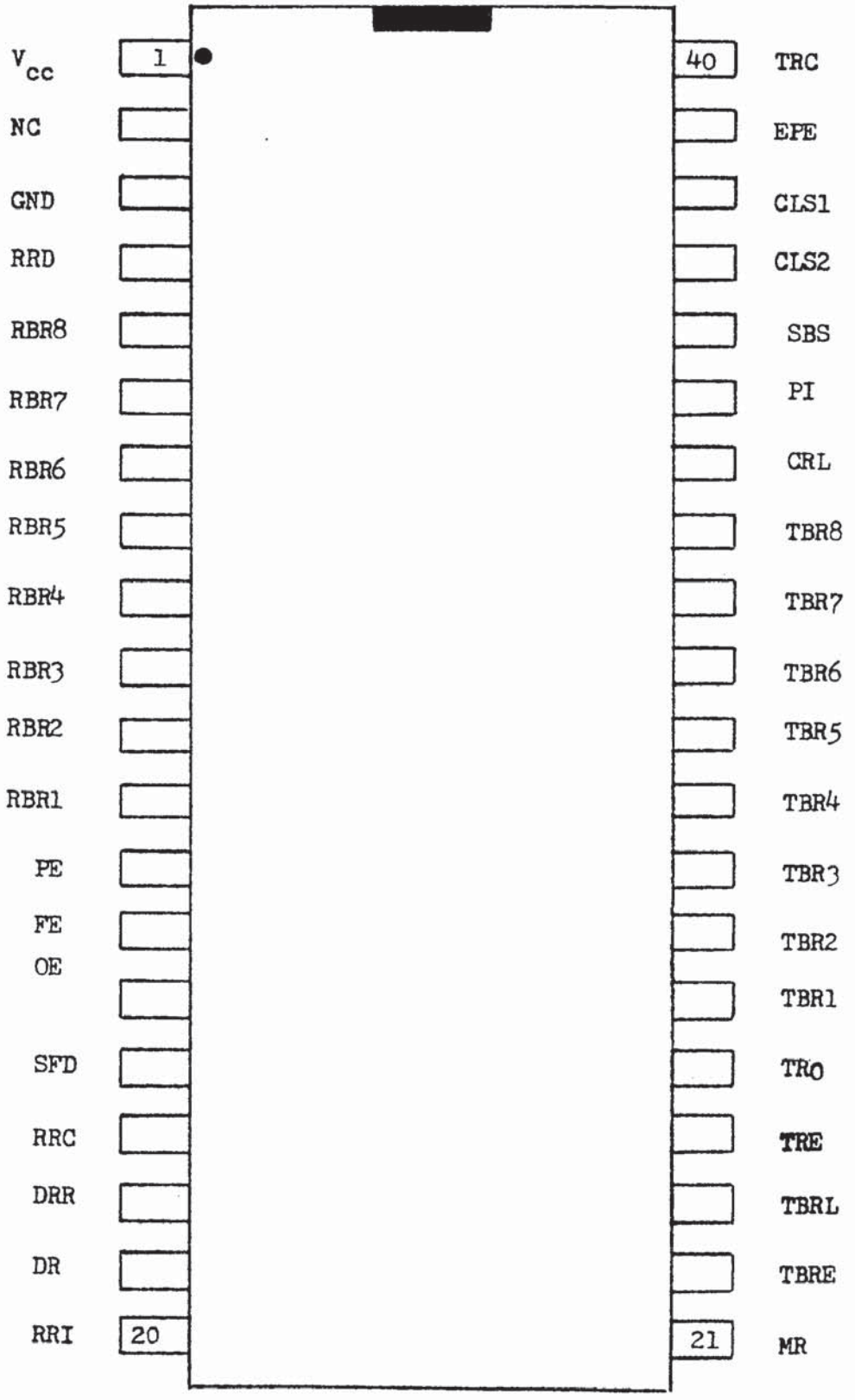


Fig. (7.7.)

7.4.1. UART Interface.

Figure 7.8. : shows the internal interface required to operate the UART, so that it may communicate with the external devices. The serial data is modified by the associated gates to provide several important standard output signals.

a) Cassette Output.

The output is available to drive any domestic tape recorder or cassette unit and consists of groups of serial data, compatible with industrial recording practices.

b) RS 232 Output.

This controlled output voltage is available to feed a printer, or a VDU, or a teletype and has been set up to operate on the international " RS 232 " standard.

c) 20 mA Output.

Flexibility is provided by a second alternative to the " RS 232 " standard, this is known as the " 20 mA current loop", which can drive any system configured to operate on this loop standard.

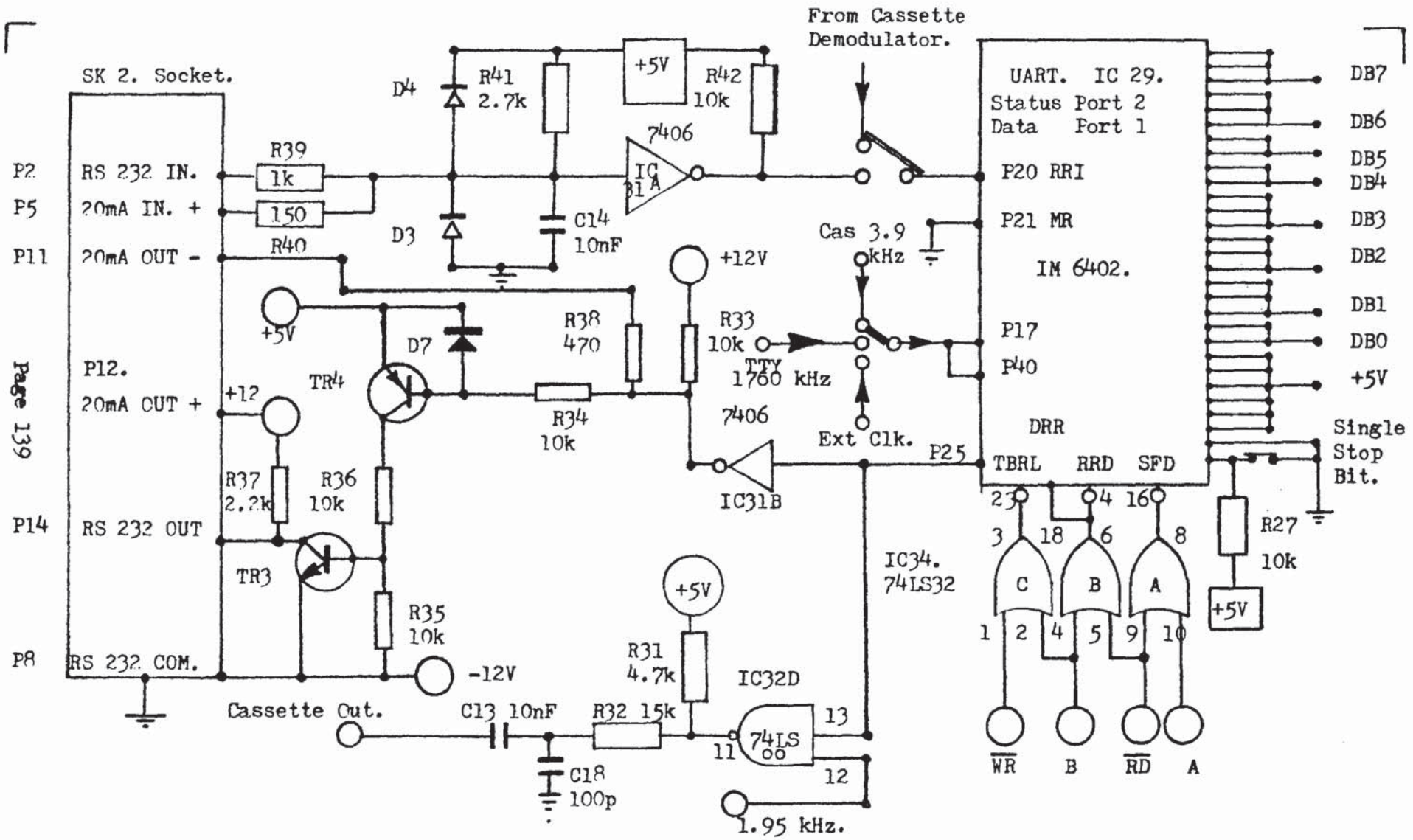
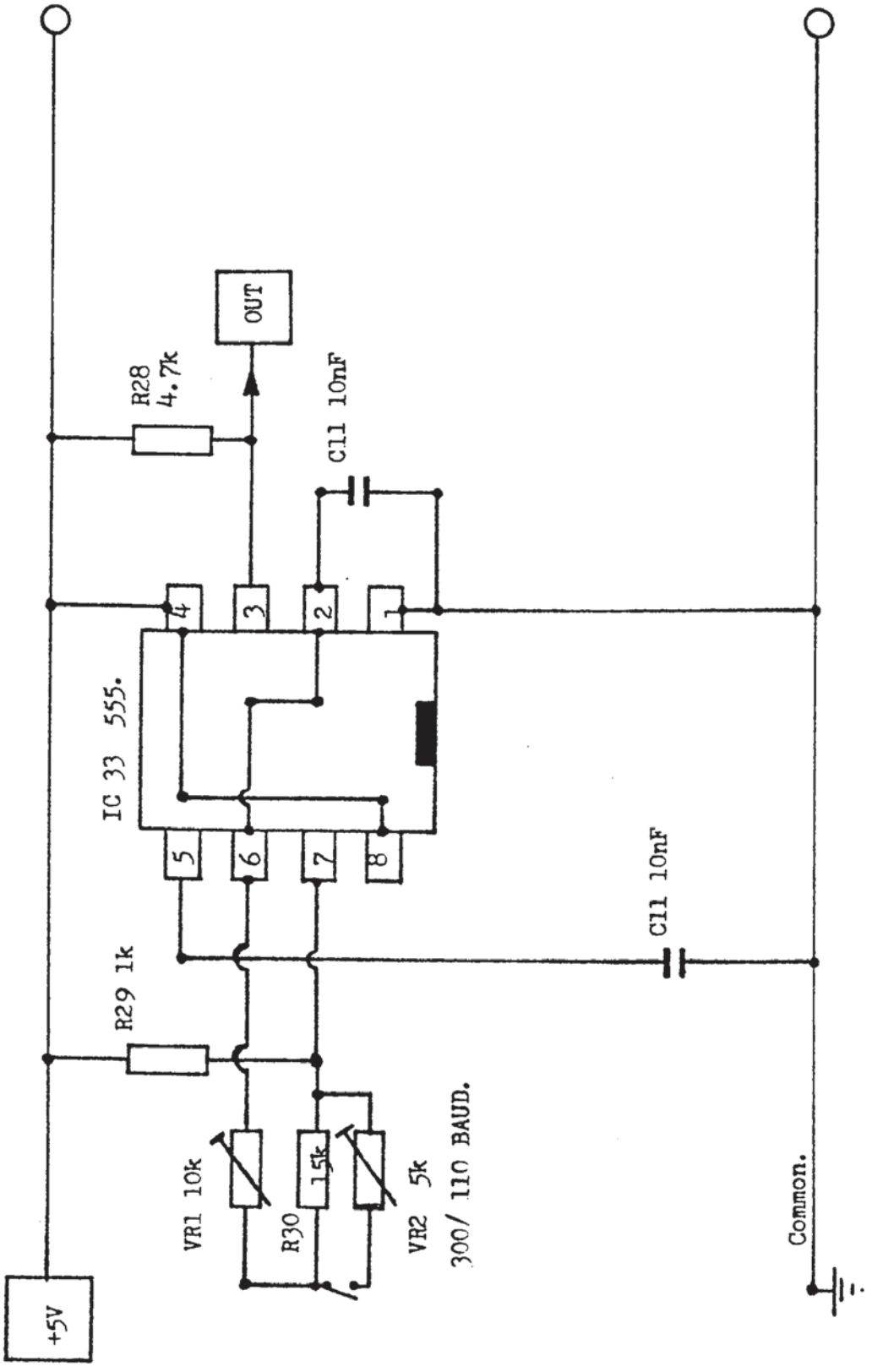


Fig. (7.8.) UART TO SERIAL PORT.

d) A third oscillator is provided internally and consists of a 555 type of integrated circuit chip (3) with associated circuitry which may be adjusted in frequency by utilising a variable resistor. To comply with the data format of teletypes operating at their standard rate of 110 bits per second, it is necessary to set the oscillator frequency to 1760 KHz and use two stop bits after the data stream.

Further improvement is possible as shown in figure 7.9 , this enables a faster rate of 300 Baud to be obtained . Modern equipment are generally adjustable and their next fastest rate above 110 Baud is 300 Baud. The required modification consists of inserting a change over switch and an extra resistor allowing the new Baud rate to be accurately tuned. This rate must be set accurately to match the speed of the communicating device.


OUT
R28
4.7k
C11 10nF
IC 33 555.
C11 10nF
R29 1k
VR1 10k
R30
15k
VR2 5k
300/ 110 BAUD.
+5V
Common.

7.4.2. UART Timing.

The rate at which data is shifted is determined by applying a correctly generated clock pulse, to the receiver and transmitter inputs. The source of the clock can be provided by one of three types of generators:-

a) The main microprocessor system has a 16 MHz crystal oscillator with dividers to bring the fundamental frequency down to 3.90625 KHz , so that it may operate the UART at 244.14 bits per second. The UART always divides the clock frequency by 16, to obtain the bit rate.

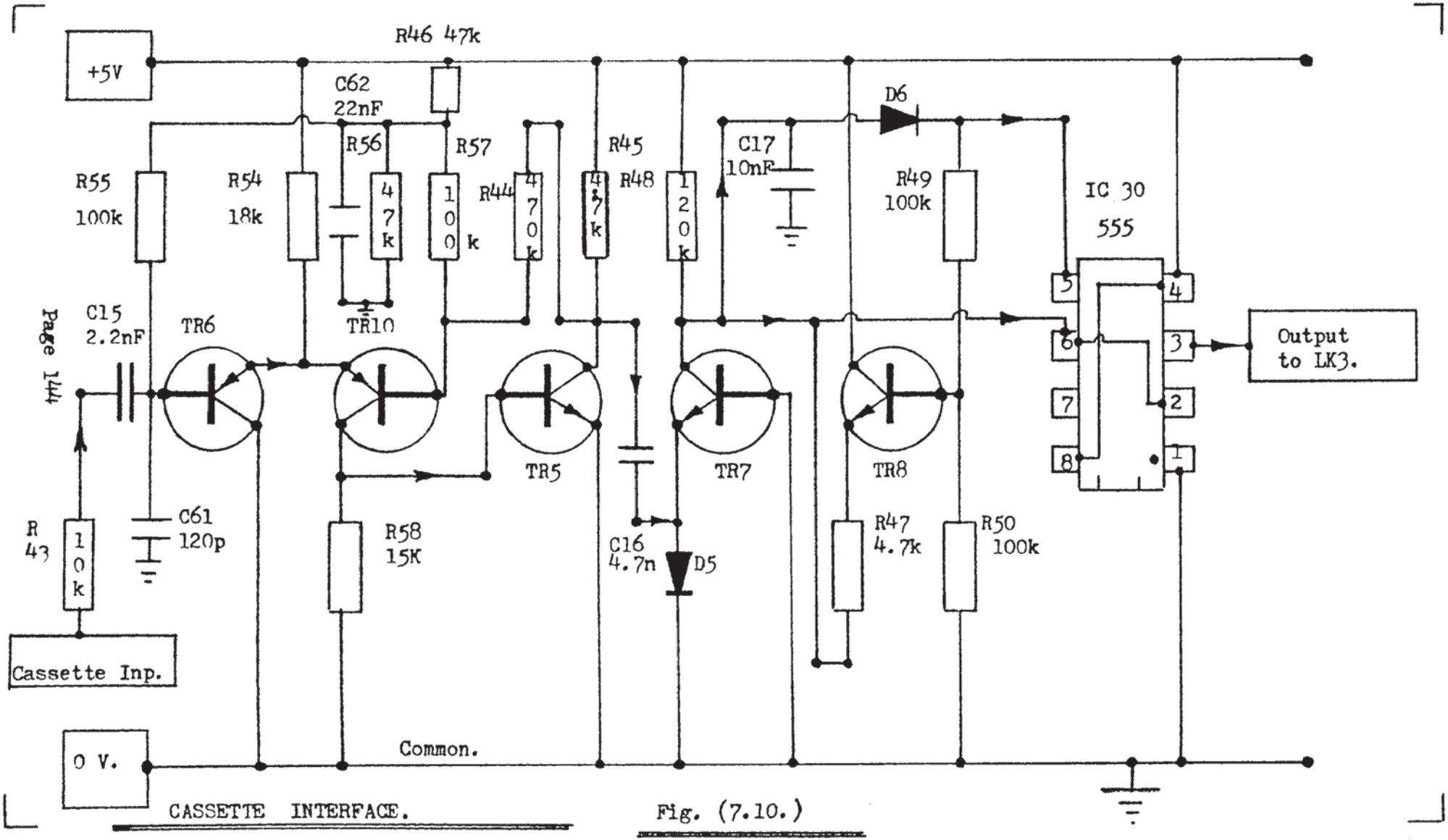
A stop and start bit is added to each byte resulting in 10 bits, for every transmitted word. If pin 36 of the UART is connected to the +5 volt rail, then an extra stop bit is added by the UART, this is required by some systems.

b) An external oscillator may be applied to pins 17 and 40 of the UART, this external clock would then control the transfer rate.

7.5. Cassette Demodulator.

It is necessary to provide a tone detector circuit when receiving data from a cassette recorder because the playback signals consist of a series of tone bursts corresponding to the incoming data stream. Figure 7.10 shows such a tone detector which receives the recorder tone burst by way of two emitter coupled transistors.

The circuit output is first fed into a dc coupled amplifier and the bursts are processed by the 555 type integrated circuit (3) and the associated components, to recover the conventional logic levels from the tone signal. The serial output is then fed into the UART at pin 20 for further processing and converting from the serial form to parallel data suitable for the CPU bus and control system.



CASSETTE INTERFACE.

Fig. (7.10.)

7.6. The PIO.

It is not possible generally to connect the microprocessor CPU directly to the outside world or electronic peripherals for several reasons :-

a) The incompatibility of voltage and logic levels.
b) The incompatibility of digital and analogue signals.
c) Power , driving , sinking and sourcing requirements.
d) The lack of buffering, resulting in a catastrophic CPU and thus system failure should a fault occur or a design error be produced in the external device.

All these problems may be overcome by fitting family devices such as the input output parallel controller known as the PIO . The required conversions to signals and levels generally occur within the family chip. Figure 7.11 shows the physical layout of the PIO integrated circuit chip. The chip is numbered MK 3881 and is manufactured mainly by the Zilog and Mostek companies (8).

IC 35 MK 3881 PIO CHIP.

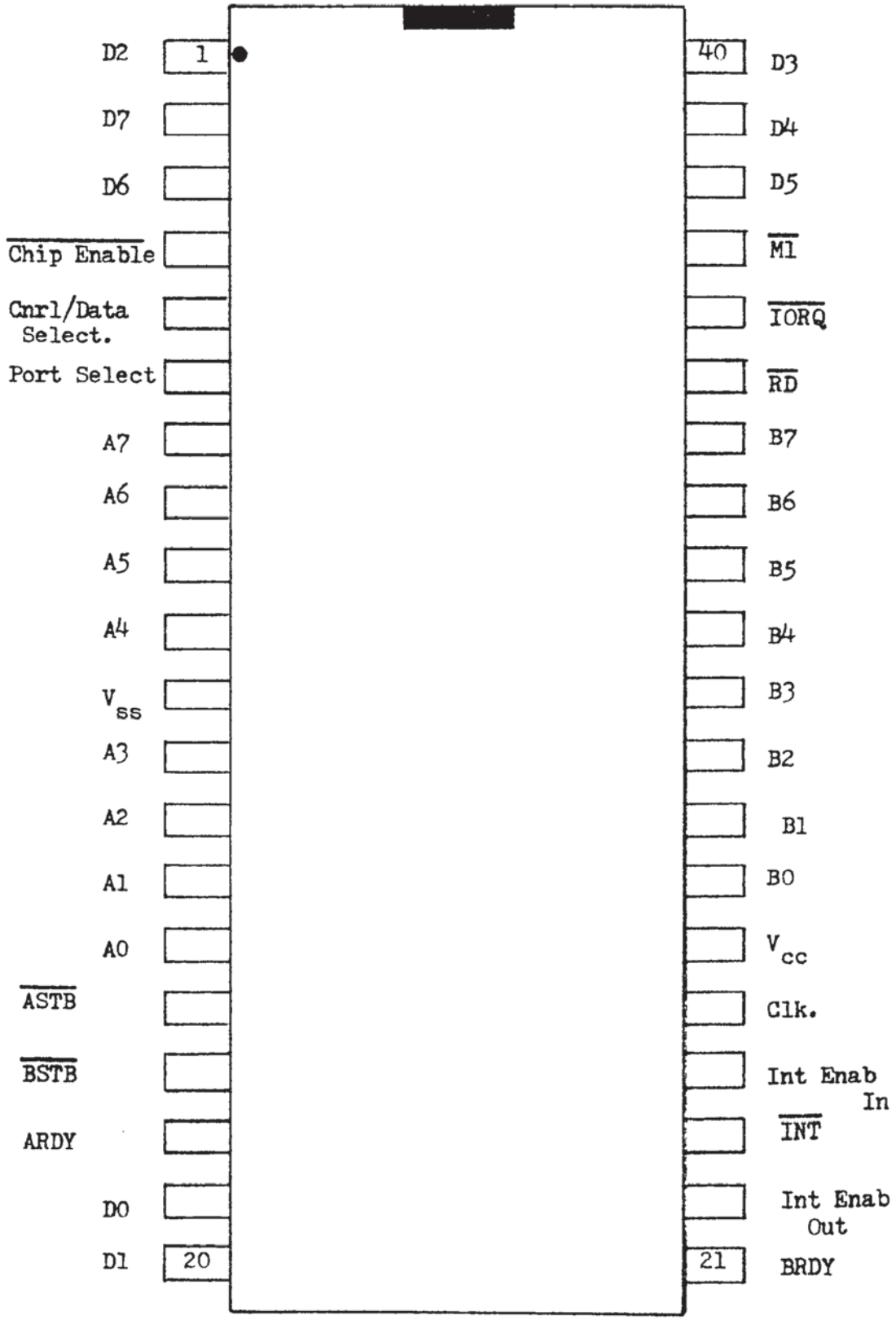


Fig. (7.11.)

7.6.1. The PIO Output Ports.

The MK 3881 controller chip is an LSI package from the Z80 microprocessor family set [8] and requires connection to the CPU for control signals to and from the outside world , via two sockets or ports designated Port A and B. The PIO has the register addresses defined by hardware select logic, but each function is programmable by the controlling CPU. The device interfaces to the CPU and to the user circuits by providing 16 lines , 8 via Port A and the other 8 via Port B, the lines may be defined , by the PIO , to be inputs, outputs or a mixture of both. The PIO can also provide additional handshake signals to enable two way communication with a further external device.

Figure 7.12 shows how the circuit was implemented in this particular design , data is sent from the CPU along data lines DB0 to DB7 inclusive . The input and output signals for both ports enter and leave on the data lines B0 to B7 and A0 to A7 inclusive. Handshake lines are provided by signals $\overline{\text{BSTB}}$ and $\overline{\text{ASTB}}$ with their associated signals $\overline{\text{BRDY}}$ and $\overline{\text{ARDY}}$. The PIO also has interrupt control logic so that it can deviate from the programme execution by a change of external logic state.

The following chapter deals with how the interfaces may be controlled and gives example routines.

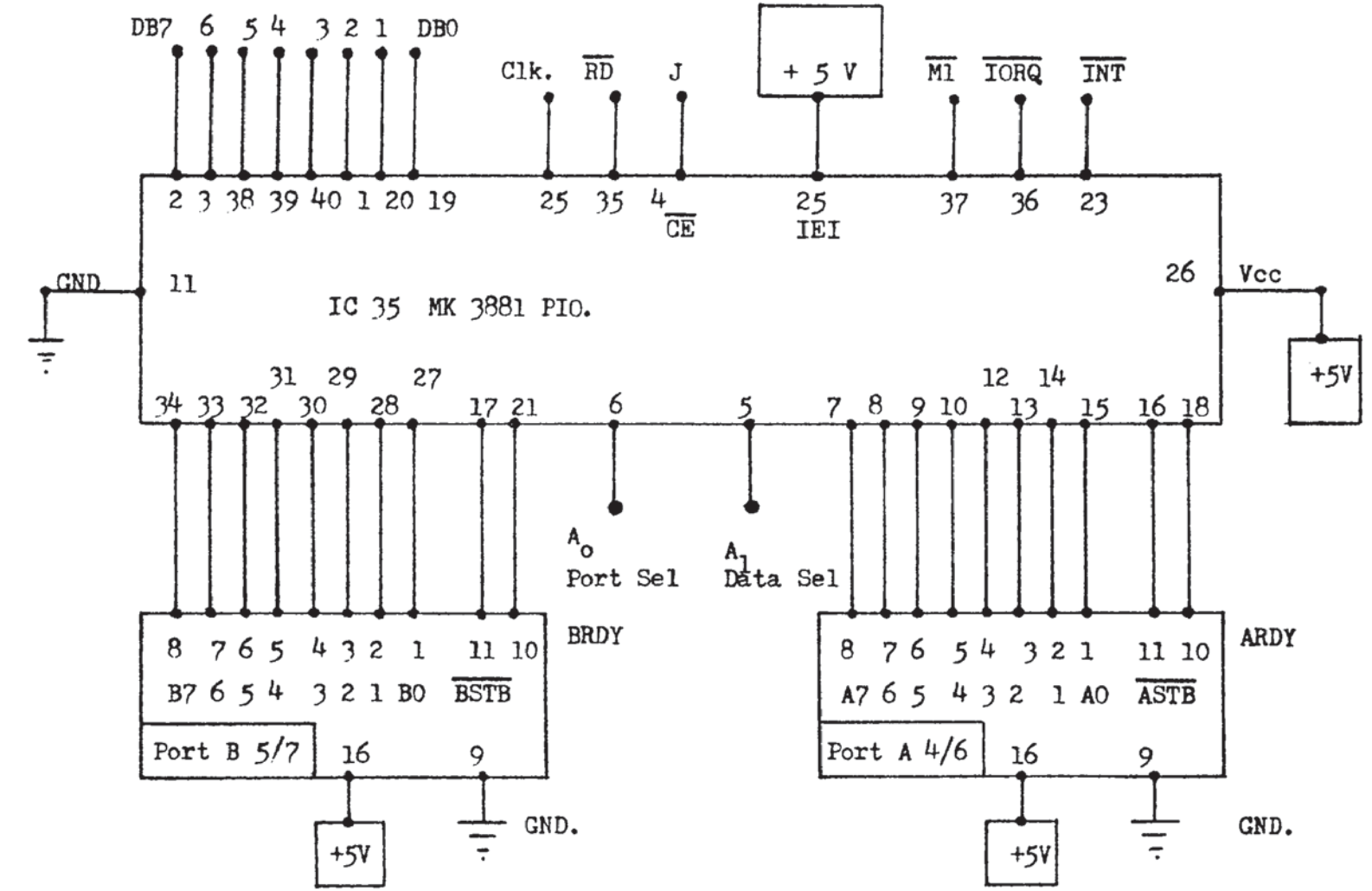


Fig. (7.12.) PIO TO PERIPHERAL PORTS A AND B.

INTERFACE CONTROL ROUTINES.

CHAPTER 8.

CHAPTER 8.

INTERFACE CONTROL ROUTINES.

8.1. Communicating in Machine Code.

There has been a communications problem in history, even back to the time of "Adam and Eve", the inability to communicate resulting in disastrous results. But no matter how complex was early man's problems, they seem minor compared to the barrier confronted today in attempting to communicate with a computer or microprocessor.

Our language consists of words such as "Add or Store" and albeit easily understood by the modern person, they still mean absolutely nothing to a microprocessor; a machine understands only machine language which today is in the form of "Binary Bits", with "0" representing the electrical "OFF" condition and "1" representing the electrical "ON" condition.

A simple instruction such as "Add quantity A to quantity B" would look like "1000 0000" in machine code, it is then a simple matter for the microprocessor to deal with such an instruction, but a very arduous task for a human to follow and understand such numbers, especially when attempting to decode a full page of machine code such as :-

Eg 0110 0111, 1101 1110, 1110 0001, 1100 0010, etc.

Nevertheless when computers were in their infancy, this was exactly the way man communicated with them. Programmes were entered with binary switches, which was very slow and errors became prevalent.

The introduction of the keyboard greatly simplified data entry, machines could then be instructed by easier numbers externally, such as "Octal or Hexadecimal" numbers. An "Octal" number is one expressed to the base of "8", a "Hexadecimal" number is one expressed to the base of "16" and a "Denary" number is one expressed to the base of "10", which is the one we use in every day life.

When a number is written to a different base, the number itself does not change, but only the way in which it is expressed.

For example, the "Denary" number 159 , (One hundred and fifty nine.), can be expressed in various ways as shown below:-

Denary	=	159	Base 10.
Binary	=	10011111	Base 2.
Octal	=	237	Base 8.
Hexadecimal	=	9F	Base 16.

A number is now entered via a keyboard control system that converts the "Hexadecimal" number into the correct "Binary" bit patterns, so that the microprocessor can deal with them. This method ensures that less key closures are required by the operator , enabling faster and more accurate programming to occur.

Albeit the keyboard is a big step forward, it still has drawbacks, because the human mind finds "Hexadecimal" numbers difficult to comprehend, unless the operator is trained. If a programme is to be inserted it must first be converted into the "Hexadecimal" system, this requires the use of a special table or chart called the "Op Code Table".

For example, the previous operation of "Add quantity A to quantity B", had a binary equivalent of "1000 0000", which when converted into "Hexadecimal" gives the code "80", to compound the problem, an opcode table may have several hundred entries to look at, before finding the correct code.

To review the problem:

To write a programme for keyboard entry

it is necessary to :-

a) Write the programme in terms of human understanding.

b) Look up the appropriate " Opcode " in a special table.

c) Rewrite the programme in " Hexadecimal " form.

d) Enter the hexadecimal codes via the keyboard.

e) Decode the hexadecimal codes to the " Binary " machine code language.

There is however a new step that has been inserted, above the machine code hexadecimal language, which is called the "Assembly Language".

This language has two distinct advantages over machine code:-

i) Each instruction is represented by a " Mnemonic " or memory aid.

The mnemonic for the previous " Add quantity A to quantity B " is simply " ADD A , B ".

Each mnemonic can be easily identified with the associated instruction and with regular use become very easy to remember.

ii) Addresses may be symbolic.

This means that a jump or call may be identified by a label such as :-

DEL	(For delay)
PROG	(For programme)

The mnemonics and symbolic addresses are alpha-numeric, a combination of letters and numbers and cannot be entered with the " Octal " or "Hexadecimal " keyboard. This is not in fact a large disadvantage because they can be entered with a standard " QWERTY " typewriter keyboard, making the data access even easier and the office typist could be utilised to enter the programmes without further training.

The use of a standard typewriter also means that the operator need only be able to type, not understand the mnemonics. This fact enables programmers to share the task of writing and the normally laborious, to the programmer, entering of the programme into the computer.

There are two popular alphanumerical input and output devices used today, these are the "Teletypewriter" and the "Video Terminal", or VDU as it is commonly called.

The teletypewriter enables programmes to be typed into the microprocessor and the input and output is then printed on paper, giving the operator a permanent record of the work in hand.

The video terminal or VDU has a television like monitor replacing the paper printer, the data is displayed on the screen and video graphics can be obtained under the microprocessor control, however no hard paper copy is possible.

These codes and their assembly equivalents may be found in appendix 1.

8.1.1. Test Routines and the Screen Display.

The system has been designed to interface into a standard type of domestic UHF television set , by using the television screen as a memory plane peripheral video monitor. This means that not only can the television monitor be used to display the memory contents by using the M command key, or to display the register contents , by using the R command key , but also to act as a simple low resolution 16 by 48 character display area.

There now follows a simple example programme to illustrate the display principle, by indicating how a number or symbol may be displayed on the video monitor screen at a set position. It is also possible to obtain moving displays and further details may be found in the authors book on machine code programme examples (1).

8.1.2. Screen Display.

This programme will display " A5 ? " on the video screen.

Memory.	Opcode.	Mnemonic.	Meaning.
0C80	21 A0 09	LD HL, 09A0	Set the HL register to point to the screen centre.
0C83	3E 41	LD A, 41	Load A with the ASCII code for the letter A.
0C85	77	LD (HL) , A	Display the letter A.
0C86	23	INC HL	Move the screen pointer.
0C87	23	INC HL	Move the screen pointer.
0C88	3E 35	LD A , 35	Load A with the ASCII code for the number 5.
0C8A	77	LD (HL) , A	Display the number 5.
0C8B	23	INC HL	Move the screen pointer.
0C8C	23	INC HL	Move the screen pointer.
0C8D	3E 3F	LD A , 3F	Load A with the ASCII code for the symbol ? .
0C8F	77	LD (HL) , A	Display the symbol ? .
0C90	DF 5B	MRET	Return to the system monitor.

The previous programme may be executed by following the sequence below, note that "Spacebar" is the large long bar on the bottom of the keyboard and that "Newline" is a special key on the right of the keyboard. "Reset" is the hidden key on the right.

Press RESET (This will clear the screen)

Type E Spacebar 0C80 Newline

The programme will now execute starting from memory location 0C80 and will display :-

A 5 ?

The display should be near the screen centre, the coded instruction "DF 5B" will allow the system to automatically return correctly to the monitor system, which will wait for further commands.

The message may be moved in screen position by changing the address "09A0" at location 0C81 and 0C82 , try 09B0, note that the address is entered in reverse order as B0 09.

8.2. Interface Test Programme.

The system has been designed to operate two banks of 8 control lines via 74LS86 TTL address gates (9) (10), the gates are connected to address lines A_2 to A_7 inclusive . Address lines A_0 and A_1 are also used , A_0 being utilised to gate in either of the two banks of 74LS175 latches, a bank of latches can only be selected if all of the inputs to the 74LS10 gates are at high logic , ie at the + 5 volt level. The gating signals are completed by utilising the WR and IORQ signals from the CPU , a bank is considered by the system to be an output port.

A test board was built and used to confirm the operation of the initial tests on the lathe , in binary , by showing the logic levels on two banks of red and green leds connected to the data lines (3). Figure 8.1 shows the test board layout and there then follows example programmes showing how the signals are obtained to operate the appropriate output line.

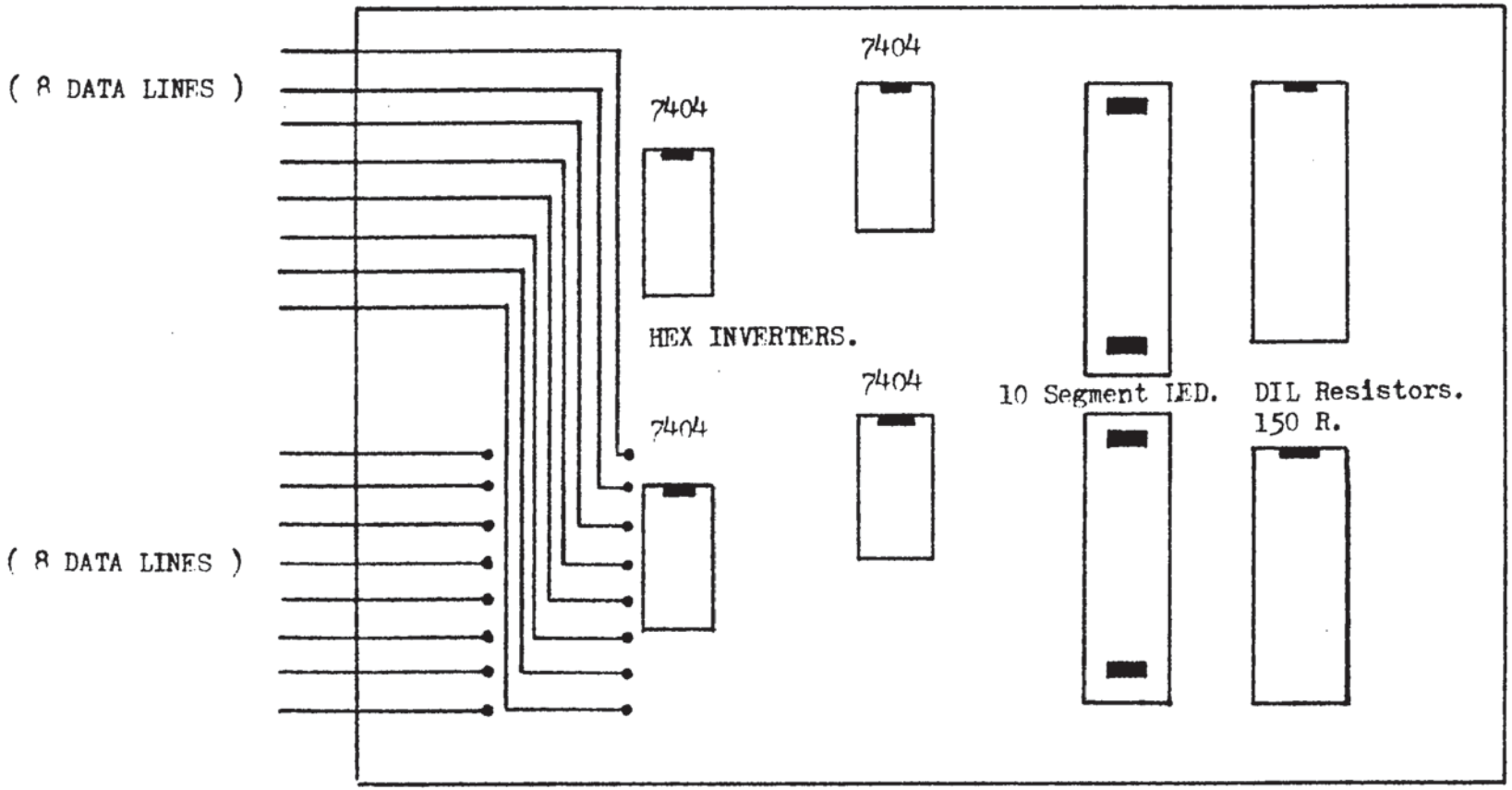


Fig. (8.1.) LED TEST BOARD LAYOUT.

8.2.1. Green LED Bank Test Programme.

Memory.	Opcode.	Mnemonic.	Meaning.
0C80	0E 02	LD C , 02	Load register C with 02, the port number of the "Green" bank of leds.
0C82	3E 81	LD A , 81	Load accumulator A with a signal to light the two end bars of the "Green" leds.
0C84	ED 79	OUT [C] , A	Send out the signal pattern in accumulator A, to the port number contained in register C. ie the "Green".
0C86	DF 5B	MRET	Return control to the monitor system.

8.2.2. Red and Green LED Bank Test Programme.

Memory.	Opcode.	Mnemonic.	Meaning.
0C80	0E 02	LD C,02	Load register C with 02, the port number of the "Green" bank.
0C82	3E FF	LD A,FF	Load accumulator A with a signal to light all the green LED.
0C84	ED 79	OUT (C),A	Send the signal to the port number contained in register C. [Green]
0C86	0C	INC C	Increment C from 2 to 3, the port number of the "Red" bank.
0C87	3E 81	LD A,81	Load accumulator A with a signal to light the two end LED of the red bank of LED.
0C89	ED 79	OUT (C),A	Send out the signal to the port number contained in register C.[Red]
0C8B	DF 5B	MRET	Return control to the monitor system.

8.2.3. A Scanning Test Programme.

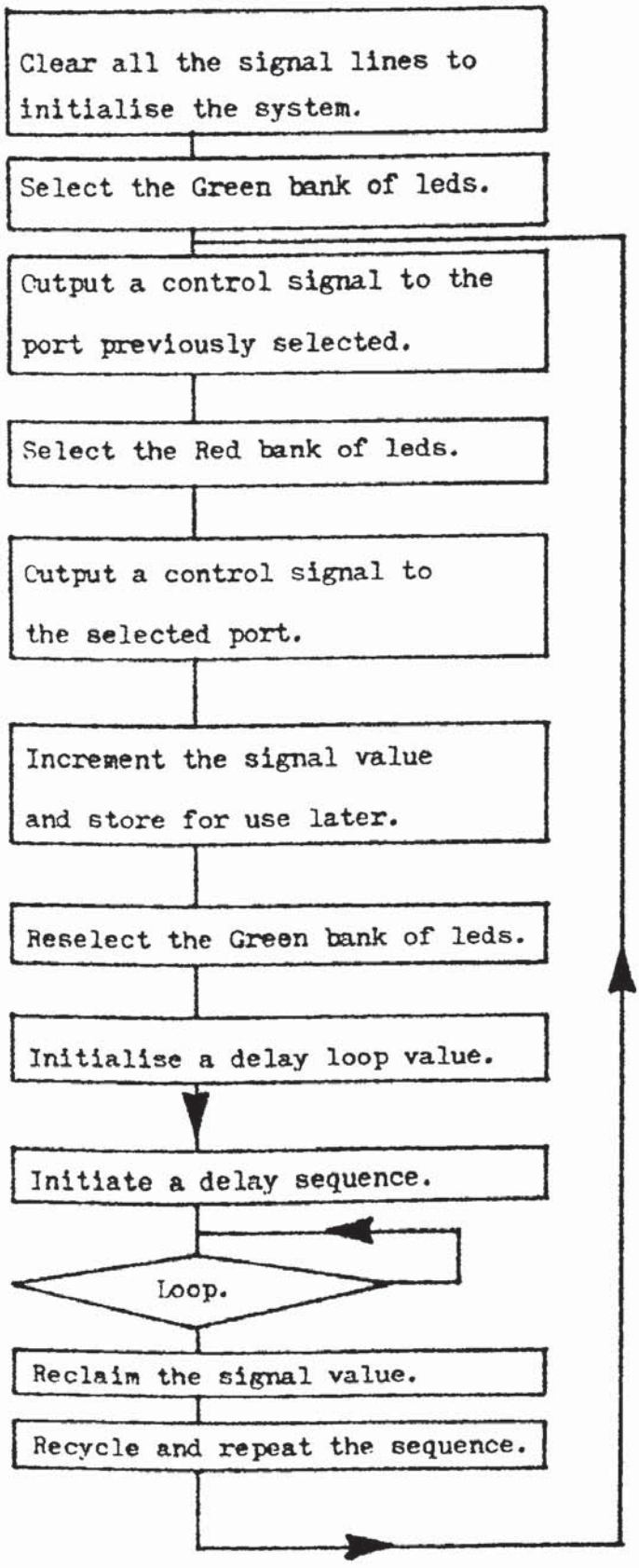
The routine will scan the output lines and light the "Red" and "Green" leds, one at a time. This is a very useful routine which may be utilised to test any further device connected to any of the output lines.

Memory.	Opcode.	Mnemonic.	Meaning.
0C80	3E 00	LD A , 00	Load accumulator A with a signal to clear all the lines.
0C82	0E 02	LD C , 02	Load C with the port number of the "Green" bank of leds, ie 2.
0C84	ED 79	OUT [C] , A	Send the signal out to the port number in C.
0C86	0C	INC C	Increment the number 2 in C, to 3, which is the port number of the "Red" bank of leds.
0C87	ED 79	OUT [C] , A	Send the signal out from A to the port number in C.

A Scanning Test Programme. [Continued]

Memory.	Opcode.	Mnemonic.	Meaning.
0C89	3C	INC A	Increment the signal value in accumulator A.
0C8A	0D	DEC C	Change the port number [3] in register C back to 2, to operate the "Green" bank.
0C8B	08	EX AF',AF	Copy the signal value.
0C8C	06 50	LD B, 50	Load register B with a loop value of 50 hexadecimal, ie 80.
0C8E	3E FF	LD A , FF	Load accumulator A with FF.
0C90	FF	RST 38H	Wait for a period dependent on the value in accumulator A.
0C91	10 FB	DJNZ	Reduce the loop value in B, if not zero then jump back to 0C8E.
0C93	08	EX AF',AF	Reclaim the signal value stored at 0C8B.
0C94	C3 84 0C	JP , 0C84	Jump back to location 0C84, to continue.

SCANNING TEST PROGRAMME



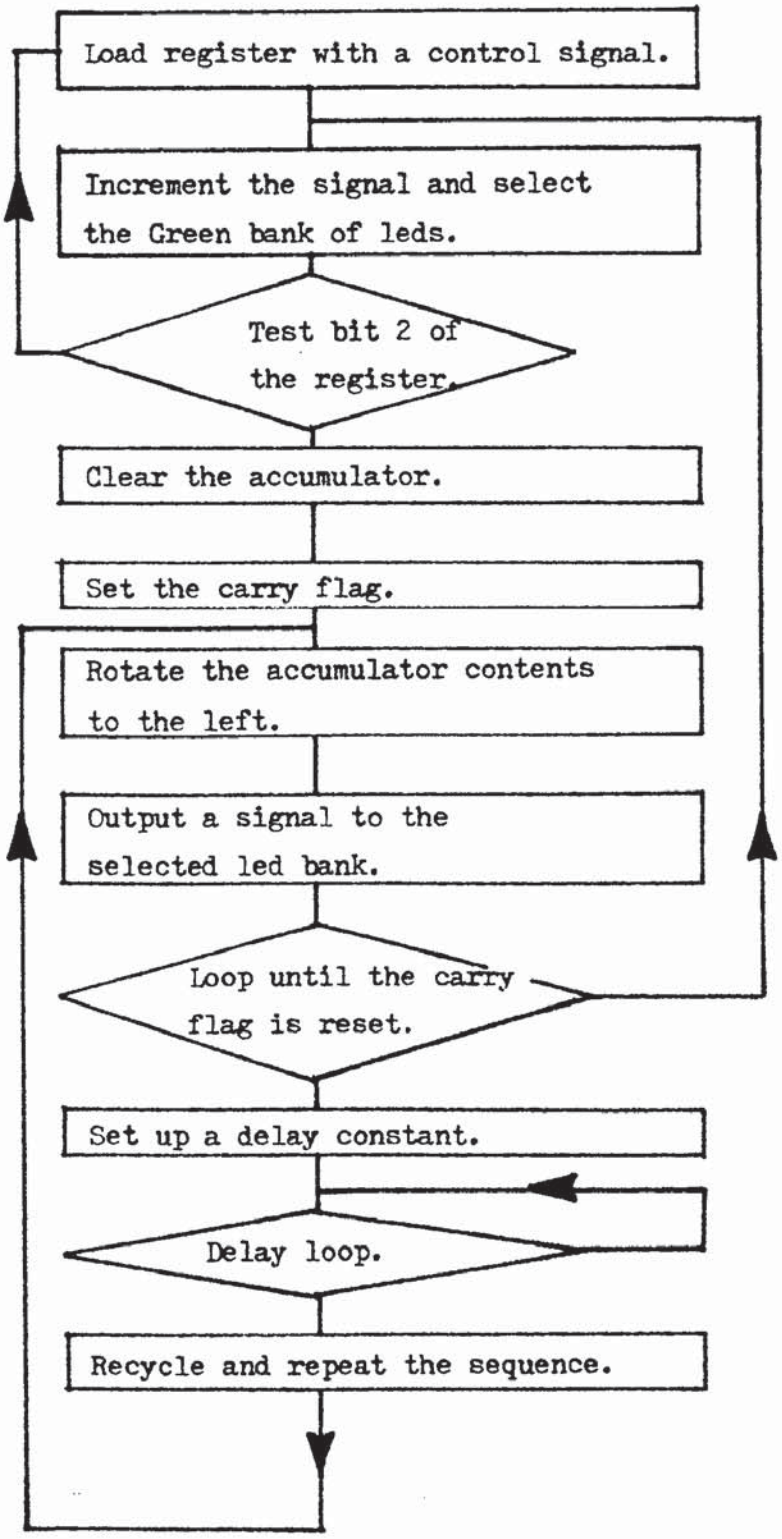
8.2.4. Alternative Red and Green Bank Test.

Memory.	Opcode.	Mnemonic.	Meaning.
0C80	0E 01	LDC , 01	Load register C with 01.
0C82	0C	INC C	Increment the value in register C to 02, the port number of the "Green" bank of leds.
0C83	CB 51	BIT 2,C	Test bit 2 of register C.
0C85	20 F9	JR NZ	If bit 2 of register C was zero jump back to 0C80.
0C87	3E 00	LDA , 00	Clear accumulator A.
0C89	37 17	RLA SCF	Rotate the contents of accumulator A to the left, through the carry flag, after setting the carry flag to 1.
0C8B	ED 79	OUT (C), A	Send the signal out to the port number contained in register C.
0C8D	38 F3	JR C	When the value in A has been completely rotated so that the 1 in the carry is back in the carry flag, then jump back to location 0C82.

Alternative Bank Test Continued.

Memory.	Opcode.	Mnemonic.	Meaning.
0C8F	06 00	LDB , 00	Load register B with a delay variable of 256.
0C91	C5	PUSH BC	Save this value on the system stack.
0C92	10 FE	DJNZ	Reduce the value in register B by 1, if not zero jump back to location 0C92, if B=0 then continue.
0C94	C1	POP BC	Recall the value that was saved, into register B, from the stack.
0C95	10 FA	DJNZ	Reduce this value in register C, if not zero jump back to location 0C91, if zero continue.
0C97	18 F1	JR	Jump back to 0C8A and continue.

RED AND GREEN BANK TESTING



8.3. An Example of Pegboard Control.

An examination of the " Behind the Pegboard " diagram will show that 8 lines have been utilised in the original control system. The table , on the diagram, shows where each line, from the PIO, is connected.

This table must be interpreted correctly before a control sequence can be initiated and therefore an example is now provided, to clarify the table.

Example Sequence.

a) Rotate slowly anticlockwise.

b) Stop the rotation.

c) Rotate slowly clockwise.

d) Stop the rotation.

e) Rotate quickly anticlockwise.

f) Stop the rotation.

g) Rotate quickly clockwise.

Taking each in turn gives :- .....

The two following tables should assist to clarify the example.

Speed Control.

Peg Number.	X	ROTATIONAL SPEED.
--	28	158
10	In	In
13	In	Out

Bit Values of Each Peg.

Peg	14	13	8	2	1	7	10	15
Bit	7	6	5	4	3	2	1	0
Value	80	40	20	10	08	04	02	01
Action	Anti	Rot	Tool Out	Tur In	Tur In	Tool In	Rot	Clock

Sequence a.

Anticlockwise rotation requires Peg 14 to be activated.

A slow rotation of 28 revolutions requires that Pegs 10 and Pegs 13 be activated.

ie: Peg 14 = Bit 7 = 80

Peg 10 = Bit 1 = 02

Peg 13 = Bit 6 = 40

Total Value = C2 (Hexadecimal)

Hence it would require that bit value " C2 " in hexadecimal form be sent out via the PIO output port.

Sequence b.

To stop: the rotation pegs are required in, for speed and out, for direction.

ie: Peg 10 = Bit 1 = 02

Peg 13 = Bit 6 = 40

Total Value = 42 (Hexadecimal)

Sequence c.

Clockwise.

Peg 15 = Bit 0 = 01

Peg 10 = Bit 1 = 02

Peg 13 = Bit 6 = 40

Total Value = 43

Sequence d.

Peg 10 = Bit 1 = 02

Peg 13 = Bit 6 = 40

Total Value = 42

Sequence e.

To rotate quickly at 158 revolutions : requires Peg 10 to be in and Peg 13 to be out, with Peg 14 giving anticlockwise rotation.

ie: Peg 14 = Bit 7 = 80

Peg 10 = Bit 1 = 02

Total Value = 82

Sequence f.

To stop the rotation requires the speed peg, but not the rotation peg. ie:- Peg 10 = Bit 1 = 02.

Total value = 02 Hexadecimal.

Sequence g.

Clockwise. Peg 15 = Bit 0 = 01.

Peg 10 = Bit 1 = 02.

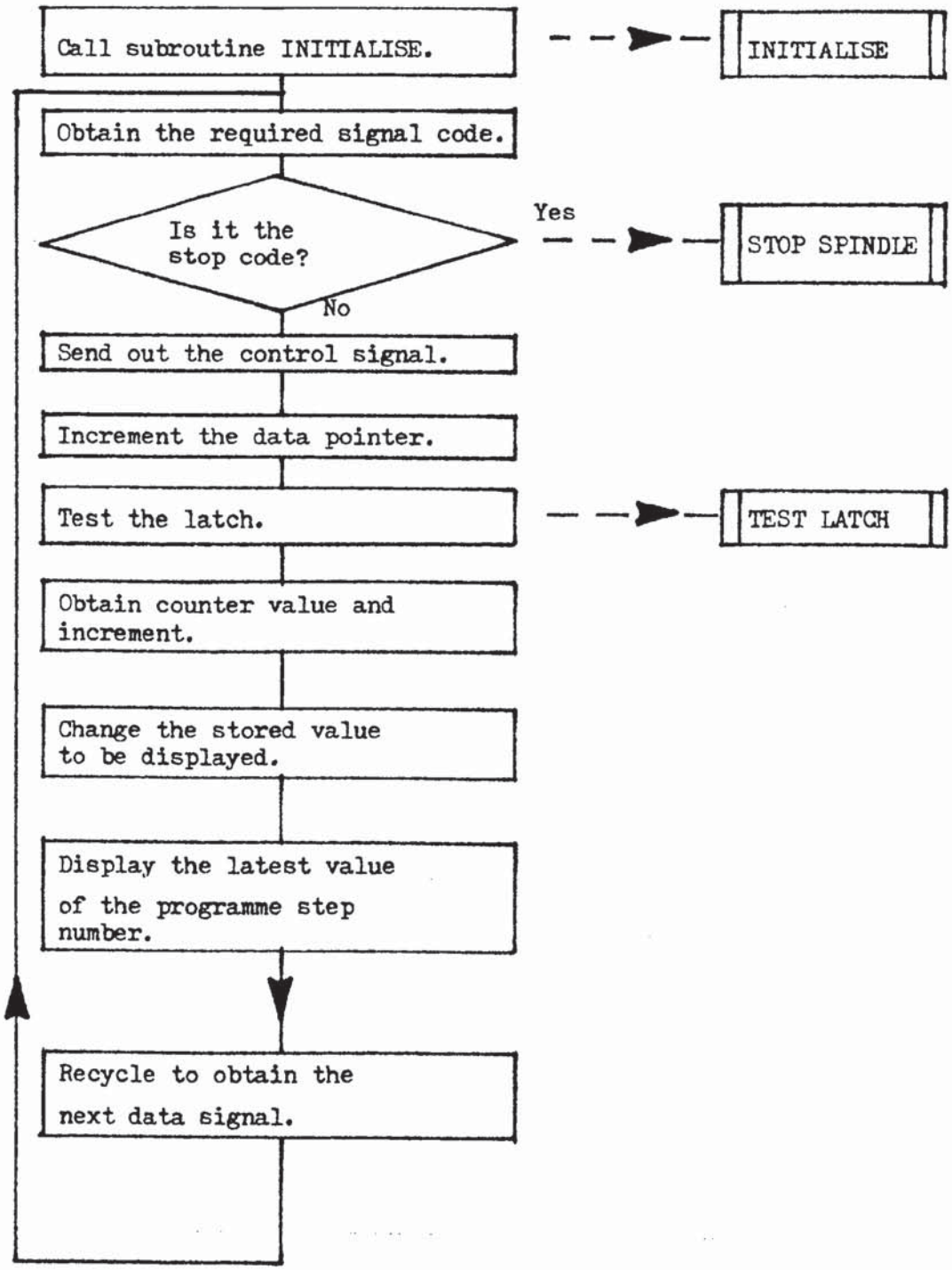
Total value = 03. Hexadecimal.

The above sequence can be carried out by sending out each hexadecimal code in turn, to the output port of the PIO. It is more convenient to save all the required codes for the complete programme in a table:-

ie. C2 . 42 . 43 . 42 . 82 . 02 . 03 . (Hexadecimal).

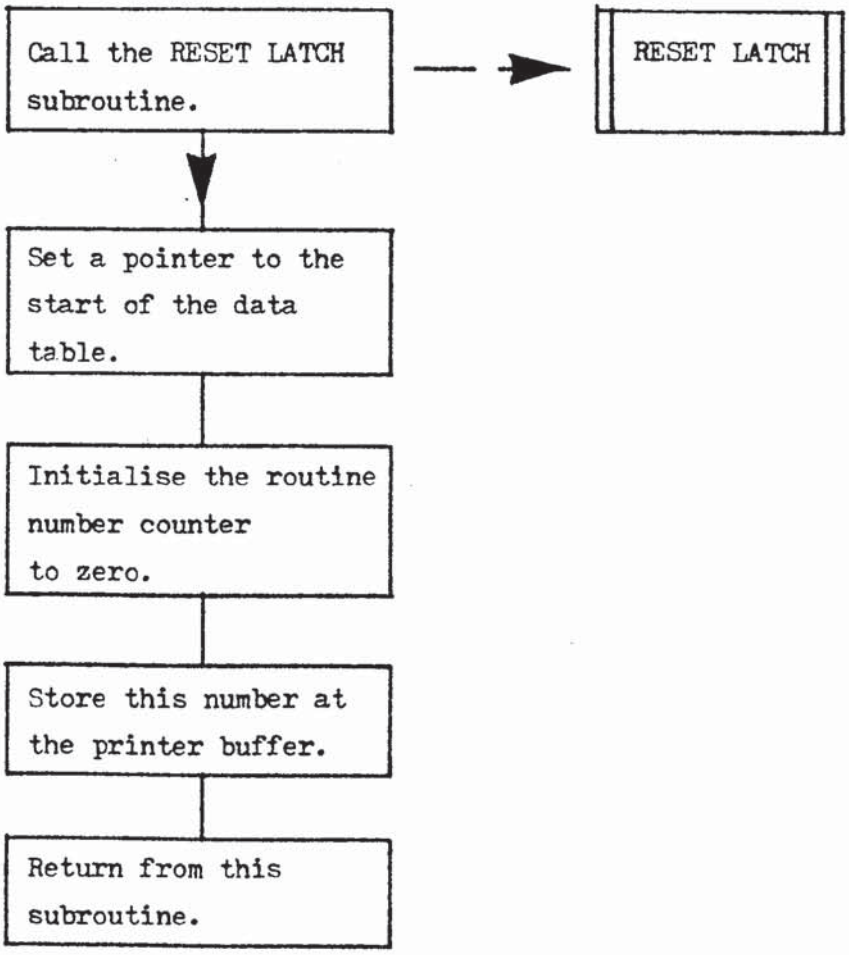
A monitor programme is then required to send the codes out from the table at the correct time, the programme would require handshaking, with the lathe, so that the pegs could be activated in step with the lathe operation.

8.3.1. WARD Lathe Control.

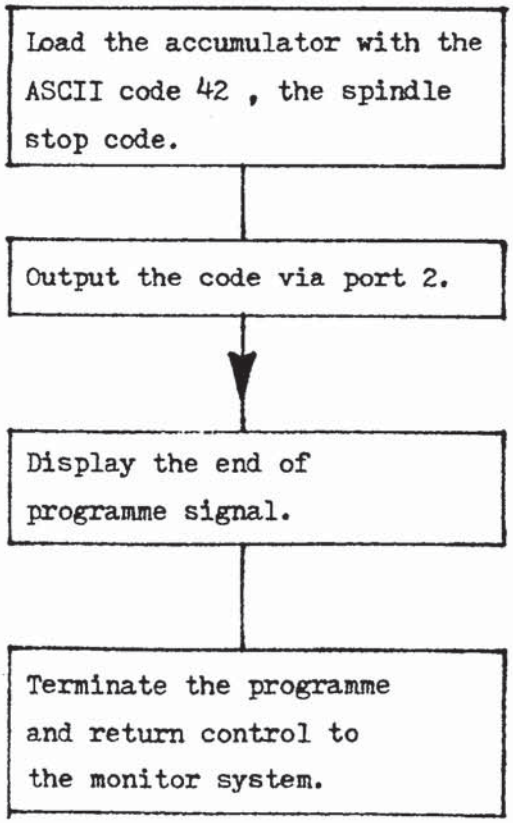


8.3.2. Pegboard Control Flowcharts.

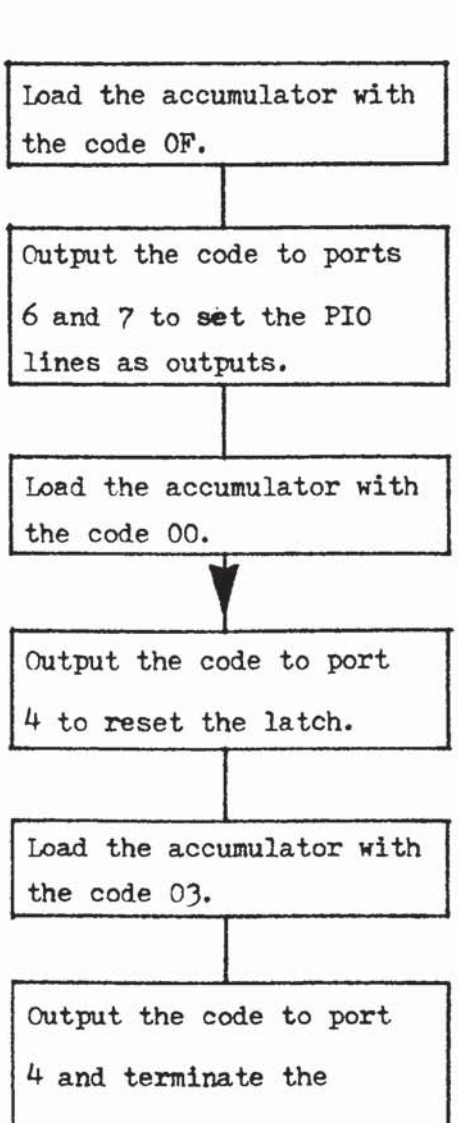
I N I T I A L I S E



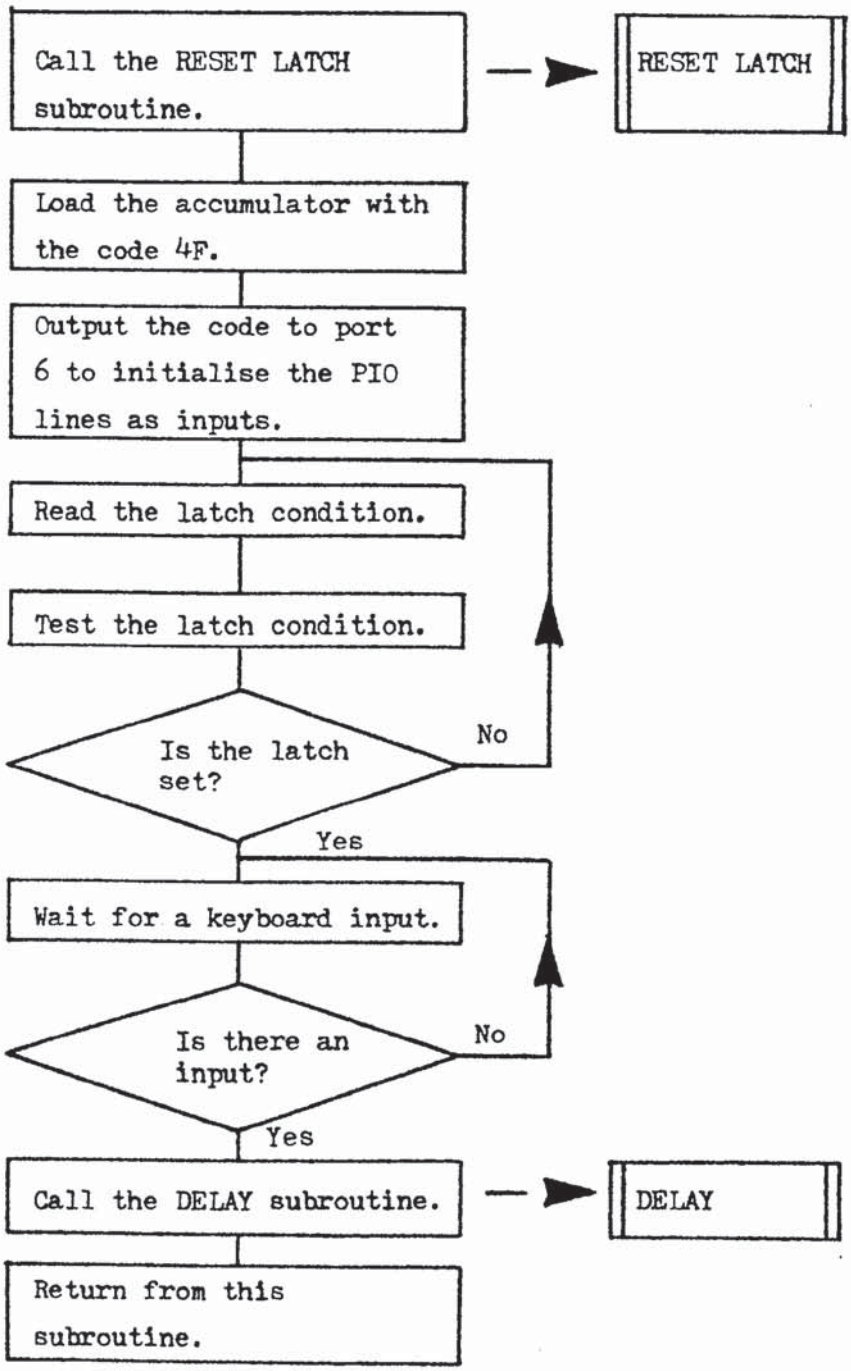
S T O P S P I N D L E



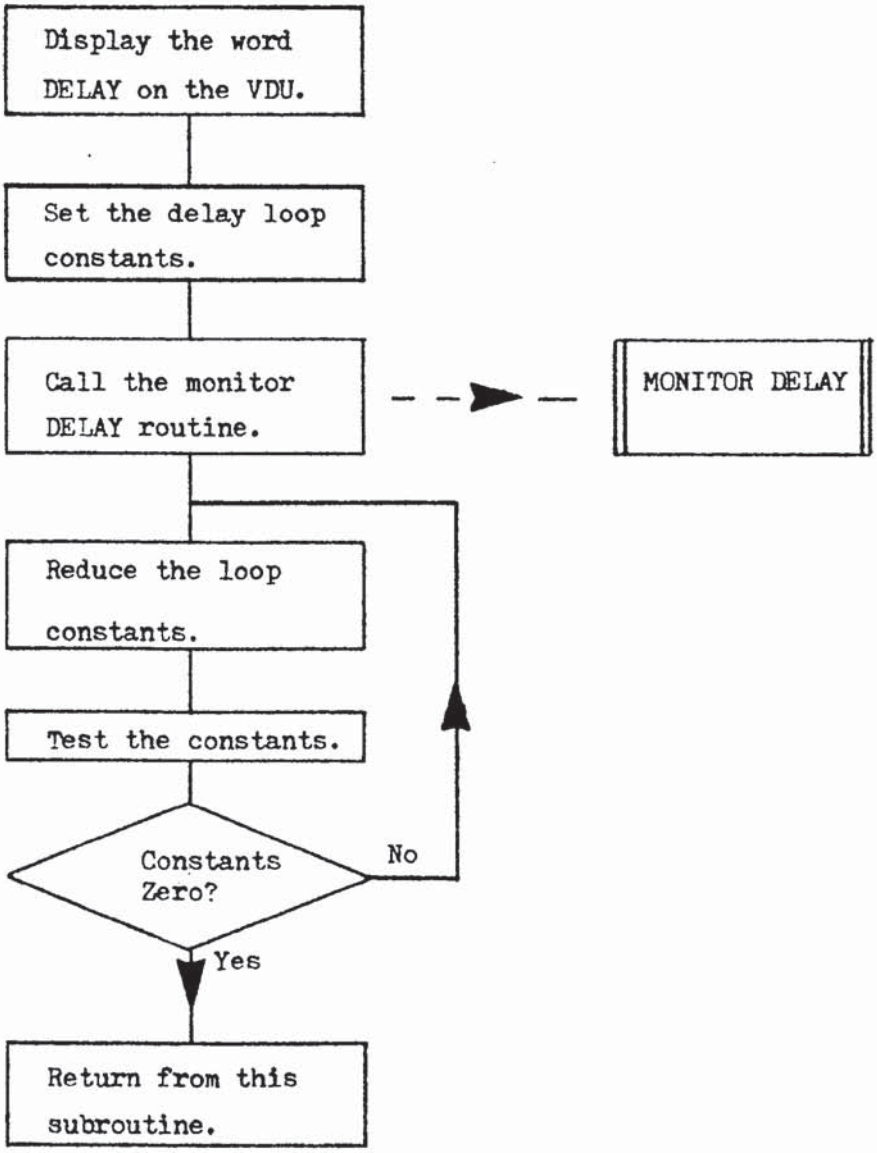
R E S E T L A T C H


Load the accumulator with the code 0F.
Output the code to ports 6 and 7 to set the PIO lines as outputs.
Load the accumulator with the code 00.
Output the code to port 4 to reset the latch.
Load the accumulator with the code 03.
Output the code to port 4 and terminate the reset pulse.
Return from this subroutine.

T E S T L A T C H


Call the RESET LATCH subroutine.
RESET LATCH
Load the accumulator with the code 4F.
Output the code to port 6 to initialise the PIO lines as inputs.
Read the latch condition.
Test the latch condition.
Is the latch set?
No
Yes
Wait for a keyboard input.
Is there an input?
No
Yes
Call the DELAY subroutine.
DELAY
Return from this subroutine.

D E L A Y


Display the word DELAY on the VDU.
Set the delay loop constants.
Call the monitor DELAY routine.
MONITOR DELAY
Reduce the loop constants.
Test the constants.
Constants Zero?
No
Yes
Return from this subroutine.

8.4. System Tests.

(Without the lathe operating)

When operating a new system, for the first time it is much safer to operate with the lathe non activated. The following procedure gives such a test:-

1. Place a peg into hole 4, if the toolpost is in and hole 7, if the toolpost is out.
2. If required the "CF" at location 0D7E may be replaced by "00", in the test programme, this will prevent the system waiting for a keyboard command.
3. Connect a "Test" switch from pin 9 of the latch IC (74 LS 175) to the + 5 volt rail . See figure 2.11 for details.
4. Enter the test programme and execute by typing E 0D00 and pressing Newline.

   No screen response.

   The green test bar will indicate C2 (1100 0010) which is the first control character from the table.

   The latch reset light will be on.
5. Momentarily close the "Test" switch.

   The latch set light will come on.

   The screen will indicate "Delay Step 1 ".

   The green bar will indicate the next control character, 42 (0100 0010)

   The latch reset light will come on.

6. Momentarily close the "TEST" switch again:-

The "LATCH SET" light will come on.

The screen will indicate "Delay Step 2" .

The Green bar will indicate 43 , ie 0100 0011.

The "LATCH RESET" light will come on.

7. Repeat by operating the "TEST" switch momentarily until the system responds with :-

Step 8

END

HOPSYSTEM

There now follows a listing of the above test programme as indicated in section 8.4.1.

8.4.1. System Test Programme.

```
                0001 ;WARD LATHE CONTROL
                0002 ;
                0003 ;Assembled by J HOPTON. 1982.
                0004 ;
0D00            0005 START  ORG  0D00H
0D00 CD500D     0015        CALL 0D50H      ;Call Initialise.
0D03 7E         0016        LD   A,(HL)     ;Pick up character.
0D04 FEFF       0017        CP   255        ;The Stop Code?
0D06 CA300D     0018        JP   Z,0D30H    ;Stop Spindle if so.
0D09 D302       0019        OUT  (02H),A    ;Send out signal.
0D0B 23         0020        INC  HL         ;Increment pointer.
0D0C CD700D     0021        CALL 0D70H      ;Test the Latch.
0D0F 3A210D     0022        LD   A,(0D21H) ;Get counter value.
0D12 3C         0023        INC  A          ;Increment counter.
0D13 32210D     0024        LD   (0D21H),A ;Change Print value.
0D16 EF         0025        RST  28H        ;Display "STEP N ".
0D17 0C0D       0026        DEFB 0CH,0DH    ;Where N is the
0D19 20202053   0027        DEFM /   S/     ;latest step number.
0D1D 54455020   0028        DEFM /TEP /     ;
0D21 30         0029        DEFM /0/        ;
0D22 0D00       0030        DEFB 0DH,00H    ;
0D24 C3030D     0031        JP   0D03H      ;Loop from 0D03.
```

```
                0032 ;WARD LATHE CONTROL.
                0033 ;
                0034 ;Assembled by J HOPTON. 1982.
                0035 ;
0D30            0036 STOP   ORG  0D30H
                0037 ;
                0038 ;STOP SPINDLE
                0039 ;
0D30 3E42       0040        LD   A,42H      ;Stop the spindle.
0D32 D302       0041        OUT  (02H),A
0D34 EF         0042        RST  28H        ;Display "END".
0D35 0C         0043        DEFB 0CH
0D36 20202020   0044        DEFM /    /
0D3A 454E44     0045        DEFM /END/
0D3D 0D00       0046        DEFB 0DH,00H
0D3F DF         0047        RST  18H        ;End of program.
0D40 5B         0048        DEFB 5BH
```

```
                0049 ;WARD LATHE CONTROL
                0050 ;
                0051 ;Assembled by J HOPTON. 1982.
                0052 ;
0D50            0053 INIT   ORG  0D50H
                0054 ;
                0055 ;INITIALISE
                0056 ;
                0057 ;
                0058 ;
0D50 CD600D     0059        CALL 0D60H     ;Call Reset Latch.
0D53 21B00D     0060        LD   HL,0DB0H  ;Point to the table.
0D56 3E30       0061        LD   A,30H     ;Zero routine counter.
0D58 32210D     0062        LD   (0D21H),A ;at the Print line.
0D5B C9         0063        RET
                0064 ;
                0065 ;
```

```
                0066 ;WARD LATHE CONTROL
                0067 ;
                0068 ;Assembled by J HOPTON. 1982.
                0069 ;
0D60            0070 RESET  ORG  0D60H
                0071 ;
                0072 ;RESET LATCH
                0073 ;
                0074 ;
0D60 3E0F       0075        LD   A,0FH   ;Set the PIO
0D62 D307       0076        OUT  (07H),A ;as outputs.
0D64 D306       0077        OUT  (06H),A ;
0D66 3E00       0078        LD   A,00H   ;Reset the latch,
0D68 D304       0079        OUT  (04H),A ;with a pulse.
0D6A 3E03       0080        LD   A,03H   ;End the pulse.
0D6C D304       0081        OUT  (04H),A ;
0D6E C9         0082        RET
                0083 ;
                0084 ;
```

```
                0085 ;WARD LATHE CONTROL.
                0086 ;
                0087 ;Assembled by J HOPTON. 1982.
                0088 ;
0D70            0089 TEST   ORG  0D70H
                0090 ;
                0091 ;TEST LATCH
                0092 ;
                0093 ;
                0094 ;
0D70 CD600D     0095        CALL 0D60H      ;Call Reset Latch.
0D73 3E4F       0096        LD   A,4FH      ;Set PIO lines
0D75 D306       0097        OUT  (06H),A    ;as inputs.
0D77 DB04       0098        IN   A,(04H)    ;Read Latch.
0D79 FE03       0099        CP   03         ;Latch Set?
0D7B C2770D     0100        JP   NZ,0D77H   ;If not wait.
0D7E CF         0101        RST  08H        ;Wait for keybd input.
0D7F CD900D     0102        CALL 0D90H      ;Call Delay.
0D82 C9         0103        RET
                0104 ;
```

```
                0105 ;WARD LATHE CONTROL.
                0106 ;
                0107 ;Assembled by J HOPTON. 1982.
                0108 ;
0D90            0109 DELAY  ORG  0D90H
                0110 ;
                0111 ;DELAY
                0112 ;
                0113 ;
                0114 ;
0D90 EF         0115        RST  28H      ;Display "DELAY".
0D91 0D0D       0116        DEFB 0DH,0DH  ;
0D93 20202044   0117        DEFM /   D/   ;
0D97 454C4159   0118        DEFM /ELAY/   ;
0D9B 0D00       0119        DEFB 0DH,00H  ;
0D9D 06A0       0120        LD   B,160    ;Set Delay time.
0D9F 3EFF       0121        LD   A,255    ; "    "    "
0DA1 FF         0122        RST  38H      ;Delay Loop.
0DA2 10FB       0123        DJNZ 0D9FH    ;  "     "
0DA4 C9         0124        RET
                0125 ;
                0126 ;
```

```
                0127 ;WARD LATHE CONTROL.
                0128 ;
                0129 ;Assembled by J HOPTON. 1982.
                0130 ;
0DB0            0131 TABLE  ORG  0DB0H
                0132 ;
                0133 ;TABLE OF MOVEMENTS.
                0134 ;
                0135 ;
                0136 ;
                0137 ;C2    = Slow anticlock rotation.
                0138 ;42    = Stop rotation.
                0139 ;43    = Slow clockwise rotation.
                0140 ;42    = Stop rotation.
                0141 ;82    = Fast anticlock rotation.
                0142 ;02    = Stop rotation.
                0143 ;03    = Fast clockwise rotation.
                0144 ;02    = Stop rotation.
                0145 ;FF    = End of table code.
                0146 ;
                0147 ;
                0148 ;These codes should be entered into memory
                0149 ;starting at location 0DB0
                0150 ;
                0151 ;
```

8.4.2. Lathe Testing Under Microprocessor Control.

When the previous test is completed the lathe may then be operated directly from the microprocessor. If keyboard control is required to force a stop after every operation, then the code "CF" must be inserted at memory location "0D7E" manually in the teset programme. Pressing any key would restart the programme. However if continuous operation is desired then the codes "00" must be left in memory location "0D7E".

During the previous test two pegs were entered in diode holes 4 and 7 for safety so that the system could be tested without moving the lathe components, these pegs must now be removed, however the peg action is now under microprocessor control and so codes "04" or "20" must be added to each table entry, except FF the stop code.

These new codes are required so that the slide position may be controlled, code "04" or 0000 0100 is required to either send the tool post in, or when it has come to rest in the "In " position. Code "20" or 0010 0000 is required to either send the tool post out, or when it has come to rest in the "Out" position. There now follows a table of movements to control rotation and toolpost movement together, the table must end with the stop code "FF".

```
0001 ;WARD LATHE CONTROL.
0002 ;
0003 ;Assembled BY J HOPTON. 1982.
0004 ;
0005 ;Demonstration Table of Movements.
0006 ;
0007 ;These codes should be entered into memory
0008 ;starting at location 0DB0, end with FF.
0009 ;
0010 ;
0011 ; 62=42+20 = Stop Rotation with Tool Out.
0012 ; A2=82+20 = Fast Anti-rot with Tool Out.
0013 ; 86=82+04 = Fast Anti-rot with Tool In.
0014 ; 46=42+04 = Stop Rotation with Tool In.
0015 ; 07=03+04 = Fast Clock-rt with Tool In.
0016 ; 23=03+20 = Fast Clock-rt with Tool Out.
0017 ; 22=02+20 = Stop Rotation with Tool Out.
0018 ; A2=82+20 = Fast Anti-rot with Tool Out.
0019 ; 86=82+04 = Fast Anti-rot with Tool In.
0020 ; 46=42+04 = Stop Rotation with Tool In.
0021 ; FF       = End of Table Code.
0022 ;
0023 ;
```

CONCLUSIONS.

CHAPTER 9.

CHAPTER 9.

CONCLUSIONS

9.1. The Industrial Advantages of the System.

British industry has never before faced so much competitive production and so contributions towards microprocessor control especially in a Direct Numerical Control system, is essential to achieve or maintain competitiveness in a technological era.

The microprocessor system developed in this work utilises up to the minute technology and can thus bring considerable economical and technical benifits, particularly to the small and medium sized industries, using the old generation of Numerical Control machines. These industries are lagging behind their competitors, who utilise improved production techniques brought about mainly by microprocessor control and computer aided manufacture.

This system is an inexpensive and highly cost effective method of retrofitting a microprocessor controller to an existing machine, without the necessity to change or purchase any special new machine tools. The costing schedule which is minimal, is detailed in the next section.

9.2. Cost Schedule.

The retrofitting costs of a microprocessor system are minimal, compared to the cost of replacing a machine with the modern counterpart. This also gives a chance to spread the cost of replacement over several years.

The following list gives the approximate value of materials required to implement a small control system, it does not include labour.

ITEM.	PURPOSE.	COST.
Z80 16K System.	To hold the monitor programme and control all peripherals.	£ 500.00
Television Set.	To act as a Video Monitor.	£ 90.00
Cassette.	To act as an external memory.	£ 50.00
EPROM Programmer.	To enable custom design of extra programmes.	£ 100.00
EPROM Eraser.	To clean and prepare EPROMS.	£ 100.00
Hardware.	To connect the system to an external machine. Cost will depend very much on the application, an example is quoted and the prices are typical in 1981.	£ 200.00
	Approximate Total.	£1000.00

9.3. System Development.

The developed system utilises the versatile Zilog Z80 microprocessor chip which is fast becoming one of the British industry standards. It is backed up with a powerful custom built 2K machine code in two EPROMS. The use of machine code gives several advantages which include the very efficient use of Random Access Memory , a fast response time which is essential in any machine control application. Machine code is admirably suited to this task and the Z80 microprocessor chip has many advantages over its rivals when operating in the field of machine control.

The monitor or administrative control programme is situated in two 1K EPROMS giving further flexibility because they may be easily replaced at any time, should the routines require further customising to suit some particular application. Further economy of memory is provided by the inbuilt subroutines which are prewritten for input and output control and these may be called by a programme with an overhead of only 3 memory Bytes , thus saving vast amounts of RAM , with the convenience of having ready made control subroutines as and when required. A full listing of the Monitor machine code system can be found in Appendix 3.

9.4. Interface Control Details.

The Z80 microprocessor controller was connected to the lathe pegboard by means of an interface control box containing electronic latches connected to the address and data bus of the microprocessor. These latches which act as temporary memories enable the Central Processing Unit to present data to the lathe and then continue with another task, until the lathe has completed the alloted task. Use was also made of a peripheral input output chip MK 3881, to allow the lathe electronics to be interrogated by the microprocessor . This gives a handshaking facility enabling the lathe to request further information or commands from the CPU if and when required.

The two way communication method provides a far more versatile control mode than was ever possible with the original pegboard, with the added advantage of eliminating the time consuming effort of setting up the pegs in the pegboard whilst the lathe stood idle, with loss of actual production time.

9.5. Details of the Main Controller.

The main controller which consists of a Z80 microprocessor has been augmented by several additional devices, such as an Universal Assynchronous Receiver Transmitter, a Parallel Input Output Adapter and a Video Modulator. Thus the unit has the possibility of communicating with many devices, such as a Cassette recorder which may store further routines for a particular production task; a standard domestic television set, thus reducing the necessity to purchase an expensive VDU and a serial or parallel printer so that a hard copy may be obtained to record any programme or results of a test run. The parallel printer capability allows the system to produce three dimensional high resolution graphics.

The system boasts two peripheral ports and the UART supports two serial methods of communication, RS 232 and the 20mA current loop. Cassette and serial Baud rates are variable, leading to an extremely powerful and versatile compact low cost unit. The system monitor has been configured to operate a memory plane peripheral, such as a domestic television set and the serial output simultaneously, thus enabling the operator to follow control sequence instructions on the TV screen and graphics, such as following tool contours on a graphics VDU, this dual mode visual output is quite unique in microprocessor control applications.

9.6. System Tests and Results.

Test sequencies were developed to allow the system to be assessed and these sequencies are given in Chapter 8. The routines were used to cut some components and the results showed a great improvement in operating ease and manufacturing versatility over the original method. The interface box was improved later by placing an LED test readout connected to its control lines, this enables the lathe control operation to be monitored at the binary level and will be very useful should the system be expanded as proposed later. Figure 9.1 illustrates the layout of the interface network .

Due to the fact that the equipment in the laboratory were modified to a common communication standard, it is now possible for this Z80 controller to be interfaced into a Diect Numerical Control cell, thus opening up many other possibilities and facilities. Further work would be greatly facilitated by system expansion as detailed in the next section.

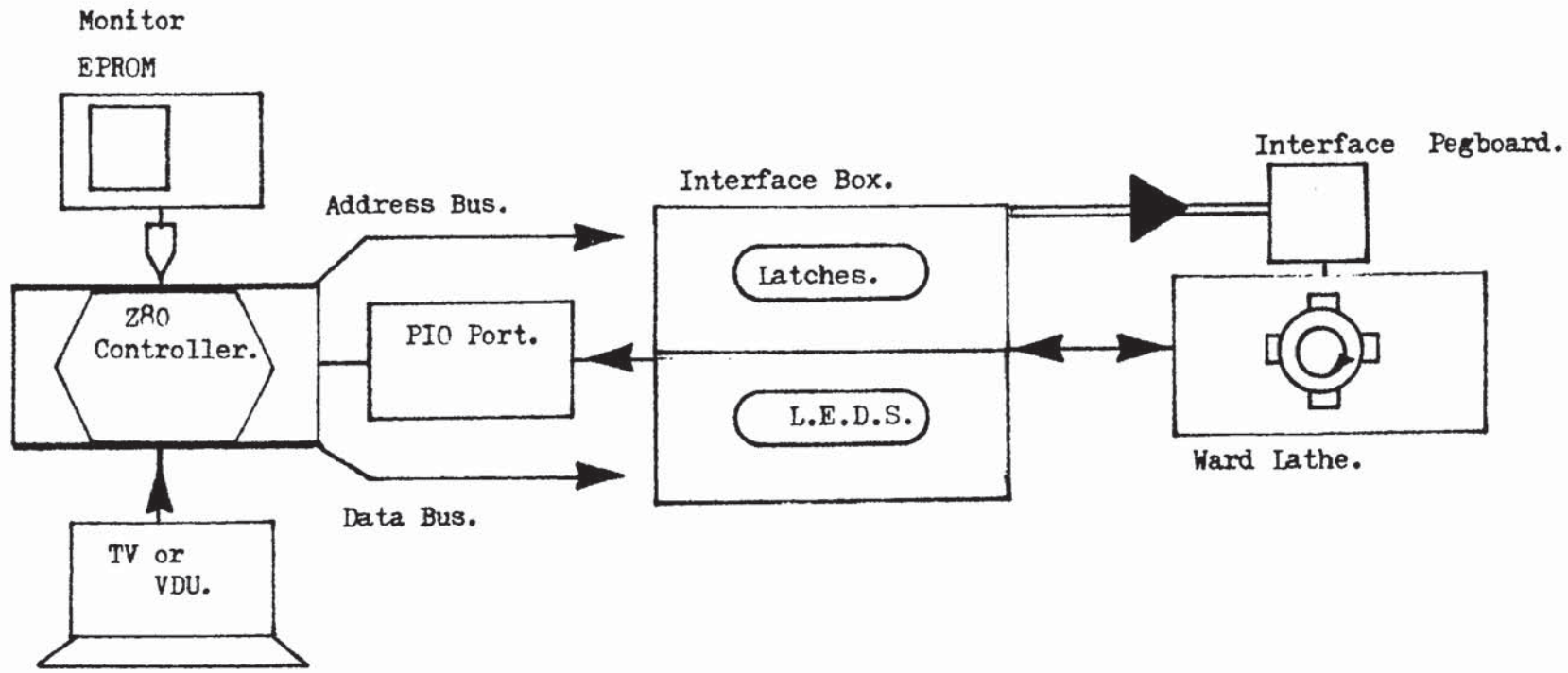


Fig. (9.1.) Microprocessor To Lathe Interface.

9.7. System Expansion.

Albeit the system memory was adequate for this limited budget project, it could be greatly improved by the addition of buffering and further memory. This would add further versatility giving the possibility of running an Assembler and other high level languages such as FORTRAN or BASIC, the cost of course would be in excess of that possible within a project budget. An expanded system would also allow the central processor to be utilised as an executive monitor within a DNC cell as indicated in figure 9.2.

The Author has utilised his experience in this work and now acts as a consultant to several local industries in his locality, some of which have financed some private research at the author's own home laboratory. This enabled the author to try out some of the improvements and example designs are included in the next chapter of some of the working prototypes, such as an EPROM programmer and some high level language software to enable tool contours to be shown visually in simulated three dimensions. Examples of this three dimensional capability is included in Appendix 5. A section now follows indicating some of the future proposals for expanding the system.

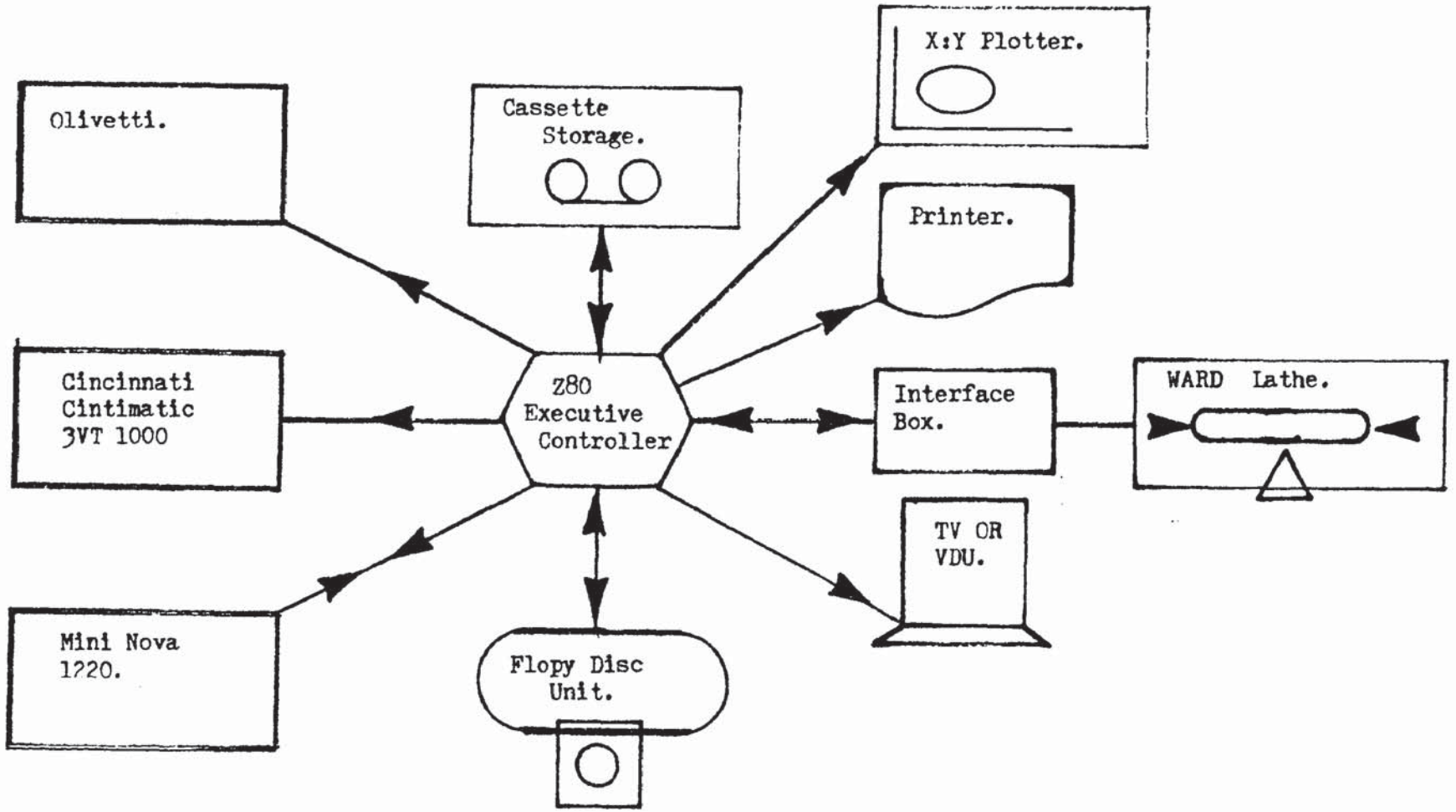


Fig. (9.2.) Proposed DNC Cell.

FUTURE WORK.

CHAPTER 10.

CHAPTER 10.

FUTURE WORK.

10.1. System Developments.

If the system is to be improved by further development, then some fundamental extras are required and these could include some of the following :-

a) Added memory, so that larger and better control programmes could be written and monitored.

b) The ability to store set programmes in EPROMS and then simply plug in a specific routine as and when required, to perform a certain task.

c) The ability to programme these specific tasks into the EPROMS so that a library of routines could be built up.

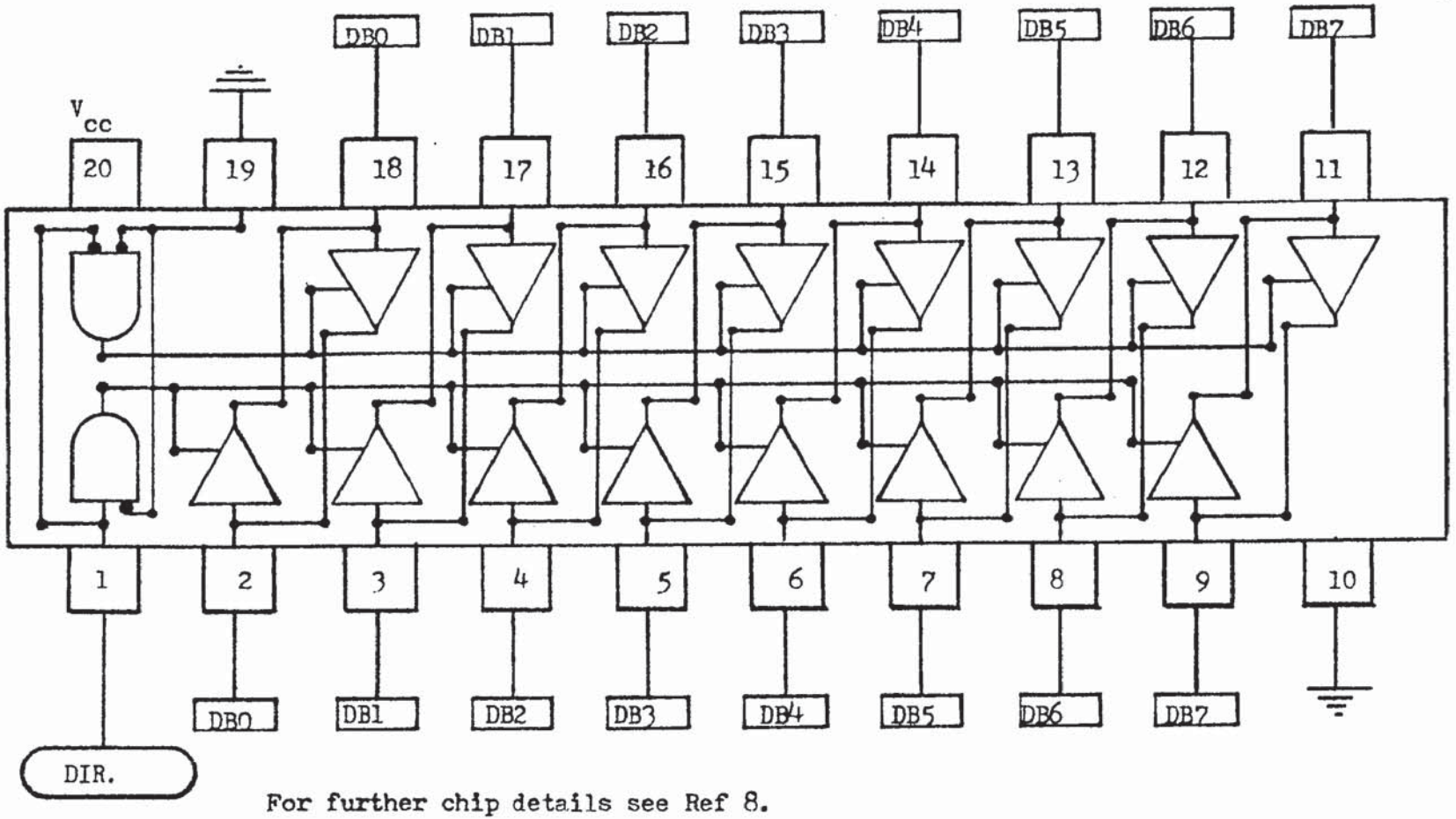
d) The visual display of tool movement, or contouring of the proposed workpiece.

10.1.1. Memory Expansion.

The single Z80 microprocessor board is not buffered in the original form , hence the first task in any memory expansion developement is to ensure that the CPU is adequately buffered from the additional components. There are two main areas that require treatment as detailed below :-

10.1.2. Data Lines.

Figure 10.1 shows a proposed circuit indicating that the data lines be connected to a 74LS245 (9) integrated circuit octal bus transceiver,giving eight non-inverting three state outputs. The package provides bidirectional data transfer with full CPU buffering. The enable gate of the 74LS245 is on pin 19 and is permanently enabled by grounding it to the zero volts level. It is also necessary to provide directional control and this is provided on pin 1 by an acurately timed pulse which will be dealt with in greater detail later.



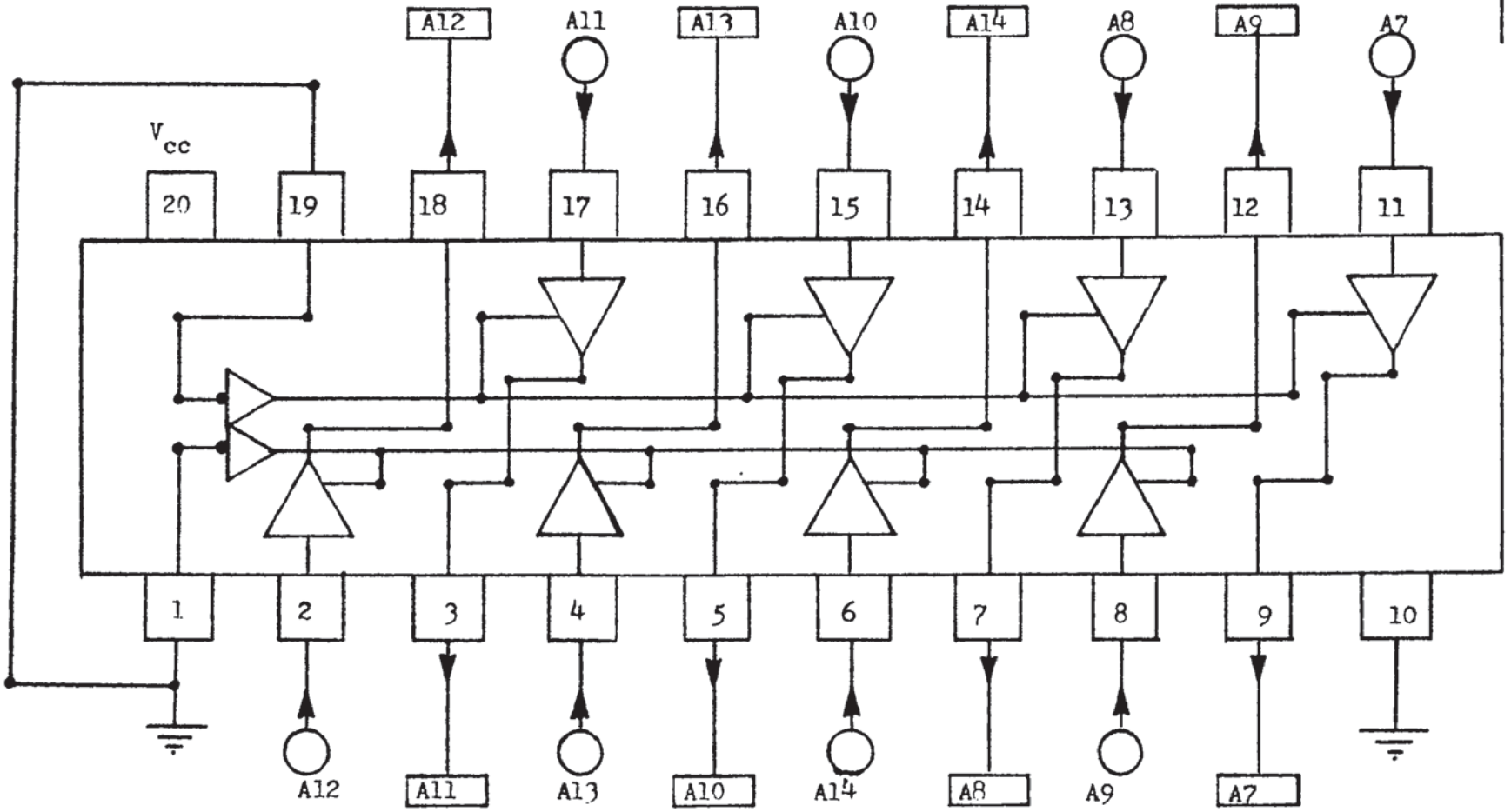
For further chip details see Ref 8.

Fig. (10.1.) DATA BUFFER CHIP 74 245

10.1.3. Address Line Buffering.

The address lines only require buffering in one direction because the address sequence is always sent out from the CPU to the other devices , this makes the buffering task a little easier to implement.

Figure 10.2 shows the address lines connected to a 74LS244 (9) integrated circuit octal buffer and line driver which has three state outputs . These octal buffers and line drivers are designed to specifically improve the performance and density of the three state memory address drivers . There is a selection of combinations of inverting and non-inverting outputs available to the user. The 74LS244 device features a high fan out with inproved fan in and a very low noise margin of 400 mV. The gates have been permanently enabled by grounding pins 1 and 19 to the zero volts level as shown in figure 10.2.



For further chip details see Ref 8.

Fig. (10.2.) 74 244 ADDRESS BUFFER CHIP.

10.2. EPROM Decoding.

Extreme care must be taken when extending any memory system to ensure that no address or data line conflict occurs , such as two memories being read simultaneously. To avoid such conflicts strictly timed address decoding and timing pulses are required. Figure 10.3 illustrates an example in which the three adress lins A_{12} , A_{13} and A_{14} are connected into a 74138[9] three line to eight line decoder de-multiplexer.

Figure 10.4 gives the truth table of the 74138 device and examination of the table shows that the output on pin 7 will be low only when all three of the address lines are at a high logic level . This means that the output from pin 7 can be utilised as a signal to Enable an EPROM located at address location 7000 hexadecimal . This will occur because the number 7 in the address has a binary equivalent of 0111 and each one in the binary number is represented by one of the address lines being at logic level 1.

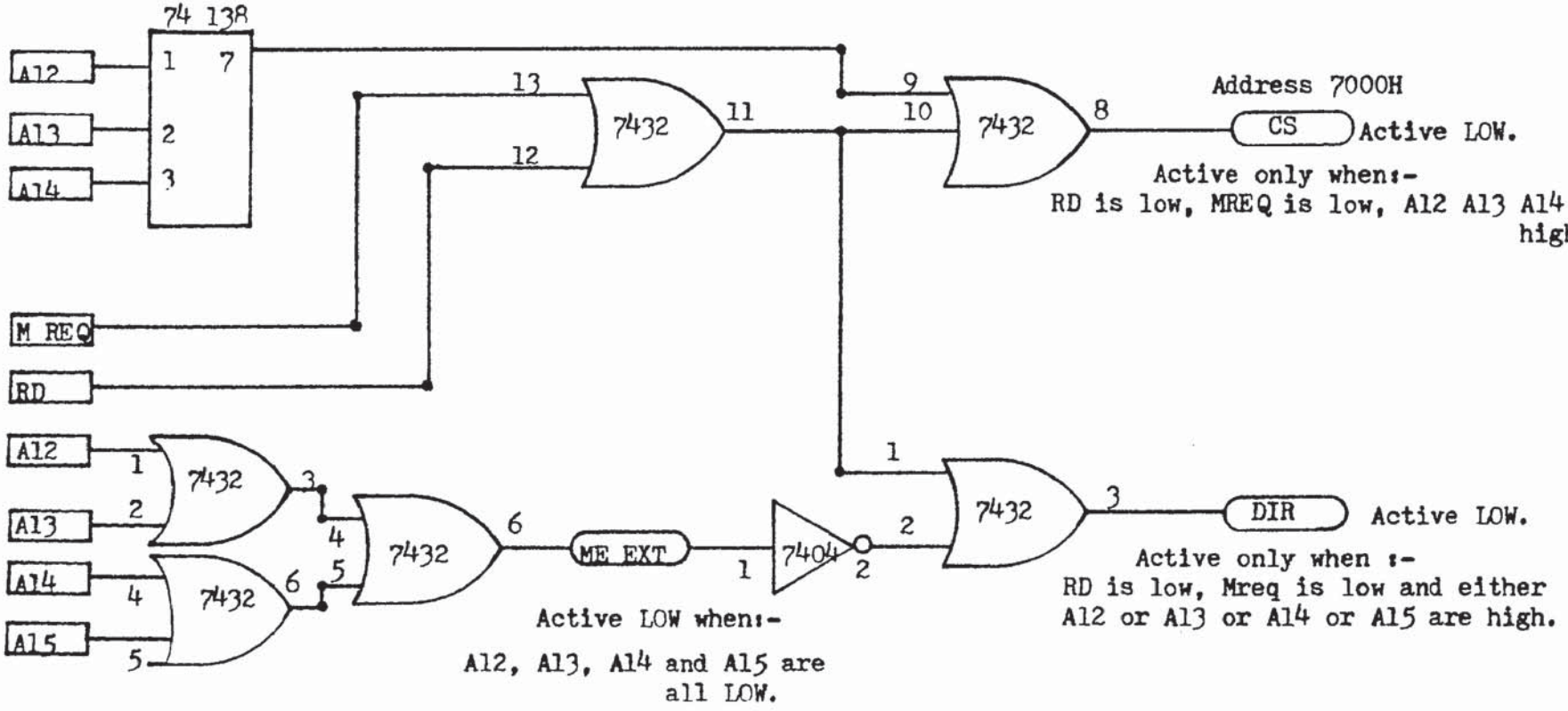


Fig. (10.3.) MEMORY EXTENSION DECODING.

74LS138 TRUTH TABLE.

INPUTS					OUTPUTS								
ENABLE		SELECT											
G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Pin Code.
6	4/5	3	2	1	15	14	13	12	11	10	09	07	Pin No.
X	H	X	X	X	H	H	H	H	H	H	H	H	
L	X	X	X	X	H	H	H	H	H	H	H	H	
H	L	L	L	L	L	H	H	H	H	H	H	H	
H	L	L	L	H	H	L	H	H	H	H	H	H	
H	L	L	H	L	H	H	L	H	H	H	H	H	
H	L	L	H	H	H	H	H	L	H	H	H	H	
H	L	H	L	L	H	H	H	H	L	H	H	H	
H	L	H	L	H	H	H	H	H	H	L	H	H	
H	L	H	H	L	H	H	H	H	H	H	L	H	
H	L	H	H	H	H	H	H	H	H	H	H	L	

H = High Level Logic.

L = Low Level Logic.

X = Irrelevant.

Fig. (10.4.)

It is also necessary for pins 4 (G2A) and 5 (G2B) to be connected to the zero volts level and pin 6 (G1) to be connected to the logic 1 level . This requirement follows from the truth table in figure 10.4.

Simply decoding the address lines to give a chip select at address 7000 hexadecimal is not enough to prevent bus conflict , it is also necessary to time the select signals accurately. The $\overline{RD}$ and $\overline{MREQ}$ pulses are used to facilitate the decoding and timing by feeding both signals through a 7432 (9) two input logic OR gate. The composite output is then fed together with the address decode signal into a second OR gate , the output of which is used as the chip select pulse $\overline{CS}$ for the EPROM at location 7000 hexadecimal. This $\overline{CS}$ signal will only be active when the $\overline{RD}$ and $\overline{MREQ}$ signals are both at the low logic level and the three address lines A_{12} , A_{13} and A_{14} are all at the high logic level. The resulting signal gives correct address decoding at the correct time thus preventing any possible data conflict.

10.3. Direction Pulse to the Data Buffer.

A similar problem to the EPROM decoding exists regarding the timing of the direction pulse to the bidirectional 74LS245 (9) bus transeiver. It is not only important to time the data correctly but to ensure that the data is steered in the correct direction , at precisely the correct time. Figure 10.5 shows the necessary timing requirements with the direction decoding.

The $\overline{RD}$ and $\overline{MREQ}$ pulses are combined through the two input 7432 (9) logic OR gate and as previously, the same combined output signal can be utilised, removing the necessity for circuit duplication. The four higher address lines A_{12} , A_{13} , A_{14} , A_{15} are all combined through the 7432 OR gate to provide the composite $\overline{MEXT}$ signal. It is necessary for this signal to be injected into the system by changing the position of link LK5 on the board, the $\overline{MEXT}$ signal is then fed through an inverter before joining the other $\overline{RD}$: $\overline{MREQ}$ composite pulse to a further OR gate. The final $\overline{DIR}$ pulse becomes active low only when both the $\overline{MREQ}$ and $\overline{RD}$ signals are low and also either of the address lines are at the high logic level.

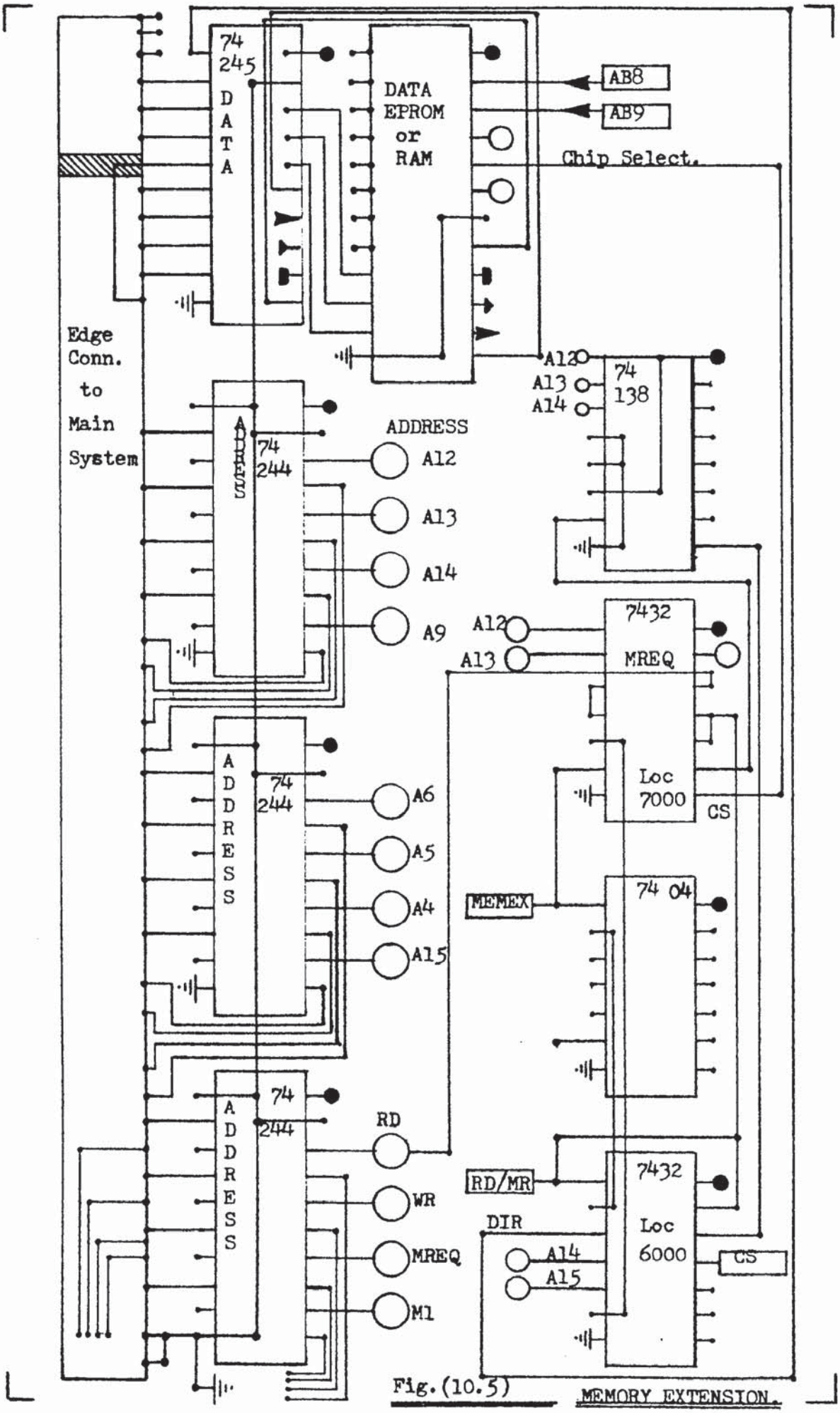


Fig.(10.5) MEMORY EXTENSION.

10.4. EPROM Programming and Timing.

Another requirement for system improvement is the ability to write a custom designed programme into a permanent memory cell such as an EPROM. An EPROM programmer design was developed together with the necessary software , the design is included later , see figures10.7 and10.8 and together with the previous memory expansion design , could form an excellent subject for future developement work . The first task in the design was to decide on which type of EPROM to use and the choice was made of the popular, easy to obtain 2708 type (5). The design was later modified to encompass the 2716 and 2704 types (5) as well, giving greater flexibility.

The next consideration for programming EPROMS is the very strict timing pulses that are required, figure 10.6 shows the timing requirements which must be met for successful programming. The 2708 EPROM (6) is electrically programmable and Ultra Violet erasable by illuminating the chip through a transparent window , this exposure induces a flow of photo current from the floating gate to the substrate, thereby discharging the gate to the initial state.

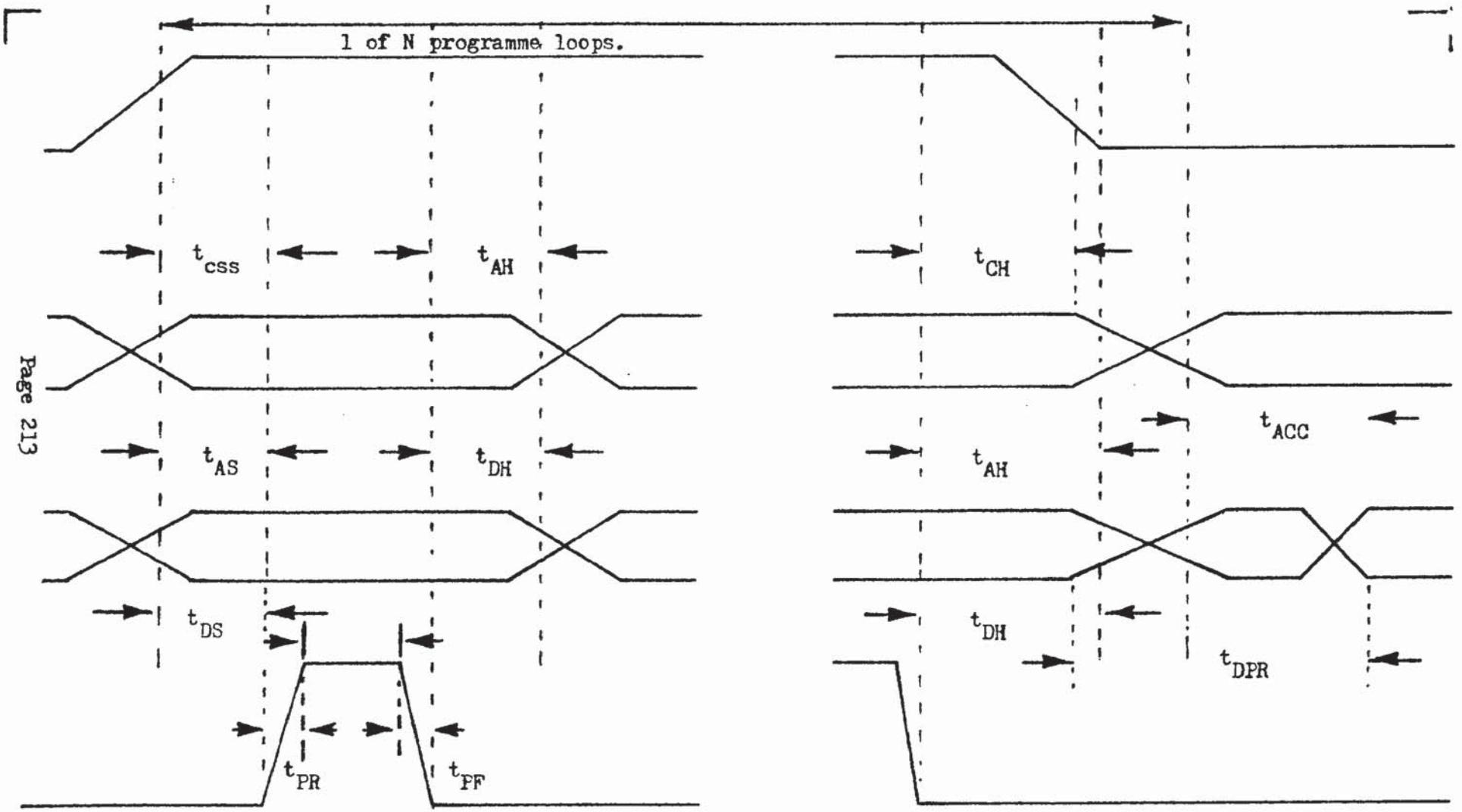


Fig. (10.6.) 2708 PROGRAMMING PULSE TIMING.

10.5. EPROM Erasure: Pulsing.

An Ultra Violet source of light at 2357 $\overset{o}{A}$ yielding a total dosage of 10 Wsec / cm^2 is required for correct erasure with an exposure time of about 20 mins. Programming occurs by raising the $\overline{CE}$ / $\overline{WE}$ input on pin 20 to the + 12 volt level and the data should be presented as 8 parallel bits to the data output lines on pins 9 to 11 and pins 13 to 17. In this mode the ouput pins serve as input lines. Caution should be observed regarding the end of a programme sequence as the $\overline{CS}$ / $\overline{WE}$ pulse falling edge transition must occur before the first address transition , when changing from a programme cycle to a read cycle. The programme pin should also be pulled down to the V_{ILP} level with an active rather than a passive device, as the pin will source a small current I_{IPL} when the CS / WE pin is at the V_{IHW} level of + 12 volts and the programme pulse is at the V_{ILP} level.

Figures 10.7and10.8 show a proposed circuit design for the programmer which could be utilised for future development. When erased all the bits of the 2708 EPROM will be at logic level 1 and each location will contain FF hexadecimal, hence information is introduced by selectively programming 0 into the bit locations. One programme pulse per address is applied to pin 18 when the address and data lines are correctly set up.

One pass through the addresses is defined as a programme loop and the number of loops required [N] is a function of the programme pulse width [t_{pw}] according to the equation :-

$$N \times t_{pw} = 100 \text{ msec.}$$

The width of the programme pulse lies between :-

0.1 msec and 1.0 msec.

This indicates that the number of programme loops [N] lies between

100 when t_{pw} is 1 msec and

1000 when t_{pw} is 0.1 msec.

It should be noted that there must be "N" successive loops through all the " 1K " addresses, it is not permitted to apply "N" programming pulses to an address and then change to the next address to be programmed.

Note that " 1K " is 1024 decimal, $1024 = 2^{10}$.

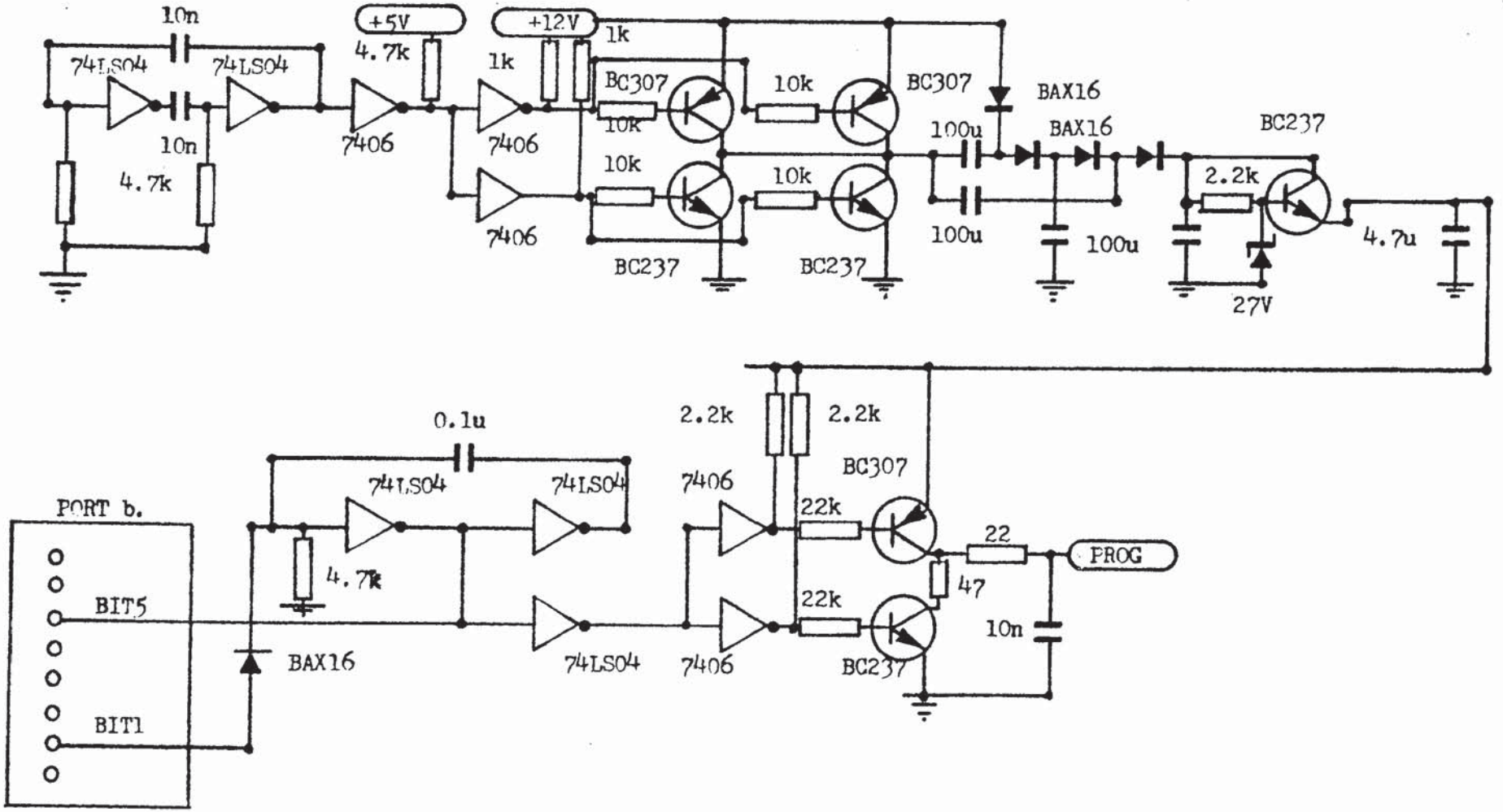


Fig. (10.7.) 26Volt OSCILLATOR PULSE.

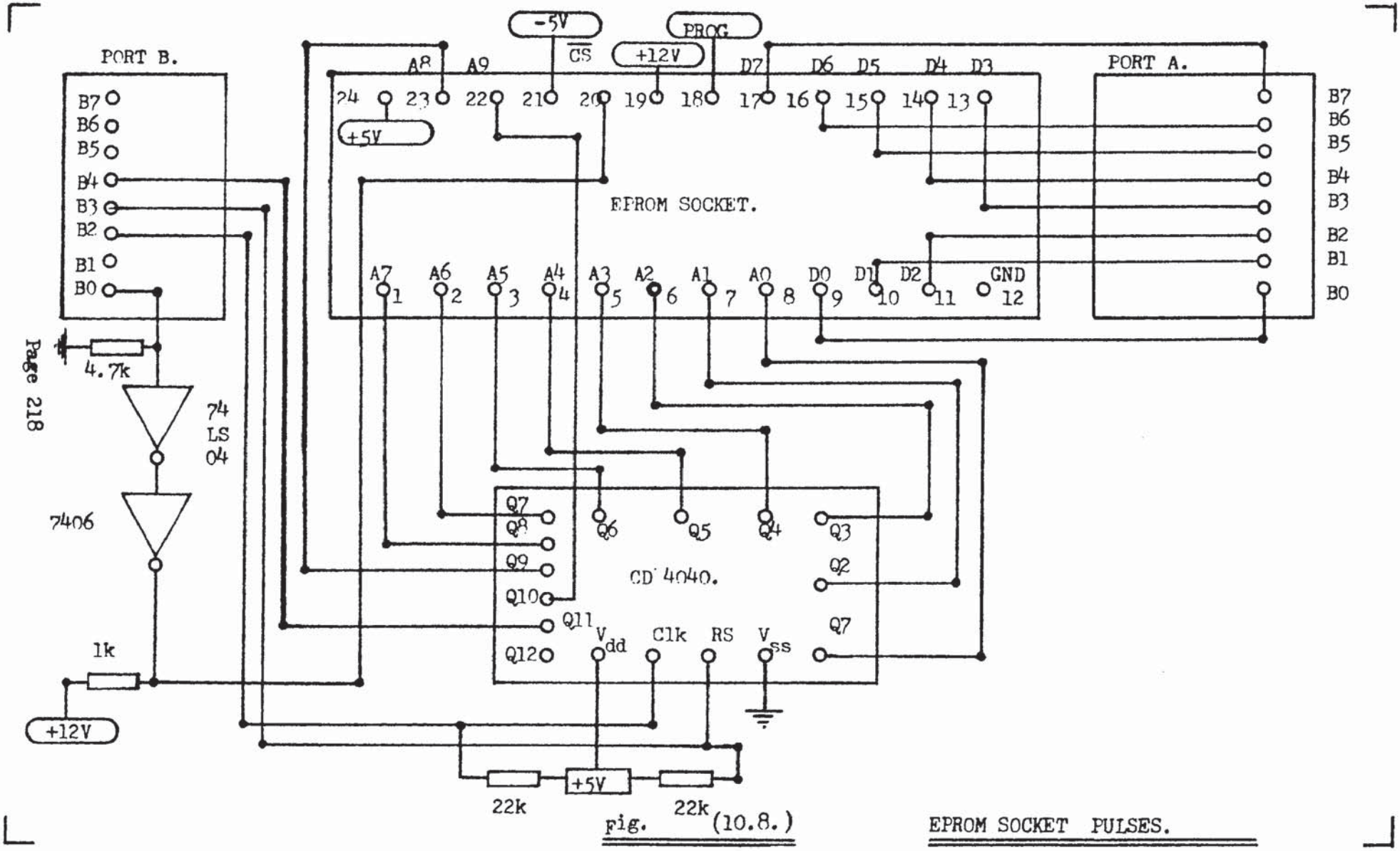
10.5.1. Programmer Design Circuit.

The heart of the design is a 74LS04 integrated circuit oscillator (9) which drives an inverter to produce the 26 volt power source, which can be utilised as the programming pulse. The pulse must be timed and gated by signals from the CPU and then applied to the programming pin 18 on the 2708 EPROM. Control is ensured by feeding pulses from the CPU through Port B of the PIO as shown in figure 10.8. These pulses ensure two conditions :-

a) Correct timing of the $\overline{CS}$ voltage pulse to pin 20.

b) Correct timing of the PROG pulse with 26 volts to pin 18.

Address decoding is achieved by the use of a Cosmos CD 4040 (4) (5) twelve stage ripple carry binary counter divider chip,which consists of an input pulse shaping circuit and a twelve stage ripple carry binary counter stage. The counter may be reset to the zero state by placing a high logic level on the reset line. A master slave flip flop is used for each counter stage and the state of the counter is binary stepped by the negative transition of the buffered input. All derived signals from the decoder with the $\overline{CS}$ and PROG pulses are fed to the EPROM socket. Data lines from the socket are taken from Port A of the PIO , which has address 06 for control and address 04 for data transference. Port B uses address 7 and 5.



Page 218

fig. (10.8.) EPROM SOCKET PULSES.

10.5.2. EPROM Programmer Control Sequence.

A suggested programme to control this design of EPROM programmer is included in appendix 7 , the unit is used at home by the author and is well tried and tested.

The only point to note is that the sequence has been written to operate with the CPU running at a clock rate of 4 MHz . It is possible to either select this faster rate on the Z80 system or to change the timing of the programme to run at 2 MHz . The first method would be easier for the reader , albeit the second is not too difficult with a knowledge of machine code.

The programme is in several sections and executing from location "0C80" gives three options :-

a) To programme an EPROM.

b) To reprogramme.

c) To copy the contents of an EPROM from the programmer into the system memory.
This last facility can only be used if the system's memory has been expanded.

10.6. Graphics on the VDU.

Some of the topics for future development include the visual display of tool movement by utilising the graphics mode capability of the VDU (12) and the visual contouring of the proposed workpiece. These aims were furthered by interfacing the RG 512 Lear Siegler ADM 3A VDU terminal directly to the Z80 RS 232 port. The VDU has both vector and point mode capability.

10.6.1. Vector Mode.

This mode performs automatically giving vector generation from vector endpoints and the terminal uses 512 points horizontally, with 250 points vertically. The X10 option scales the graphics grid,giving 1024 by 780 points and thus the two versions require different vector endpoint coordinates. The vector mode may be entered in two ways:-

a) Manually , by pressing CTRL and] simultaneously.

b) By programme , by sending out the hexadecimal code 1D.

10.6.2. Point Mode and Clear Screen.

The screen may be cleared in two ways :-

a) Manually , by pressing CTRL and y simultaneously.

b) By programme , by sending out the hexadecimal code 19.

Point Mode is activated also in two ways:-

a) Manually , by pressing CTRL and \ simultaneously.

b) By programme , by sending out the hexadecimal code 1C.

10.6.3. Connecting the VDU to the Z80 System.

The VDU must be connected to the RS 232 output port of the Z80 and the "TTI : TAPE " switch moved to the "TTI", up position, with the Baud select switch set to 300 Baud, down. The switches are on the Z80 rear panel.

To operate the VDU enter " X20 Newline " on the Z80 keyboard, the VDU will then respond, however if the Z80 reset button is pressed for any reason, control is lost and "X20 Newline" must again be entered to reinitialise the systems, before two way communication is possible. Note that "Newline" is a special key on the right of the Z80 keyboard, it is not necessary to type in the word.

Control signals may be sent from the Z80 by two ways.

a)Direct Transfer.

If the correct hexadecimal codes are entered, in a table starting at some convenient location eg 0E00 , then the entries may be sent directly to the VDU as follows:-

Type in	X20	Newline
Type in	T 0E00 0EC0	Newline

Each entry will be sent out automatically to the VDU from locations "0E00 to 0EC0" by the system monitor, the VDU will respond on receipt of the control or coordinate characters.

The Z80 system will terminate the output routine when the table pointer reaches location 0EC0.

ie T 0E00 0EC0 Newline

means type out or transfer data from the starting location of 0E00 to the terminating location of 0EC0.

b) Programme Transfer.

If transfer and control is required within a programme, then the following code entries will control the output, if the same data is entered in a table, from locations 0E00 to 0EC0.

PAGE NUMBERING AS ORIGINAL

10.6.4. Sample Data.

There now follows a data table for use by the previous programme, which will control the output characteristics of the VDU.

Note.

There are three control characteristics used in the table.

a) "19" Hexadecimal = Clear the screen.

b) "1C" Hexadecimal = Set the "Point Mode".

c) "1D" Hexadecimal = Set the "Vector Mode".

All the other characters in the table are used as vector coordinates. The data is in three sections.

a) Locations 0E00 to 0E15.

These characters are used to send out control symbols to draw a plane rectangle on the screen, in the vector mode.

The data starts with code "19" hexadecimal, which clears the screen.

The next code "1D" hexadecimal sets the vector mode, there then follows the "Y" coordinate pair of "23 , 64" which are the "Y" coordinates of the left hand bottom corner of the rectangle. The next pair "23 , 58" are the "X" coordinates of the left hand bottom corner of the rectangle.

b) Locations 0E16 to 0E7A.

This section holds the control data characters required to draw a circle inside the rectangle.

Data starts with the "1C" code which forces the VDU into the "Point Mode" , there then follows the hexadecimal pairs "31 , 70" and "2F , 54" which are the first "Y" and "X" coordinates respectively.

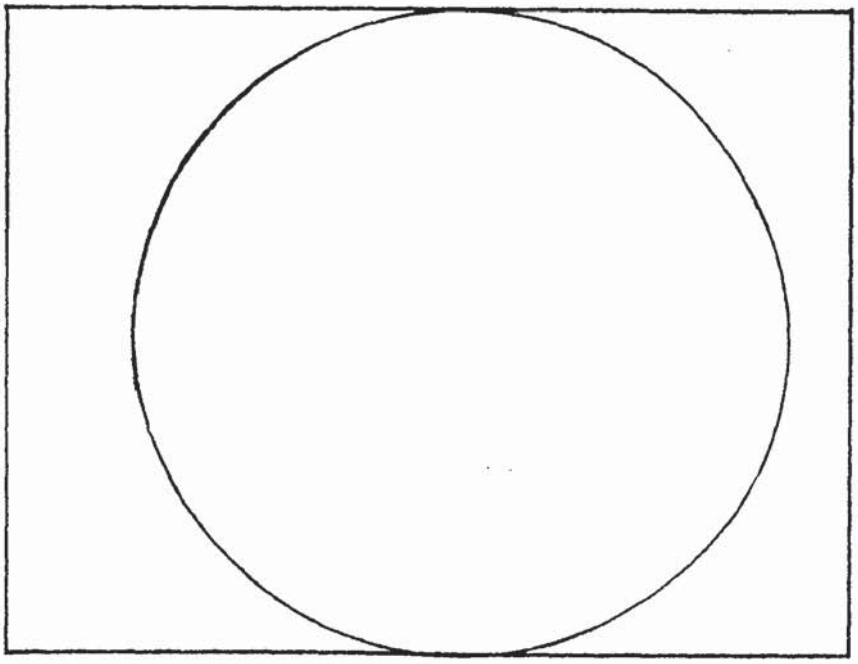
c) Locations 0E7B to 0EA9.

This section starts by sending the code "19" which clears the screen, it then sends out the code "1D" to force entry to the "Vector Mode".

Data is then sent out which will draw a three dimensional cube.

10.6.5. Typical Output.

Plane Rectangle and Circle.



Three Dimensional Cube.

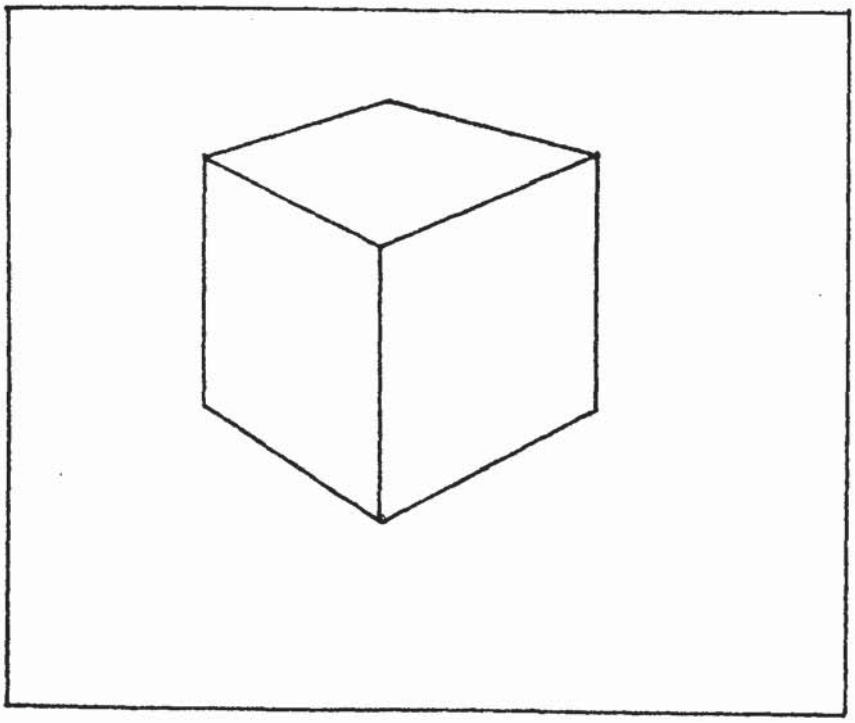


Fig. (10.9.)

Sample Data.

```
0E00   19 1D 23 64 23 58 34 74
0E08   23 58 34 74 3B 50 23 64        Plane Rectangle.
0E10   3B 50 23 64 23 58

0E16                     1D 31
0E18   70 2F 54 31 6A 31 43 30
0E20   78 32 4E 2F 7B 33 53 2E        Circle.
0E28   76 34 50 2D 6B 35 42 2B
0E30   7C 35 48 2A 6D 35 42 29
0E38   62 34 50 27 7D 33 53 27
0E40   60 32 4E 26 6E 31 43 26
0E48   68 2F 54 26 6E 2E 45 27
0E50   60 2C 5A 27 7D 2B 55 29
0E58   62 2A 58 2A 6D 2A 46 2B
0E60   7C 2A 40 2D 6B 2A 46 2E
0E68   76 2A 58 2F 7B 2B 55 30
0E70   78 2C 5A 31 6A 2E 45 31
0E78   70 2F 54

0E7B            19 1D 26 68 2C
0E80   5A 2D 62 2C 5A 2F 62 36        3 D  Cube.
0E88   40 29 62 36 40 26 68 2C
0E90   5A 29 62 22 5A 2E 72 22
0E98   5A 2D 62 2C 5A 2F 62 36
0EA0   40 2F 72 2E 40 2E 72 22
0EA8   5A 1C
```

10.6.6. Obtaining Vector Coordinates.

The screen size is assumed to be :-

X = 0 to X = 1024 Horizontal.

Y = 0 to Y = 780 Vertical.

Each vector point " X , Y " must be broken down into four hexadecimal coordinate data numbers:-

Y_{High} , Y_{Low} , X_{High} , X_{Low} .

Example:

Consider the coordinate points "X , Y = 123 , 633 ".
It is required to define the number system that will be used, before finding these points.

Denary numbers (To the base 10) will be denoted thus 123D.
Hexadecimal numbers (To the base 16) thus 2FH.

As an example:

The denary number 67 D , has the equivalent hexadecimal number of 43 H.

The following page gives a table of equivalents.

Note:
The largest X and Y values are $X_h = 3F$, $X_l = 5F$.
(In Hexadecimal) $Y_h = 38$, $Y_l = 6C$.

Table of Denary to Hexadecimal equivalents.

Denary.	Hexadecimal.	Denary.	Hexadecimal.
00	00	16	10
01	01	32	20
02	02	48	30
03	03	64	40
04	04	80	50
05	05	96	60
06	06	112	70
07	07	128	80
08	08	144	90
09	09	160	A0
10	0A	176	B0
11	0B	192	C0
12	0C	208	D0
13	0D	224	E0
14	0E	240	F0
15	0F	256	100
16	10	512	200
17	11	1024	400
18	12	4096	1000

a) To convert the "Y" coordinate of "633D". (Denary.)

$Y_{High} = \frac{633}{32} = 19$ D. (Integer part only)

$Y_{Low} = 633 - (19 \times 32) = 25$D. (Fractional part.)

Note:

Y_{High} can range from "00D to 31D" or from "20H to 3FH", hence a weighting factor must be added, because the ranges are not direct equivalents, the factor is "20H".

Continuing the example:

$Y_{High} = 19D = 13H$ and adding the weight of "20H" gives

$Y_{High} = 33H.$

Y_{Low} can range from "00D to 31D" or from "60H to 7FH", which requires a weighting factor of "60H" to be added, hence

$Y_{Low} = 25D = 19H$ and adding the weight of "60H" gives

$Y_{Low} = 79H.$

The "Y" coordinate of "633D" now transforms to :-

633 D = Y_{High} : 33H.

Y_{Low} : 79H.

The programme would be required to output the two hexadecimal numbers of "33H and 79H" to force the VDU to move to Y = 633D.

b) To convert the "X" coordinate "123" D.

$X_{High} = \frac{123}{32} = 3D$. (Integer part only.)

$X_{Low} = 123 - (3X\ 32) = 27$ D. (Fractional part.)

Note.

X_{High} can range from "00D to 31D" or from "20H to 3FH", hence as previously, the weighting factor is "20H".

X_{Low} can range from "00D to 31D" or from "40H to 5FH", with a weighting factor of "40H".

Continuing the example :-

X_{High} = 03D = 03H and adding the factor, X_{High} = 23H.

X_{Low} = 27D = 1BH and adding the factor, X_{Low} = 5BH.

The programme would be required to output the two hexadecimal numbers "23H and 5BH" to force the VDU to X = 123D.

The order of data transfer is very important, to force movement to :-

X, Y = 123D , 633D .

the data must be transferred in the order 33H , 79H , 23H , 5BH, ie with the "Y" coordinate first and the "X" coordinate last.

The coordinate transformation formulae are collected overleaf for easy reference.

10.6.7. Coordinate Transformation Formulae.

a) $Y_{High.D} = \left[32D + Int \left(\frac{Y_{coord}}{32D} \right) \right]$

b) $Y_{Low.D} = \left[96D + Fract \left(\frac{Y_{coord}}{32D} \right) \times 32D \right]$

c) $X_{High.D} = \left[32D + Int \left(\frac{X_{coord}}{32D} \right) \right]$

d) $X_{Low.D} = \left[64D + Fract \left(\frac{X_{coord}}{32D} \right) \times 32D \right]$

Each of the above coordinates must then be converted into a "Hexadecimal" number, as follows:-

e) $N_{Hex} = Int \left[\frac{N_D}{16D} \right] , \quad Fract \left[\frac{N_D}{16D} \right] .$

Example: 67D = 4 , 3 or

4 3 H.

10.7. Tool Contour Simulation Programme.

Appendix 4 gives a programme which simulates the trajectory of a machine tool in real time on a VDU. The programme contains all the routines necessary to process the incoming data from the tool interface. Because the interface has not yet been developed, the keyboard is used as the input device for demonstration purposes, finally the input would come from two A/D convertors connected to sensors on the actual tool, at the machine workface.

The display unit used was the RG 512 Lear Siegler ADM 3A visual display unit, with inputs from the Z80 keyboard processed by the microprocessor and sent out through the IM 6402 UART to the ADM 3A interface. The control programme waits for the operator to press any of the four keys :-

R for move to the right.

L for move to the left.

U for move upwards.

D for move downwards.

If another key is pressed the system cycles, waiting for a legal key entry. As the keys are pressed the display unit indicates a moving dot on the screen as directed.

10.8. Surface Generation.

If this control system is developed further as proposed then it should be possible to control movement in three dimensions, enabling machining of complicated surfaces to be performed. The hardware from the microprocessor is simply duplicated for each required axis, however the software necessary to control the hardware would increase greatly in complexity and some form of surface definition is required. The following section shows one method of defining the surface by the use of Bezier curves and the results were tested by forcing a printer to produce three dimensional representations of the required article.

10.8.1. The Bezier Surface. (13) (14)

The Bezier surface is represented in the form of a Cartesian product surface as shown in figure 10.10. In the Bezier formulation only the corner points are actually on the surface, these are denoted by A , B , C and D. The point $B_{1,2}$ defines the slope vector from the first to the second point on the first Bezier polygon in the U direction and the point $B_{2,1}$ defines the slope vector from the first to the second point on the first Bezier polygon in the W direction. The points $B_{1,3}$, $B_{2,4}$, $B_{3,4}$ $B_{4,3}$, $B_{4,2}$, $B_{3,1}$ are utilised in a similar manner. The points $B_{2,2}$, $B_{2,3}$, $B_{3,3}$ and $B_{3,2}$ interior to the defining polygonal surface define the twist vectors at A,B,C and D respectively.

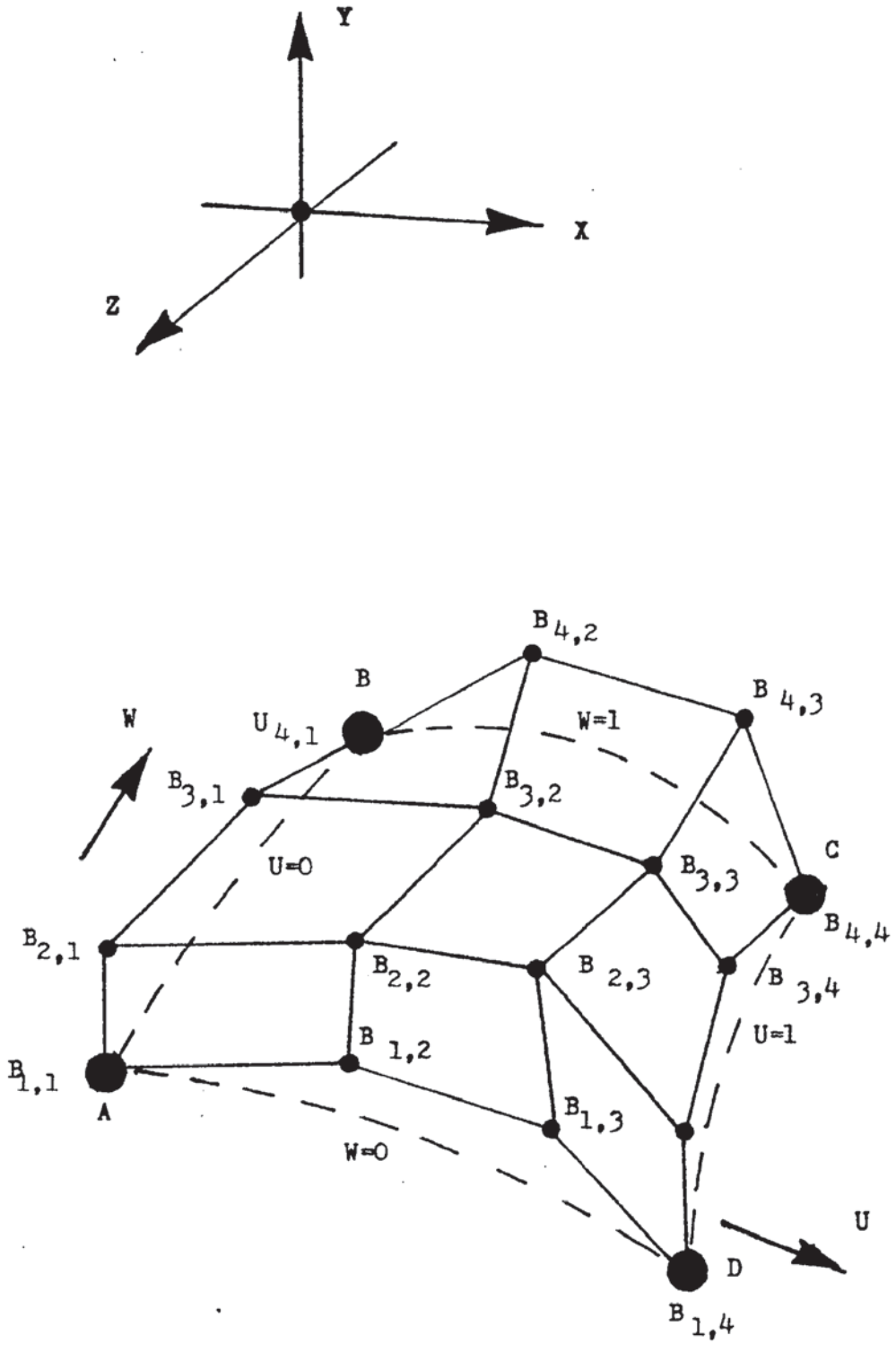


Fig. (10.10.) Bezier Surface.

The Bezier surface is completely defined by a net of design points on the polygonal surface and these points serve to define a two parameter family of Bezier curves on the design surface. To generate the parametric curves on the surface it is necessary to scan the surface by $0 \leqslant U \leqslant 1.0$ and $0 \leqslant W \leqslant 1.0$, the surface may then be represented in the form of a Cartesian product surface given by the equations :-

$$Q(U, W) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i+1,j+1} \cdot J_{n,i} \cdot U \cdot K_{m,j} \cdot W$$

Where $J_{n,i} = \binom{n}{i} . U^{i} . (1 - U)^{n-i}$

and $K_{m,j} = \binom{m}{j} . W^{j} . (1 - W)^{m-j}$

With m and n being one less than the number of polygon vertices in the W and U directions respectively. The term at the front of each equation is given by :-

$$\binom{n}{i} = \frac{n!}{i!\,(n-i)!}$$

$$\binom{m}{j} = \frac{m!}{j!\,(m-j)!}$$

The previous surface may be generated by feeding the coordinate data points through the RS 232 interface, or to the Port output of the PIO, to a three dimensional forming machine, the data beeing presented in ASCII format. A programme has been developed for this task and is given in Appendix 5, the procedure is demonstrated by sending the information to a printer in such a way that it appears to print in three dimensions. The administration routines may be written in a high level language such as BASIC , but it is essential that the machine control (or printer in this simulation) be controlled at machine code level, due to the speed requirement for the tool printhead control, in fact one of the simulations took 25 minutes to simulate one of the three dimensional objects in the BASIC language.

The programme given in Appendix 5 is ideal in this respect because machine code routines are automatically entered into RAM at steps 1220 to 1440, by the administration programme and called automatically when speed is required, such as at steps 605, 1280, 1320, 1340. This unique trick gives an ideal match between high and low level languages. The machine code data is stored at steps 1500 to 1880 at high level and at locations BE00 to BF70 hexadecimal at low level. Two examples follow of the printout with further examples in Appendix 5.

10.8.2. Example Printouts.

```
32 PRINT"32 Inp Cond":N=2:M=2:P=7
Ok
LIST3000

3000 DATA 150,110,0,150,200,250,150,290,0
3010 DATA 100,110,0,100,200,250,100,290,0
3020 DATA  50,110,0, 50,200,250, 50,290,0
Ok
```

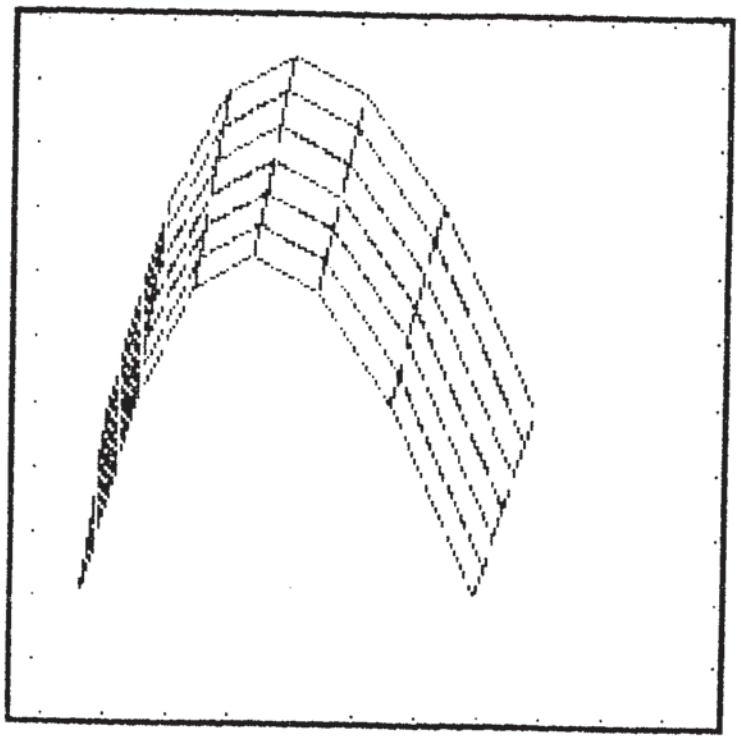


Fig. (10.11.)

Three Dimensional Printout.

```
32 PRINT"32 Inp Cond":N=3:M=3:P=7
Ok
LIST3000

3000 DATA 120,120,  0,120,120,120,120,  0,120
3010 DATA 120,  0,  0, 80, 80, 40, 80, 80, 80
3020 DATA  80, 40, 80, 80, 40, 40, 40, 80, 40
3030 DATA  40, 80, 80, 40, 40, 80, 40, 40, 40
3040 DATA   0,120,  0,  0,120,120,  0,  0,120
3050 DATA   0,  0,  0
Ok
```

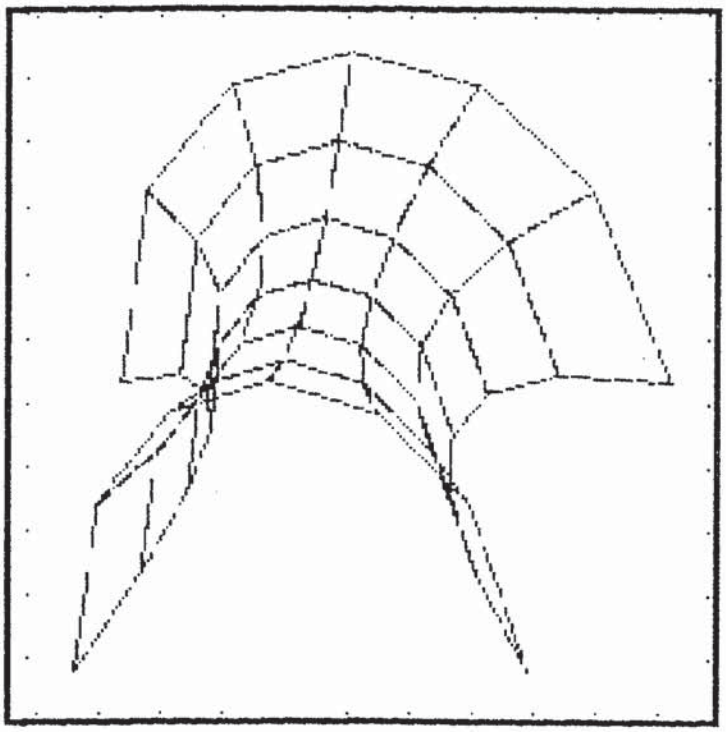


Fig. (10.12.)

10.9. Additional Peripherals.

If the system were expanded as proposed in the section on future developement , then it would be possible to add further control facilities to the Ward Capstan Lathe , in fact a second small microprocessor system could be utilised to control the turret block , or lead screws could be incorporated to control the slide movements , enabling complicated shapes to be machined on the same lathe.

The chapter includes a design for a stepper motor control system , via a microprocessor and a section on the use of opto-isolators , to improve the isolation of the controlling device from noise pulses , obtained by operating high current components on the lathe.

A further possibility is to use the relatively new fibre optic devices to transmit the information to a video output device . The fibre optic theory is discussed and a typical test circuit is included so that an idea can be gained , of how to proceed in this direction.

10.10. The Stepper Motor.

As it stands, the slide movement on the Ward lathe cannot be controlled directly by a microprocessor unless a stepper motor was fitted to a lead screw. **Figure 10.13 shows a** Mullard SAA 1027 (15) stepper motor drive integrated circuit chip which is intended to drive a four phase two stator stepper motor.

The circuit consists of four output stages , a logic stage and three input stages, the logic stage is driven by three inputs :-

1) A trigger input stage.
2) A stage that can change the sequence of the logic to give clockwise or anticlockwise rotation.
3) A set input stage which sets the four outputs.

The three inputs are compatible with high noise immunity logic to ensure that the circuit operates correctly even in a noisy environment . The output is capable of delivering 350 mA of output current to each of the four phases of the stepper motor.

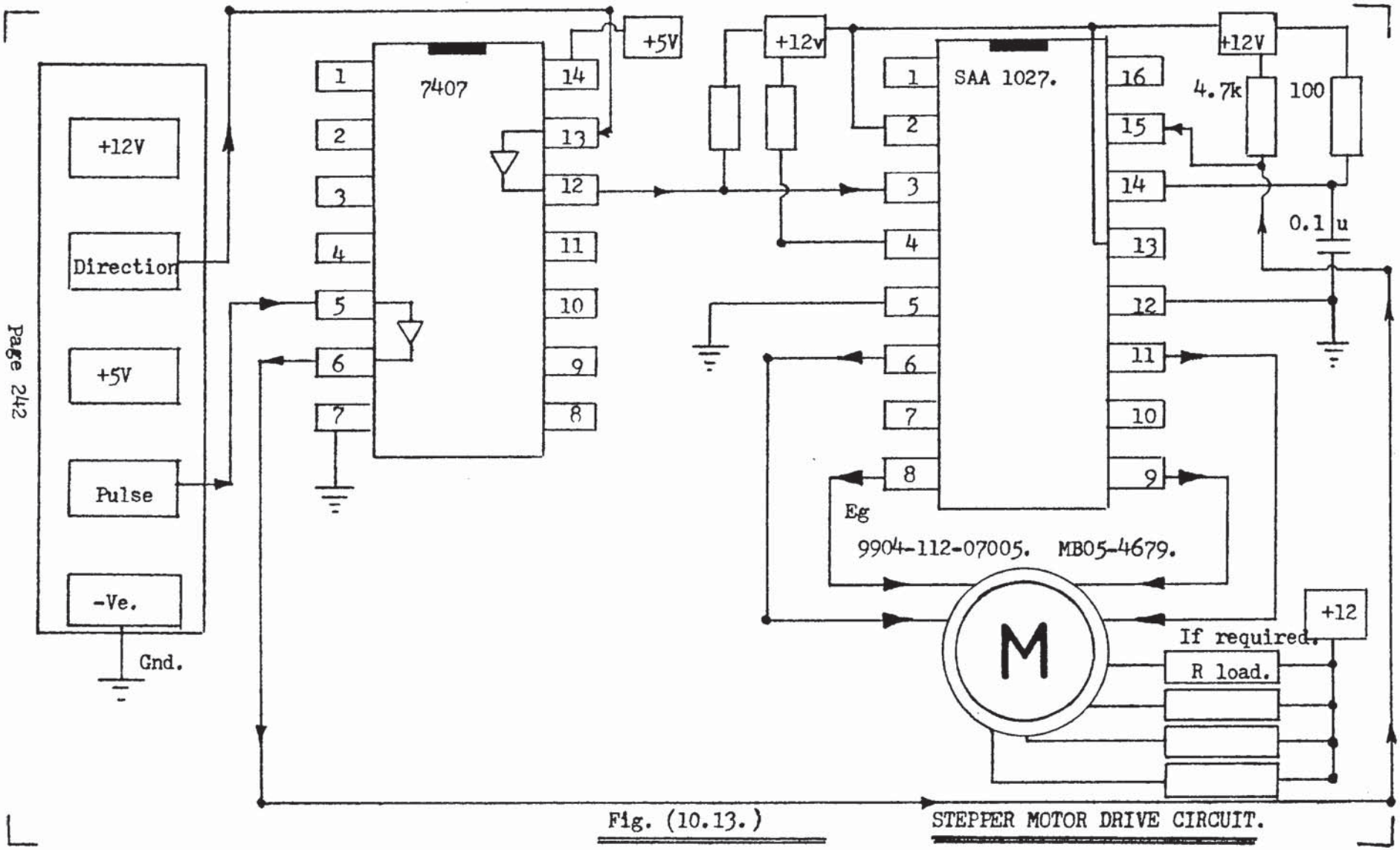


Fig. (10.13.) STEPPER MOTOR DRIVE CIRCUIT.

The SAA 1027 stepper motor drive chip requires two signals from the microprocessor :-

a) A direction bit, fed to pin 3, which sets the direction of rotation. The output is either "00" or "01" and need only be a pulse, because the signal will toggle a latch.

b) A pulsed voltage fed to pin 15, the mark to space ratio of the pulse must be adjusted to vary the rotational speed of the stepper motor. Each pulse will step the motor forward or back a specific angle, depending on the motor characteristic, in general 7.5^{o}.

The 7407 integrated circuit chip shown in the diagram utilises the buffer capability of the chip to act as a simple buffer, between the CPU and the motor driving chip.

Appendix 6 gives example programmes to demonstrate how the proposed design could be controlled, by the microprocessor. It is assumed that the control pulses are sent out in the following standard :-

a) Direction pulse to Port A bit 0.

b) Stepping pulse to Port B bit 0.

10.11. Opto Isolators.

When operating a microprocessor in workshop or factory conditions, noise pulses from heavy current switching may cause problems, causing random or spurious signals to be generated , within the processor.

Noise source isolation is possible by utilising opto isolating devices. These allow signal transfer from one device to another, by means of an infra red beam of light within a sealed package.

The only connection is via the light beam and hence noise isolation is very high, however it is important that no connection whatsoever is made on the power line or common line between the two devices.

Infra red light is used because it is not affected by the ambient light.

The following circuit shows a typical design indicating how the isolation is obtained practically. by utilising light as the communication media, see figure 10.14.

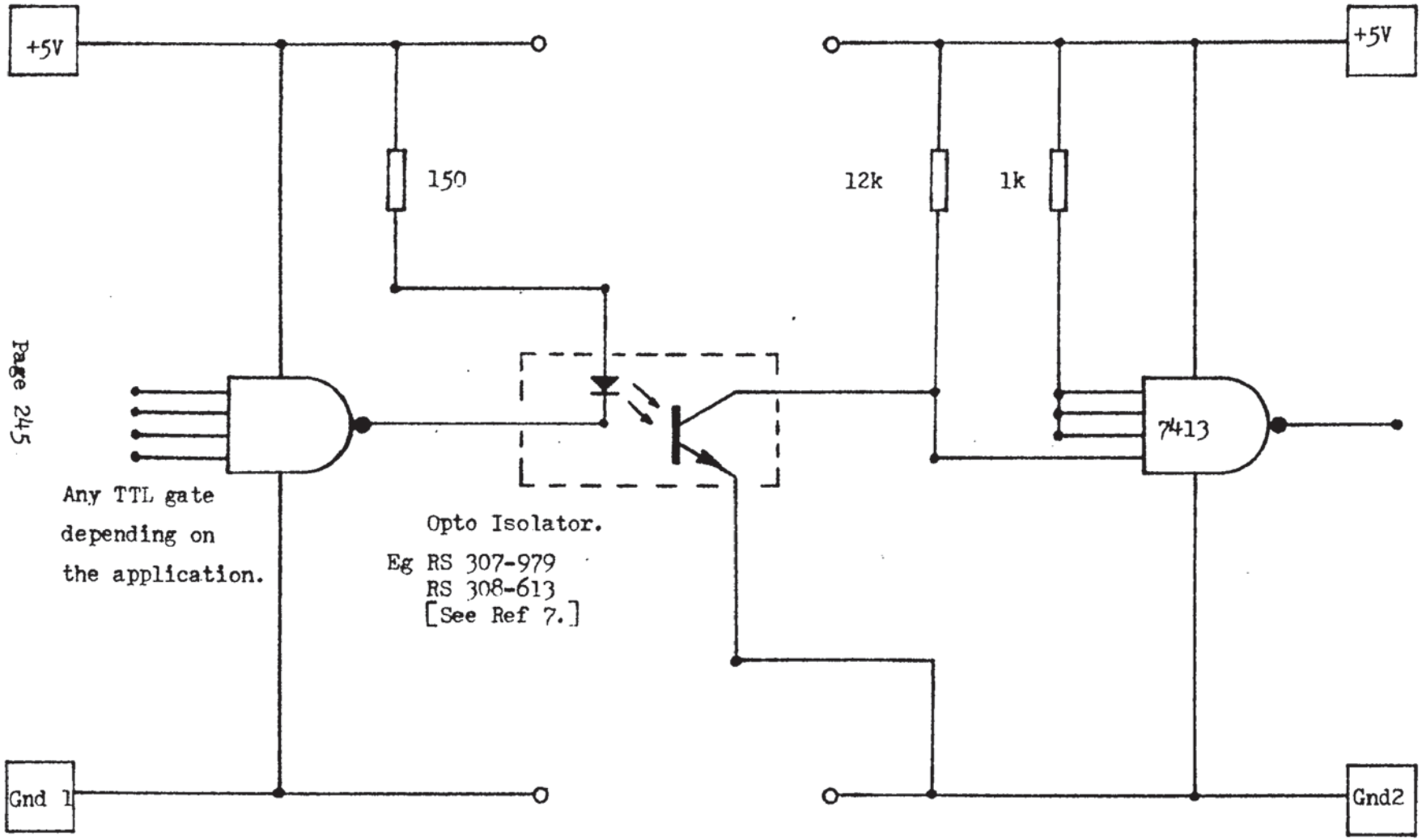


Fig. (10.14.) TYPICAL OPTO ISOLATOR.

If the rays pass from a dense medium $[N_1]$ to a less dense medium $[N_2]$, then at a certain angle θ_1 the resulting angle θ_2 could be 90^o, the critical angle θ_1 required to give this unique result is called θ_c and "Snell's Law" gives the relationship :-

$$\text{Sin } \theta_c = \frac{N_2}{N_1}$$

.....

If a fibre optic cable is used, then the rays rebound along the length of the fibre and exit only at the far end, due to internal reflection.

There is a maximum angle θ_a at which the incident ray can enter the core and experience total internal reflection, θ_a is called the "Acceptance Angle" and Sin θ_a is known as the "Numerical Aperture", hence by "Snell's Law" :-

$$\text{Numerical Aperture} = \text{Sin } \theta_a$$

$$N_o \cdot \text{Sin } \theta_a = \text{Sin } [\, 90^o - \theta_c \,] \cdot N_1$$

$$= N_1 \cdot \text{Cos } \theta_c$$

$$= N_1 \cdot [\, 1 - \text{Sin}^2 \theta_c \,]^{1/2}$$

Where N_o is the index of refraction of the external medium.

N_1 is the refraction index of the core.

N_2 is the refraction index of the outer cladding.

Sin θ_a is the numerical aperture.

If the external medium is air then $N_o = 1$.

Since $\text{Sin } \theta_c = N_2 / N_1$ then

$$N_o \cdot \text{Sin } \theta_a = N_1 \cdot [\, 1 - (N_2 / N_1)^2 \,]^{1/2}$$

$$\text{Sin } \theta_a = \frac{N_1}{N_o} \cdot [\, \frac{N_1^2 - N_2^2}{N_1^2} \,]^{1/2}$$

$$\text{Sin } \theta_a = \frac{1}{N_o} \cdot [\, N_1^2 - N_2^2 \,]^{1/2}$$

and substituting the value of $N_o = 1$ for air gives :-

$$\text{Numerical Aperture} = \text{Sin } \theta_a = [\, N_1^2 - N_2^2 \,]^{1/2}$$

The light rays in a fibre may be classified as meridional or skew. Meridional rays are those that pass through the axis of a fibre after each rebound, while skew rays never intersect the fibre axis. There are also parallel rays which travel down the fibre, never being reflected. Basic fibre optic theory is concerned with meridional rays which fall into two categories, low order modes and high order modes.

Low order modes are rays that are launched at small angles, within the acceptance angle, while high order modes occur when the rays are launched at large angles. Single mode fibres result when the core area and the numerical aperture are so small that only one mode can propagate.

Two types of fibre construction are commonly used, a step index fibre which consists of a cylindrical core of glass, silica or plastic, of refractive index N , covered by a thin cladding of a lower refractive index N_2. The second type is a graded fibre whose refractive index changes gradually from a high order index at the centre, to a low index at the perimeter.

Fibres encounter two types of light dispersion, which limits achievable bandwidths. Material dispersion results from the fact that different wavelengths travel at different velocities in the same medium, consequently the various wavelengths launched simultaneously within the flux, will not arrive at the receiver together but will suffer time dispersion, due to differences in travel time. The effect can be reduced by using an emitter with a narrow spectrum of emission, such as a laser. (17)

Modal dispersion is caused by the difference in path lengths between low order modes and high order modes . The high orders have a longer travel time and simultaneously launched rays will suffer dispersion on arrival . Modal dispersion can be reduced in step index fibres by decreasing the numerical aperture, to allow only the lower modes to propagate . There are four main causes of transmission loss in optical fibres.

1) Material absorption caused by molecular impurities within the core of the fibre , which absorb certain wavelengths . The problem can be limited by choosing a suitable emitter whose peak wavelength corresponds approximately to the spectral region of maximum transmission of the fibre.

2) Material scattering is caused by particle impurities and by fluctuations in temperature and composition.

3) Further scattering is caused by irregularities at the core to cladding interface , which results in transmission into the cladding and subsequent loss of energy on reflection.

4) It is also possible for the curvature of the fibre to cause some transmission loss.

If the curvature is too great,then some rays will strike the boundary at angles less than the critical angle and partial transmission, into the cladding will occur,with the resultant loss of reflection.

Coupling losses must be considered, when designing a fibre optic link, the three main loss mechanisms are :-

a) Fibre to fibre. (Inline.)

b) Fibre to fibre. (Bulkhead.)

c) Fibre to emitter detector unit.

The loss is due to a number of factors,but in particular, reflections at the mating faces, slight misalignment due to manufacturing tolerances and separation between the mating surfaces. (18) (19)

If the reader is inexperienced in the use of fibre optics then the author suggests a simple test circuit , which could be built to gain some practical experience on the subject.

The following circuit shows how a simple fibre optic link may be used to transmit an audio signal, such as the input from a microphone or pickup stylus and then fed into an audio amplifier.

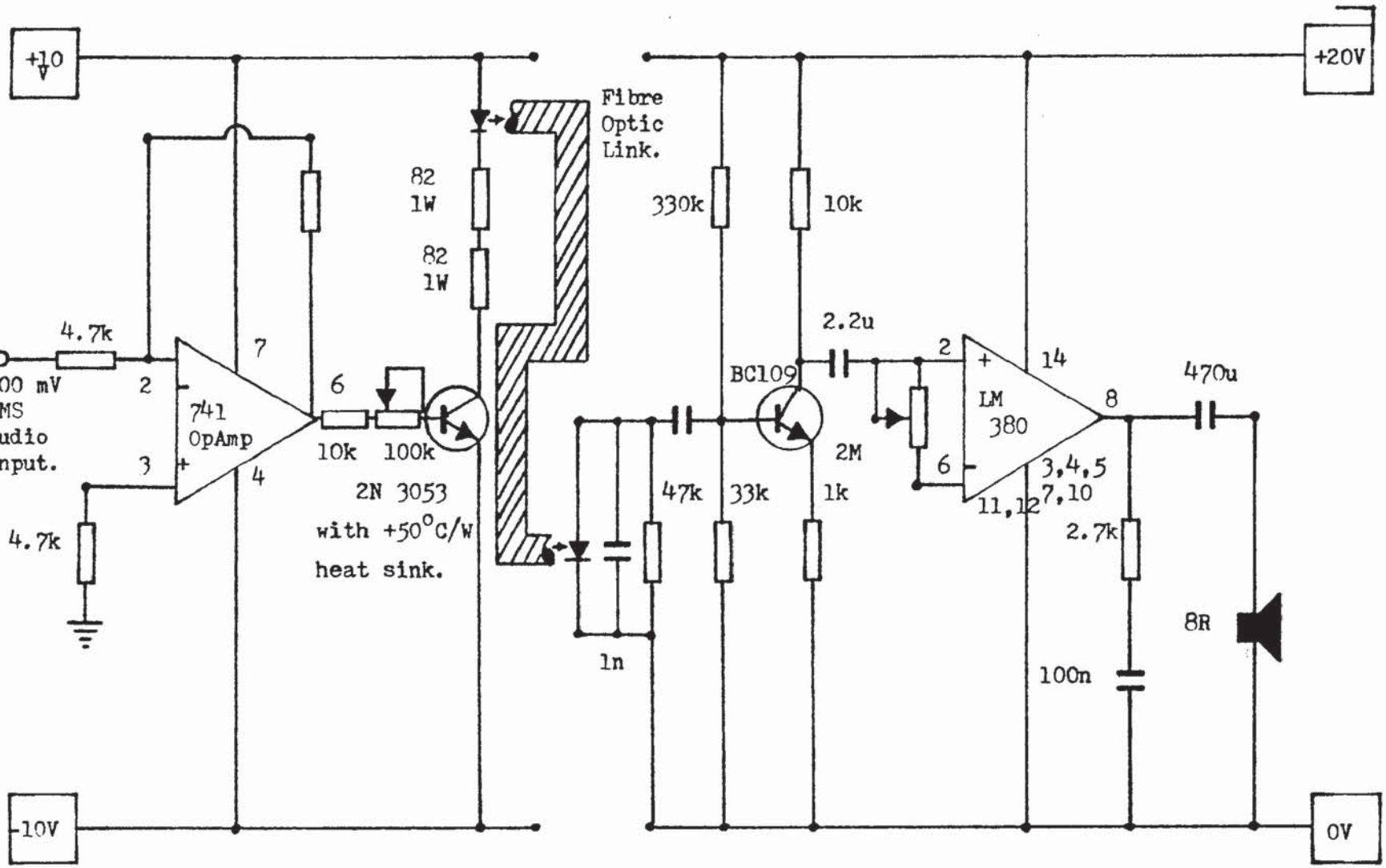


Fig. (10.15.) EXAMPLE FIBRE OPTIC LINK.

APPENDICES.

Appendix 1. Z80 CPU INSTRUCTION SET.

The following is a list of the Op-codes and Mnemonics used on the Z80 microprocessor system.

All the codes are standard and are reproduced by kind permission of Zilog Ltd.

The system as it stands must be operated by using the OBJ CODE column, however if the memory expansion is fitted then it would be possible to write in Assembly or the Source Statement column.

Z80–CPU INSTRUCTION SET

OBJ CODE	SOURCE STATEMENT		OPERATION
8E	ADC	A,(HL)	Add with Carry Operand to Acc.
DD8E05	ADC	A,(IX+d)	
FD8E05	ADC	A,(IY+d)	
8F	ADC	A,A	
88	ADC	A,B	
89	ADC	A,C	
8A	ADC	A,D	
8B	ADC	A,E	
8C	ADC	A,H	
8D	ADC	A,L	
CE20	ADC	A,n	
ED4A	ADC	HL,BC	Add with Carry Reg. Pair to HL
ED5A	ADC	HL,DE	
ED6A	ADC	HL,HL	
ED7A	ADC	HL,SP	
86	ADD	A,(HL)	Add Operand to Acc.
DD8605	ADD	A,(IX+d)	
FD8605	ADD	A,(IY+d)	
87	ADD	A,A	
80	ADD	A,B	
81	ADD	A,C	
82	ADD	A,D	
83	ADD	A,E	
84	ADD	A,H	
85	ADD	A,L	
C620	ADD	A,n	
09	ADD	HL,BC	Add Reg. Pair to HL
19	ADD	HL,DE	
29	ADD	HL,HL	
39	ADD	HL,SP	
DD09	ADD	IX,BC	Add Reg. Pair to IX
DD19	ADD	IX,DE	
DD29	ADD	IX,IX	
DD39	ADD	IX,SP	
FD09	ADD	IY,BC	Add Reg. Pair to IY
FD19	ADD	IY,DE	
FD29	ADD	IY,IY	
FD39	ADD	IY,SP	
A6	AND	(HL)	Logical 'AND' of Operand and Acc.
DDA605	AND	(IX+d)	
FDA605	AND	(IY+d)	
A7	AND	A	
A0	AND	B	
A1	AND	C	
A2	AND	D	
A3	AND	E	
A4	AND	H	
A5	AND	L	
E620	AND	n	
CB46	BIT	0,(HL)	Test Bit b of Location or Reg.
DDCB0546	BIT	0,(IX+d)	
FDCB0546	BIT	0,(IY+d)	
CB47	BIT	0,A	
CB40	BIT	0,B	
CB41	BIT	0,C	
CB42	BIT	0,D	
CB43	BIT	0,E	
CB44	BIT	0,H	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB45	BIT	0,L	Test Bit b of Location
CB4E	BIT	1,(HL)	or Reg.
DDCB054E	BIT	1,(IX+d)	
FDCB054E	BIT	1,(IY+d)	
CB4F	BIT	1,A	
CB48	BIT	1,B	
CB49	BIT	1,C	
CB4A	BIT	1,D	
CB4B	BIT	1,E	
CB4C	BIT	1,H	
CB4D	BIT	1,L	
CB56	BIT	2,(HL)	
DDCB0556	BIT	2,(IX+d)	
FDCB0556	BIT	2,(IY+d)	
CB57	BIT	2,A	
CB50	BIT	2,B	
CB51	BIT	2,C	
CB52	BIT	2,D	
CB53	BIT	2,E	
CB54	BIT	2,H	
CB55	BIT	2,L	
CB5E	BIT	3,(HL)	
DDCB055E	BIT	3,(IX+d)	
FDCB055E	BIT	3,(IY+d)	
CB5F	BIT	3,A	
CB58	BIT	3,B	
CB59	BIT	3,C	
CB5A	BIT	3,D	
CB5B	BIT	3,E	
CB5C	BIT	3,H	
CB5D	BIT	3,L	
CB66	BIT	4,(HL)	
DDCB0566	BIT	4,(IX+d)	
FDCB0566	BIT	4,(IY+d)	
CB67	BIT	4,A	
CB60	BIT	4,B	
CB61	BIT	4,C	
CB62	BIT	4,D	
CB63	BIT	4,E	
CB64	BIT	4,H	
CB65	BIT	4,L	
CB6E	BIT	5,(HL)	
DDCB056E	BIT	5,(IX+d)	
FDCB056E	BIT	5,(IY+d)	
CB6F	BIT	5,A	
CB68	BIT	5,B	
CB69	BIT	5,C	
CB6A	BIT	5,D	
CB6B	BIT	5,E	
CB6C	BIT	5,H	
CB6D	BIT	5,L	
CB76	BIT	6,(HL)	
DDCB0576	BIT	6,(IX+d)	
FDCB0576	BIT	6,(IY+d)	
CB77	BIT	6,A	
CB70	BIT	6,B	
CB71	BIT	6,C	
CB72	BIT	6,D	
CB73	BIT	6,E	
CB74	BIT	6,H	
CB75	BIT	6,L	
CB7E	BIT	7,(HL)	
DDCB057E	BIT	7,(IX+d)	

OBJ CODE	SOURCE STATEMENT		OPERATION
FDCB057E	BIT	7,(IY+d)	Test Bit b Location or Reg.
CB7F	BIT	7,A	
CB78	BIT	7,B	
CB79	BIT	7,C	
CB7A	BIT	7,D	
CB7B	BIT	7,E	
CB7C	BIT	7,H	
CB7D	BIT	7,L	
DC8405	CALL	C,nn	Call Subroutine at Location nn if Condition True
FC8405	CALL	M,nn	
D48405	CALL	NC,nn	
C48405	CALL	NZ,nn	
F48405	CALL	P,nn	
EC8405	CALL	PE,nn	
E48405	CALL	PO,nn	
CC8405	CALL	Z,nn	
CD8405	CALL	nn	Unconditional Call to Subroutine at nn
3F	CCF		Complement Carry Flag
BE	CP	(HL)	Compare Operand with Acc.
DDBE05	CP	(IX+d)	
FDBE05	CP	(IY+d)	
BF	CP	A	
B8	CP	B	
B9	CP	C	
BA	CP	D	
BB	CP	E	
BC	CP	H	
BD	CP	L	
FE20	CP	n	
EDA9	CPD		Compare Location (HL) and Acc. Decrement HL and BC
EDB9	CPDR		Compare Location (HL) and Acc. Decrement HL and BC, Repeat until BC = 0
EDA1	CPI		Compare Location (HL) and Acc., Increment HL and Decrement BC
EDB1	CPIR		Compare Location (HL) and Acc. Increment HL, Decrement BC, Repeat until BC = 0
2F	CPL		Complement Acc. (1's Comp)
27	DAA		Decimal Adjust Acc.
35	DEC	(HL)	Decrement Operand
DD3505	DEC	(IX+d)	
FD3505	DEC	(IY+d)	
3D	DEC	A	
05	DEC	B	
0B	DEC	BC	
0D	DEC	C	
15	DEC	D	
1B	DEC	DE	

OBJ CODE	SOURCE STATEMENT		OPERATION
1D	DEC	E	Decrement Operand
25	DEC	H	
2B	DEC	HL	
DD2B	DEC	IX	
FD2B	DEC	IY	
2D	DEC	L	
3B	DEC	SP	
F3	DI		Disable Interrupts
102E	DJNZ	e	Decrement B and Jump Relative if B = 0
FB	EI		Enable Interrupts
E3	EX	(SP),HL	Exchange Location and (SP)
DDE3	EX	(SP),IX	
FDE3	EX	(SP),IY	
08	EX	AF,AF'	Exchange the Contents of AF and AF'
EB	EX	DE,HL	Exchange the Contents of DE and HL
D9	EXX		Exchange the Contents of BC,DE,HL with Contents of BC',DE',HL' Respectively
76	HALT		HALT (Wait for Interrupt or Reset)
ED46	IM	0	Set Interrupt Mode
ED56	IM	1	
ED5E	IM	2	
ED78	IN	A,(C)	Load Reg. with Input from Device (C)
ED40	IN	B,(C)	
ED48	IN	C,(C)	
ED50	IN	D,(C)	
ED58	IN	E,(C)	
ED60	IN	H,(C)	
ED68	IN	L,(C)	
34	INC	(HL)	Increment Operand
DD3405	INC	(IX+d)	
FD3405	INC	(IY+d)	
3C	INC	A	
04	INC	B	
03	INC	BC	
0C	INC	C	
14	INC	D	
13	INC	DE	
1C	INC	E	
24	INC	H	
23	INC	HL	
DD23	INC	IX	
FD23	INC	IY	
2C	INC	L	
33	INC	SP	
DB20	IN	A,(n)	Load Acc. with Input from Device n
EDAA	IND		Load Location (HL) with Input from Port (C), Decrement HL and B

OBJ CODE	SOURCE STATEMENT		OPERATION
EDBA	INDR		Load Location (HL) with Input from Port (C), Decrement HL and Decrement B, Repeat until B = 0
EDA2	INI		Load Location (HL) with Input from Port (C); Increment HL and Decrement B
EDB2	INIR		Load Location (HL) with Input from Port (C), Increment HL and Decrement B, Repeat until B = 0
C38405	JP	nn	Unconditional Jump to Location
E9	JP	(HL)	
DDE9	JP	(IX)	
FDE9	JP	(IY)	
DA8405	JP	C,nn	Jump to Location if Condition True
FA8405	JP	M,nn	
D28405	JP	NC,nn	
C28405	JP	NZ,nn	
F28405	JP	P,nn	
EA8405	JP	PE,nn	
E28405	JP	PO,nn	
CA8405	JP	Z,nn	
382E	JR	C,e	Jump Relative to PC+e if Condition True
302E	JR	NC,e	
202E	JR	NZ,e	
282E	JR	Z,e	
182E	JR	e	Unconditional Jump Relative to PC+e
02	LD	(BC),A	Load Source to Destination
12	LD	(DE),A	
77	LD	(HL),A	
70	LD	(HL),B	
71	LD	(HL),C	
72	LD	(HL),D	
73	LD	(HL),E	
74	LD	(HL),H	
75	LD	(HL),L	
3620	LD	(HL),n	
DD7705	LD	(IX+d),A	
DD7005	LD	(IX+d),B	
DD7105	LD	(IX+d),C	
DD7205	LD	(IX+d),D	
DD7305	LD	(IX+d),E	
DD7405	LD	(IX+d),H	
DD7505	LD	(IX+d),L	
DD360520	LD	(IX+d),n	
FD7705	LD	(IY+d),A	
FD7005	LD	(IY+d),B	
FD7105	LD	(IY+d),C	
FD7205	LD	(IY+d),D	
FD7305	LD	(IY+d),E	
FD7405	LD	(IY+d),H	
FD7505	LD	(IY+d),L	
FD360520	LD	(IY+d),n	
328405	LD	(nn),A	
ED438405	LD	(nn),BC	

OBJ CODE	SOURCE STATEMENT		OPERATION
ED538405	LD	(nn),DE	Load Source to Des-
228405	LD	(nn),HL	tination
DD228405	LD	(nn),IX	
FD228405	LD	(nn),IY	
ED738405	LD	(nn),SP	
0A	LD	A,(BC)	
1A	LD	A,(DE)	
7E	LD	A,(HL)	
DD7E05	LD	A,(IX+d)	
FD7E05	LD	A,(IY+d)	
3A8405	LD	A,(nn)	
7F	LD	A,A	
78	LD	A,B	
79	LD	A,C	
7A	LD	A,D	
7B	LD	A,E	
7C	LD	A,H	
ED57	LD	A,I	
7D	LD	A,L	
3E20	LD	A,n	
ED5F	LD	A,R	
46	LD	B,(HL)	
DD4605	LD	B,(IX+d)	
FD4605	LD	B,(IY+d)	
47	LD	B,A	
40	LD	B,B	
41	LD	B,C	
42	LD	B,D	
43	LD	B,E	
44	LD	B,H	
45	LD	B,L	
0620	LD	B,n	
ED4B8405	LD	BC,(nn)	
018405	LD	BC,nn	
4E	LD	C,(HL)	
DD4E05	LD	C,(IX+d)	
FD4E05	LD	C,(IY+d)	
4F	LD	C,A	
48	LD	C,B	
49	LD	C,C	
4A	LD	C,D	
4B	LD	C,E	
4C	LD	C,H	
4D	LD	C,L	
0E20	LD	C,n	
56	LD	D,(HL)	
DD5605	LD	D,(IX+d)	
FD5605	LD	D,(IY+d)	
57	LD	D,A	
50	LD	D,B	
51	LD	D,C	
52	LD	D,D	
53	LD	D,E	
54	LD	D,H	
55	LD	D,L	
1620	LD	D,n	
ED588405	LD	DE,(nn)	
118405	LD	DE,nn	
5E	LD	E,(HL)	
DD5E05	LD	E,(IX+d)	
FD5E05	LD	E,(IY+d)	
5F	LD	E,A	
58	LD	E,B	
59	LD	E,C	

OBJ CODE	SOURCE STATEMENT		OPERATION
5A	LD	E,D	Load Source to Destination
5B	LD	E,E	
5C	LD	E,H	
5D	LD	E,L	
1E20	LD	E,n	
66	LD	H,(HL)	
DD6605	LD	H,(IX+d)	
FD6605	LD	H,(IY+d)	
67	LD	H,A	
60	LD	H,B	
61	LD	H,C	
62	LD	H,D	
63	LD	H,E	
64	LD	H,H	
65	LD	H,L	
2620	LD	H,n	
2A8405	LD	HL,(nn)	
218405	LD	HL,nn	
ED47	LD	I,A	
DD2A8405	LD	IX,(nn)	
DD218405	LD	IX,nn	
FD2A8405	LD	IY,(nn)	
FD218405	LD	IY,nn	
6E	LD	L,(HL)	
DD6E05	LD	L,(IX+d)	
FD6E05	LD	L,(IY+d)	
6F	LD	L,A	
68	LD	L,B	
69	LD	L,C	
6A	LD	L,D	
6B	LD	L,E	
6C	LD	L,H	
6D	LD	L,L	
2E20	LD	L,n	
ED4F	LD	R,A	
ED7B8405	LD	SP,(nn)	
F9	LD	SP,HL	
DDF9	LD	SP,IX	
FDF9	LD	SP,IY	
318405	LD	SP,nn	
EDA8	LDD		Load Location (DE) with Location (HL), Decrement DE,HL and BC
EDB8	LDDR		Load Location (DE) with Location (HL), Repeat until BC = 0
EDA0	LDI		Load Location (DE) with Location (HL), Increment DE,HL, Decrement BC
EDB0	LDIR		Load Location (DE) with Location (HL), Increment DE,HL, Decrement BC and Repeat until BC = 0
ED44	NEG		Negate Acc. (2's Complement)
00	NOP		No Operation
B6	OR	(HL)	Logical "OR" of Operand and Acc.
DDB605	OR	(IX+d)	

OBJ CODE	SOURCE STATEMENT		OPERATION
FDB605	OR	(IY+d)	Logical "OR" of Operand and Acc.
B7	OR	A	
B0	OR	B	
B1	OR	C	
B2	OR	D	
B3	OR	E	
B4	OR	H	
B5	OR	L	
F620	OR	n	
ED8B	OTDR		Load Output Port (C) with Location (HL) Decrement HL and B, Repeat until B = 0
EDB3	OTIR		Load Output Port (C) with Location (HL), Increment HL, Decrement B, Repeat until B = 0
ED79	OUT	(C),A	Load Output Port (C) with Reg.
ED41	OUT	(C),B	
ED49	OUT	(C),C	
ED51	OUT	(C),D	
ED59	OUT	(C),E	
ED61	OUT	(C),H	
ED69	OUT	(C),L	
D320	OUT	(n),A	Load Output Port (n) with Acc.
EDAB	OUTD		Load Output Port (C) with Location (HL), Decrement HL and B
EDA3	OUTI		Load Output Port (C) with Location (HL), Increment HL and Decrement B
F1	POP	AF	Load Destination with Top of Stack
C1	POP	BC	
D1	POP	DE	
E1	POP	HL	
DDE1	POP	IX	
FDE1	POP	IY	
F5	PUSH	AF	Load Source to Stack
C5	PUSH	BC	
D5	PUSH	DE	
E5	PUSH	HL	
DDE5	PUSH	IX	
FDE5	PUSH	IY	
CB86	RES	0,(HL)	Reset Bit b of Operand
DDCB0586	RES	0,(IX+d)	
FDCB0586	RES	0,(IY+d)	
CB87	RES	0,A	
CB80	RES	0,B	
CB81	RES	0,C	
CB82	RES	0,D	
CB83	RES	0,E	
CB84	RES	0,H	
CB85	RES	0,L	
CB8E	RES	1,(HL)	
DDCB058E	RES	1,(IX+d)	
FDCB058E	RES	1,(IY+d)	
CB8F	RES	1,A	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB88	RES	1,B	Reset Bit b of Operand
CB89	RES	1,C	
CB8A	RES	1,D	
CB8B	RES	1,E	
CB8C	RES	1,H	
CB8D	RES	1,L	
CB96	RES	2,(HL)	
DDCB0596	RES	2,(IX+d)	
FDCB0596	RES	2,(IY+d)	
CB97	RES	2,A	
CB90	RES	2,B	
CB91	RES	2,C	
CB92	RES	2,D	
CB93	RES	2,E	
CB94	RES	2,H	
CB95	RES	2,L	
CB9E	RES	3,(HL)	
DDCB059E	RES	3,(IX+d)	
FDCB059E	RES	3,(IY+d)	
CB9F	RES	3,A	
CB98	RES	3,B	
CB99	RES	3,C	
CB9A	RES	3,D	
CB9B	RES	3,E	
CB9C	RES	3,H	
CB9D	RES	3,L	
CBA6	RES	4,(HL)	
DDCB05A6	RES	4,(IX+d)	
FDCB05A6	RES	4,(IY+d)	
CBA7	RES	4,A	
CBA0	RES	4,B	
CBA1	RES	4,C	
CBA2	RES	4,D	
DBA3	RES	4,E	
CBA4	RES	4,H	
CBA5	RES	4,L	
CBAE	RES	5,(HL)	
DDCB05AE	RES	5,(IX+d)	
FDCB05AE	RES	5,(IY+d)	
CBAF	RES	5,A	
CBA8	RES	5,B	
CBA9	RES	5,C	
CBAA	RES	5,D	
CBAB	RES	5,E	
CBAC	RES	5,H	
CBAD	RES	5,L	
CBB6	RES	6,(HL)	
DDCB0586	RES	6,(IX+d)	
FDCB05B6	RES	6,(IY+d)	
CBB7	RES	6,A	
CBB0	RES	6,B	
CBB1	RES	6,C	
CBB2	RES	6,D	
CBB3	RES	6,E	
CBB4	RES	6,H	
CBB5	RES	6,L	
CBBE	RES	7,(HL)	
DDCB05BE	RES	7,(IX+d)	
FDCB05BE	RES	7,(IY+d)	
CBBF	RES	7,A	
CBB8	RES	7,B	
CBB9	RES	7,C	
CBBA	RES	7,D	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB88	RES	7,E	Reset Bit b of Operand
CB8C	RES	7,H	
CB8D	RES	7,L	
C9	RET		Return from Subroutine
D8	RET	C	Return from Subroutine if Condition True
F8	RET	M	
D0	RET	NC	
C0	RET	NZ	
F0	RET	P	
E8	RET	PE	
E0	RET	PO	
C8	RET	Z	
ED4D	RETI		Return from Interrupt
ED45	RETN		Return from Non-Maskable Interrupt
CB16	RL	(HL)	Rotate Left Through Carry
DDCB0516	RL	(IX+d)	
FDCB0516	RL	(IY+d)	
CB17	RL	A	
CB10	RL	B	
CB11	RL	C	
CB12	RL	D	
CB13	RL	E	
CB14	RL	H	
CB15	RL	L	
17	RLA		Rotate Left Acc. Through Carry
CB06	RLC	(HL)	Rotate Left Circular
DDCB0506	RLC	(IX+d)	
FDCB0506	RLC	(IY+d)	
CB07	RLC	A	
CB00	RLC	B	
CB01	RLC	C	
CB02	RLC	D	
CB03	RLC	E	
CB04	RLC	H	
CB05	RLC	L	
07	RLCA		Rotate Left Circular Acc
ED6F	RLD		Rotate Digit Left and Right between Acc. and and Location (HL)
CB1E	RR	(HL)	Rotate Right Through Carry
DDCB051E	RR	(IX+d)	
FDCB051E	RR	(IY+d)	
CB1F	RR	A	
CB18	RR	B	
CB19	RR	C	
CB1A	RR	D	
CB1B	RR	E	
CB1C	RR	H	
CB1D	RR	L	
1F	RRA		Rotate Right Acc. Through Carry
CB0F	RRC	(HL)	Rotate Right Circular
DDCB050E	RRC	(IX+d)	
FDCB050F	RRC	(IY+d)	
CB0F	RRC	A	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB08	RRC	B	Rotate Right Circular
CB09	RRC	C	
CB0A	RRC	D	
CB0B	RRC	E	
CB0C	RRC	H	
CB0D	RRC	L	
0F	RRCA		Rotate Right Circular Acc.
ED67	RRD		Rotate Digit Right and Left Between Acc. and Location (HL)
C7	RST	00H	Restart to Location
CF	RST	08H	
D7	RST	10H	
DF	RST	18H	
E7	RST	20H	
EF	RST	28H	
F7	RST	30H	
FF	RST	38H	
DE20	SBC	A,n	Subtract Operand
9E	SBC	A,(HL)	from Acc. with Carry
DD9E05	SBC	A,(IX+d)	
FD9E05	SBC	A,(IY+d)	
9F	SBC	A,A	
98	SBC	A,B	
99	SBC	A,C	
9A	SBC	A,D	
9B	SBC	A,E	
9C	SBC	A,H	
9D	SBC	A,L	
ED42	SBC	HL,BC	
ED52	SBC	HL,DE	
ED62	SBC	HL,HL	
ED72	SBC	HL,SP	
37	SCF		Set Carry Flag (C = 1)
CBC6	SET	0,(HL)	Set Bit b of Location
DDCB05C6	SET	0,(IX+d)	
FDCB05C6	SET	0,(IY+d)	
CBC7	SET	0,A	
CBC0	SET	0,B	
CBC1	SET	0,C	
CBC2	SET	0,D	
CBC3	SET	0,E	
CBC4	SET	0,H	
CBC5	SET	0,L	
CBCE	SET	1,(HL)	
DDCB05CE	SET	1,(IX+d)	
FDCB05CE	SET	1,(IY+d)	
CBCF	SET	1,A	
CBC8	SET	1,B	
CBC9	SET	1,C	
CBCA	SET	1,D	
CBCB	SET	1,E	
CBCC	SET	1,H	
CBCD	SET	1,L	
CBD6	SET	2,(HL)	
DDCB05D6	SET	2,(IX+d)	
FDCB05D6	SET	2,(IY+d)	
CBD7	SET	2,A	
CBD0	SET	2,B	
CBD1	SET	2,C	
CBD2	SET	2,D	

OBJ CODE	SOURCE STATEMENT		OPERATION
CBD3	SET	2,E	Set Bit b of Location
CBD4	SET	2,H	
CBD5	SET	2,L	
CBD8	SET	3,B	
CBDE	SET	3,(HL)	
DDCB05DE	SET	3,(IX+d)	
FDCB05DE	SET	3,(IY+d)	
CBDF	SET	3,A	
CBD9	SET	3,C	
CBDA	SET	3,D	
CBDB	SET	3,E	
CBDC	SET	3,H	
CBDD	SET	3,L	
CBE6	SET	4,(HL)	
DDCB05E6	SET	4,(IX+d)	
FDCB05E6	SET	4,(IY+d)	
CBE7	SET	4,A	
CBE0	SET	4,B	
CBE1	SET	4,C	
CBE2	SET	4,D	
CBE3	SET	4,E	
CBE4	SET	4,H	
CBE5	SET	4,L	
CBEE	SET	5,(HL)	
DDCB05EE	SET	5,(IX+d)	
FDCB05EE	SET	5,(IY+d)	
CBEF	SET	5,A	
CBE8	SET	5,B	
CBE9	SET	5,C	
CBEA	SET	5,D	
CBEB	SET	5,E	
CBEC	SET	5,H	
CBED	SET	5,L	
CBF6	SET	6,(HL)	
DDCB05F6	SET	6,(IX+d)	
FDCB05F6	SET	6,(IY+d)	
CBF7	SET	6,A	
CBF0	SET	6,B	
CBF1	SET	6,C	
CBF2	SET	6,D	
CBF3	SET	6,E	
CBF4	SET	6,H	
CBF5	SET	6,L	
CBFE	SET	7,(HL)	
DDCB05FE	SET	7,(IX+d)	
FDCB05FE	SET	7,(IY+d)	
CBFF	SET	7,A	
CBF8	SET	7,B	
CBF9	SET	7,C	
CBFA	SET	7,D	
CBFB	SET	7,E	
CBFC	SET	7,H	
CBFD	SET	7,L	
CB26	SLA	(HL)	Shift Operand Left
DDCB0526	SLA	(IX+d)	Arithmetic
FDCB0526	SLA	(IY+d)	
CB27	SLA	A	
CB20	SLA	B	
CB21	SLA	C	
CB22	SLA	D	
CB23	SLA	E	
CB24	SLA	H	
CB25	SLA	L	

OBJ CODE	SOURCE STATEMENT		OPERATION
CB2E	SRA	(HL)	Shift Operand Right
DDCB052E	SRA	(IX+d)	Arithmetic
FDCB052E	SRA	(IY+d)	
CB2F	SRA	A	
CB28	SRA	B	
CB29	SRA	C	
CB2A	SRA	D	
CB2B	SRA	E	
CB2C	SRA	H	
CB2D	SRA	L	
CB3E	SRL	(HL)	Shift Operand Right
DDCB053E	SRL	(IX+d)	Logical
FDCB053E	SRL	(IY+d)	
CB3F	SRL	A	
CB38	SRL	B	
CB39	SRL	C	
CB3A	SRL	D	
CB3B	SRL	E	
CB3C	SRL	H	
CB3D	SRL	L	
96	SUB	(HL)	Subtract Operand
DD9605	SUB	(IX+d)	from Acc.
FD9605	SUB	(IY+d)	
97	SUB	A	
90	SUB	B	
91	SUB	C	
92	SUB	D	
93	SUB	E	
94	SUB	H	
95	SUB	L	
D620	SUB	n	
AE	XOR	(HL)	Exclusive "OR"
DDAE05	XOR	(IX+d)	Operand and Acc.
FDAE05	XOR	(IY+d)	
AF	XOR	A	
A8	XOR	B	
A9	XOR	C	
AA	XOR	D	
AB	XOR	E	
AC	XOR	H	
AD	XOR	L	
EE20	XOR	n	

Appendix 2. CHIP COMPARISON.

When investigating the various attributes of each microprocessor chip, an attempt was made to compare each chip against the other on a points basis, so that a clearer over view could be obtained.

Each chip was given a mark out of 10, after each section was considered fully. It should be noted that these are correct only in the author's view for the particular application in question and may differ greatly for another project, at another time.

A table follows of the results obtained.

CHIP.	4040	PP94	6800	8080	Z80	F8	2650	6502	Cos	6100	Pace	9900	Specification Considered.
	10	1	5	5	7	1	8	1	5	8	9	8	Stack Register.
	6	5	5	5	8	5	5	1	5	5	7	6	Index Register.
	1	3	8	8	8	5	6	5	5	6	10	8	Memory.
	1	4	8	8	9	8	5	10	3	8	6	7	Register Time.
	5	4	7	8	10	5	7	5	9	6	3	6	Instructions.
	2	4	4	6	10	2	2	4	1	1	6	7	Interrupt.
	2	5	5	10	10	8	5	5	3	4	5	5	Input and Output.
	1	1	8	5	9	8	7	10	5	5	6	5	Address Modes.
	1	1	5	8	9	10	6	5	10	8	8	7	General Registers.
	3	3	6	8	8	6	3	3	3	4	6	4	Documentation.
	2	2	6	9	8	5	4	4	4	4	2	4	Cost and Support.
	2	2	8	10	10	5	4	2	2	2	4	3	Availability and Family.
	4	4	6	10	8	2	2	2	2	2	8	3	Industrial Standards.
	2	2	5	8	10	3	4	1	2	2	3	3	Machine Control.
	42	41	86	108	124	73	68	58	59	65	83	76	Totals.

APPENDIX 3. SYSTEM MONITOR LISTING.

```
EC400 0000 0800
0000  31 00 10        LD    SP,£1000     ;1..
0003  D7              RST   £10          ;W
0004  08              DEFB  £08          ;.
0005  C3 B2 03        JP    £03B2        ;C2.
0008  DF              RST   £18          ;_
0009  62              DEFB  £62          ;b
000A  D8              RET    C           ;X
000B  18 FB           JR    £0008        ;.{
000D  C3 9A 03        JP    £039A        ;C..
0010  E5              PUSH  HL           ;e
0011  E1              POP   HL           ;a
0012  E1              POP   HL           ;a
0013  23              INC   HL           ;£
0014  E5              PUSH  HL           ;e
0015  C3 70 05        JP    £0570        ;Cp.
0018  E5              PUSH  HL           ;e
0019  E1              POP   HL           ;a
001A  E1              POP   HL           ;a
001B  23              INC   HL           ;£
001C  E5              PUSH  HL           ;e
001D  C3 80 05        JP    £0580        ;C..
0020  E3              EX    (SP),HL      ;c
0021  2B              DEC   HL           ;+
0022  E3              EX    (SP),HL      ;c
0023  C3 1A 04        JP    £041A        ;C..
0026  00              NOP                ;.
0027  00              NOP                ;.
0028  E3              EX    (SP),HL      ;c
0029  7E              LD    A,(HL)       ;~
002A  23              INC   HL           ;£
002B  B7              OR    A            ;7
002C  20 06           JR    NZ,£0034     ; .
002E  E3              EX    (SP),HL      ;c
002F  C9              RET                ;I
0030  E5              PUSH  HL           ;e
0031  C3 5A 07        JP    £075A        ;CZ.
0034  F7              RST   £30          ;w
0035  18 F2           JR    £0029        ;.r
0037  00              NOP                ;.
0038  3D              DEC   A            ;=
0039  C8              RET    Z           ;H
003A  F5              PUSH  AF           ;u
003B  F1              POP   AF           ;q
003C  18 FA           JR    £0038        ;.z
003E  AF              XOR   A            ;/
```

```
003F  47                LD    B,A          ;G
0040  FF                RST   £38          ;.
0041  FF                RST   £38          ;.
0042  10 FC             DJNZ  £0040        ;.|
0044  C9                RET                ;I
0045  E5                PUSH  HL           ;e
0046  21 00 0C          LD    HL,£0C00     ;!..
0049  AE                XOR   (HL)         ;.
004A  D3 00              OUT   (£00),A      ;S.
004C  7E                LD    A,(HL)       ;~
004D  D3 00              OUT   (£00),A      ;S.
004F  E1                POP   HL           ;a
0050  C9                RET                ;I
0051  3E 10              LD    A,£10        ;>.
0053  E5                PUSH  HL           ;e
0054  21 00 0C          LD    HL,£0C00     ;!..
0057  AE                XOR   (HL)         ;.
0058  77                LD    (HL),A       ;w
0059  18 F2              JR    £004D        ;.r
005B  F5                PUSH  AF           ;u
005C  D3 01              OUT   (£01),A      ;S.
005E  DB 02              IN    A,(£02)      ;[.
0060  CB 77              BIT   6,A          ;Kw
0062  28 FA              JR     Z,£005E     ;(z
0064  F1                POP   AF           ;q
0065  C9                RET                ;I
0066  C3 7D 0C          JP    £0C7D        ;C}.
0069  1E C0              LD    E,£C0        ;.@
006B  DF                RST   £18          ;_
006C  62                DEFB  £62          ;b
006D  D8                RET    C           ;X
006E  1D                DEC   E            ;.
006F  20 FA              JR    NZ,£006B     ; z
0071  C9                RET                ;I
0072  2A 29 0C          LD    HL,(£0C29)   ;*).
0075  56                LD    D,(HL)       ;V
0076  36 3F              LD    (HL),£3F     ;6?
0078  D7                RST   £10          ;W
0079  EF                DEFB  £EF          ;o
007A  72                LD    (HL),D       ;r
007B  D8                RET    C           ;X
007C  D7                RST   £10          ;W
007D  EB                DEFB  £EB          ;k
007E  30 F2              JR    NC,£0072     ;0r
0080  C9                RET                ;I
0081  DB 02              IN    A,(£02)      ;[.
0083  17                RLA                ;.
```

```
0084  D0          RET   NC           ;P
0085  DB 01       IN    A,(£01)      ;[.
0087  C9          RET                ;I
0088  3E 02       LD    A,£02        ;>.
008A  CD 45 00    CALL  £0045        ;ME.
008D  21 01 0C    LD    HL,£0C01     ;!..
0090  DB 00       IN    A,(£00)      ;[.
0092  2F          CPL                ;/
0093  77          LD    (HL),A       ;w
0094  06 08       LD    B,£08        ;..
0096  3E 01       LD    A,£01        ;>.
0098  CD 45 00    CALL  £0045        ;ME.
009B  23          INC   HL           ;£
009C  DB 00       IN    A,(£00)      ;[.
009E  2F          CPL                ;/
009F  57          LD    D,A          ;W
00A0  AE          XOR   (HL)         ;.
00A1  20 04       JR    NZ,£00A7     ; .
00A3  10 F1       DJNZ  £0096        ;.q
00A5  B7          OR    A            ;7
00A6  C9          RET                ;I
00A7  AF          XOR   A            ;/
00A8  FF          RST   £38          ;.
00A9  DB 00       IN    A,(£00)      ;[.
00AB  2F          CPL                ;/
00AC  5F          LD    E,A          ;_
00AD  7A          LD    A,D          ;z
00AE  AE          XOR   (HL)         ;.
00AF  0E FF       LD    C,£FF        ;..
00B1  16 00       LD    D,£00        ;..
00B3  37          SCF                ;7
00B4  CB 12       RL    D            ;K.
00B6  0C          INC   C            ;.
00B7  1F          RRA                ;.
00B8  30 FA       JR    NC,£00B4     ;0z
00BA  7A          LD    A,D          ;z
00BB  A3          AND   E            ;£
00BC  5F          LD    E,A          ;_
00BD  7E          LD    A,(HL)       ;~
00BE  A2          AND   D            ;"
00BF  BB          CP    E            ;;
00C0  28 E1       JR     Z,£00A3     ;(a
00C2  7E          LD    A,(HL)       ;~
00C3  AA          XOR   D            ;*
00C4  77          LD    (HL),A       ;w
00C5  7B          LD    A,E          ;{
00C6  B7          OR    A            ;7
00C7  28 DA       JR     Z,£00A3     ;(Z
```

```
00C9  3A 01 0C      LD    A,(£0C01)     ;:..
00CC  E6 10         AND   £10           ;f.
00CE  B0            OR    B             ;0
00CF  87            ADD   A,A           ;.
00D0  87            ADD   A,A           ;.
00D1  87            ADD   A,A           ;.
00D2  B1            OR    C             ;1
00D3  D7            RST   £10           ;W
00D4  5B            DEFB  £5B           ;[
00D5  28 06         JR     Z,£00DD      ;(.
00D7  E6 7F         AND   £7F           ;f.
00D9  D7            RST   £10           ;W
00DA  55            DEFB  £55           ;U
00DB  20 C8         JR    NZ,£00A5      ; H
00DD  37            SCF                 ;7
00DE  ED 52         SBC   HL,DE         ;mR
00E0  7D            LD    A,L           ;}
00E1  FE 41         CP    £41           ;~A
00E3  38 1E         JR     C,£0103      ;8.
00E5  FE 5B         CP    £5B           ;~[
00E7  30 1A         JR    NC,£0103      ;0.
00E9  21 01 0C      LD    HL,£0C01      ;!..
00EC  CB 66         BIT   4,(HL)        ;Kf
00EE  21 27 0C      LD    HL,£0C27      ;!'.
00F1  20 08         JR    NZ,£00FB      ; .
00F3  CB 46         BIT   0,(HL)        ;KF
00F5  28 0C         JR     Z,£0103      ;(.
00F7  C6 20         ADD   A,£20         ;F
00F9  18 08         JR    £0103         ;..
00FB  C6 20         ADD   A,£20         ;F
00FD  CB 46         BIT   0,(HL)        ;KF
00FF  28 02         JR     Z,£0103      ;(.
0101  D6 20         SUB   £20           ;V
0103  21 01 0C      LD    HL,£0C01      ;!..
0106  FE 40         CP    £40           ;~@
0108  20 06         JR    NZ,£0110      ; .
010A  CB 66         BIT   4,(HL)        ;Kf
010C  28 97         JR     Z,£00A5      ;(.
010E  18 06         JR    £0116         ;..
0110  CB 6E         BIT   5,(HL)        ;Kn
0112  28 02         JR     Z,£0116      ;(.
0114  EE 40         XOR   £40           ;n@
0116  CB 5E         BIT   3,(HL)        ;K^
0118  28 02         JR     Z,£011C      ;(.
011A  EE 40         XOR   £40           ;n@
011C  21 06 0C      LD    HL,£0C06      ;!..
011F  CB 76         BIT   6,(HL)        ;Kv
```

```
0121  28 02          JR    Z,£0125     ;(.
0123  EE 80          XOR   £80         ;n.
0125  21 27 0C       LD    HL,£0C27    ;!'.
0128  CB 56          BIT   2,(HL)      ;KV
012A  28 02          JR    Z,£012E     ;(.
012C  EE 80          XOR   £80         ;n.
012E  37             SCF               ;7
012F  C9             RET               ;I
0130  2A 6F 0C       LD    HL,(£0C6F)  ;*o.
0133  54             LD    D,H         ;T
0134  5D             LD    E,L         ;]
0135  ED 4B 6D 0C    LD    BC,(£0C6D)  ;mKm.
0139  ED B1          CPIR              ;m1
013B  C9             RET               ;I
013C  00             NOP               ;.
013D  10 60          DJNZ  £019F       ;.`
013F  00             NOP               ;.
0140  9E             SBC   A,(HL)      ;.
0141  05             DEC   B           ;.
0142  06 07          LD    B,£07       ;..
0144  7F             LD    A,A         ;.
0145  07             RLCA              ;.
0146  82             ADD   A,D         ;.
0147  07             RLCA              ;.
0148  C3 2F 00       JP    £002F       ;C/.
014B  C3 2F 00       JP    £002F       ;C/.
014E  C3 B7 C8       JP    £C8B7       ;C7H
0151  F5             PUSH  AF          ;u
0152  FE 0A          CP    £0A         ;~.
0154  28 24          JR    Z,£017A     ;($
0156  FE 0C          CP    £0C         ;~.
0158  20 22          JR    NZ,£017C    ;  "
015A  21 0A 08       LD    HL,£080A    ;!..
015D  E5             PUSH  HL          ;e
015E  06 30          LD    B,£30       ;.0
0160  36 20          LD    (HL),£20    ;6
0162  23             INC   HL          ;£
0163  10 FB          DJNZ  £0160       ;.{
0165  06 10          LD    B,£10       ;..
0167  36 00          LD    (HL),£00    ;6.
0169  23             INC   HL          ;£
016A  10 FB          DJNZ  £0167       ;.{
016C  EB             EX    DE,HL       ;k
016D  E1             POP   HL          ;a
016E  E5             PUSH  HL          ;e
016F  01 B0 03       LD    BC,£03B0    ;.0.
0172  ED B0          LDIR              ;m0
0174  E1             POP   HL          ;a
```

```
0175  DF          RST  £18          ;_
0176  7C          DEFB £7C          ;|
0177  22 29 0C    LD   (£0C29),HL   ;").
017A  F1          POP  AF           ;q
017B  C9          RET               ;I
017C  2A 29 0C    LD   HL,(£0C29)   ;*).
017F  FE 08       CP   £08          ;~.
0181  20 11       JR   NZ,£0194     ;  .
0183  F5          PUSH AF           ;u
0184  2B          DEC  HL           ;+
0185  7E          LD   A,(HL)       ;~
0186  B7          OR   A            ;7
0187  28 FB       JR    Z,£0184     ;((
0189  F1          POP  AF           ;q
018A  FE 11       CP   £11          ;~.
018C  28 02       JR    Z,£0190     ;(.
018E  36 20       LD   (HL),£20     ;6
0190  D7          RST  £10          ;W
0191  63          DEFB £63          ;c
0192  18 E6       JR   £017A        ;.f
0194  FE 11       CP   £11          ;~.
0196  28 EB       JR    Z,£0183     ;(k
0198  FE 17       CP   £17          ;~.
019A  28 D9       JR    Z,£0175     ;(Y
019C  FE 1B       CP   £1B          ;~.
019E  20 0B       JR   NZ,£01AB     ;  .
01A0  DF          RST  £18          ;_
01A1  7C          DEFB £7C          ;|
01A2  06 30       LD   B,£30        ;.0
01A4  36 20       LD   (HL),£20     ;6
01A6  23          INC  HL           ;£
01A7  10 FB       DJNZ £01A4        ;.{
01A9  18 CA       JR   £0175        ;.J
01AB  FE 0D       CP   £0D          ;~.
01AD  28 66       JR    Z,£0215     ;(f
01AF  FE 18       CP   £18          ;~.
01B1  20 0C       JR   NZ,£01BF     ;  .
01B3  E5          PUSH HL           ;e
01B4  DF          RST  £18          ;_
01B5  7C          DEFB £7C          ;|
01B6  D1          POP  DE           ;Q
01B7  B7          OR   A            ;7
01B8  ED 52       SBC  HL,DE        ;mR
01BA  19          ADD  HL,DE        ;.
01BB  28 BA       JR    Z,£0177     ;(:
01BD  18 56       JR   £0215        ;.V
```

```
01BF  FE 13        CP   £13          ;~.
01C1  20 08        JR   NZ,£01CB     ;  .
01C3  11 C0 FF     LD   DE,£FFC0     ;.@.
01C6  19           ADD  HL,DE        ;.
01C7  D7           RST  £10          ;W
01C8  2C           DEFB £2C          ;,
01C9  18 AF        JR   £017A        ;./
01CB  FE 14        CP   £14          ;~.
01CD  20 05        JR   NZ,£01D4     ;  .
01CF  11 40 00     LD   DE,£0040     ;.@.
01D2  18 F2        JR   £01C6        ;.r
01D4  FE 15        CP   £15          ;~.
01D6  20 0E        JR   NZ,£01E6     ;  .
01D8  23           INC  HL           ;£
01D9  7E           LD   A,(HL)       ;~
01DA  2B           DEC  HL           ;+
01DB  B7           OR   A            ;7
01DC  20 04        JR   NZ,£01E2     ;  .
01DE  36 20        LD   (HL),£20     ;6
01E0  18 98        JR   £017A        ;..
01E2  77           LD   (HL),A       ;w
01E3  23           INC  HL           ;£
01E4  18 F2        JR   £01D8        ;.r
01E6  FE 16        CP   £16          ;~.
01E8  20 1F        JR   NZ,£0209     ;  .
01EA  06 20        LD   B,£20        ;.
01EC  7E           LD   A,(HL)       ;~
01ED  B7           OR   A            ;7
01EE  28 8A        JR    Z,£017A     ;(.
01F0  70           LD   (HL),B       ;p
01F1  47           LD   B,A          ;G
01F2  23           INC  HL           ;£
01F3  18 F7        JR   £01EC        ;.w
01F5  11 0A 08     LD   DE,£080A     ;...
01F8  B7           OR   A            ;7
01F9  ED 52        SBC  HL,DE        ;mR
01FB  19           ADD  HL,DE        ;.
01FC  D8           RET   C           ;X
01FD  11 BA 0B     LD   DE,£0BBA     ;.:.
0200  B7           OR   A            ;7
0201  ED 52        SBC  HL,DE        ;mR
0203  19           ADD  HL,DE        ;.
0204  D0           RET  NC           ;P
0205  F1           POP  AF           ;q
0206  C3 77 01     JP   £0177        ;Cw.
0209  FE 12        CP   £12          ;~.
```

```
020B  28 01        JR     Z,£020E     ;(.
020D  77           LD     (HL),A      ;w
020E  23           INC    HL          ;£
020F  7E           LD     A,(HL)      ;~
0210  B7           OR     A           ;7
0211  28 FB        JR     Z,£020E     ;({
0213  D7           RST    £10         ;W
0214  E0           DEFB   £E0         ;`
0215  DF           RST    £18         ;_
0216  7C           DEFB   £7C         ;|
0217  11 40 00     LD     DE,£0040    ;.@.
021A  19           ADD    HL,DE       ;.
021B  D7           RST    £10         ;W
021C  D8           DEFB   £D8         ;X
021D  11 0A 08     LD     DE,£080A    ;...
0220  21 4A 08     LD     HL,£084A    ;!J.
0223  01 70 03     LD     BC,£0370    ;.p.
0226  ED B0        LDIR               ;m0
0228  06 30        LD     B,£30       ;.0
022A  2B           DEC    HL          ;+
022B  36 20        LD     (HL),£20    ;6
022D  10 FB        DJNZ   £022A       ;.{
022F  21 8A 0B     LD     HL,£0B8A    ;!..
0232  18 D2        JR     £0206       ;.R
0234  7D           LD     A,L         ;}
0235  D6 40        SUB    £40         ;V@
0237  30 FC        JR     NC,£0235    ;0|
0239  C6 36        ADD    A,£36       ;F6
023B  5F           LD     E,A         ;_
023C  7D           LD     A,L         ;}
023D  93           SUB    E           ;.
023E  6F           LD     L,A         ;o
023F  C9           RET                ;I
0240  DF           RST    £18         ;_
0241  60           DEFB   £60         ;`
0242  22 0C 0C     LD     (£0C0C),HL  ;"..
0245  DF           RST    £18         ;_
0246  66           DEFB   £66         ;f
0247  7E           LD     A,(HL)      ;~
0248  DF           RST    £18         ;_
0249  68           DEFB   £68         ;h
024A  EF           RST    £28         ;o
024B  20           DEFM   / /         ;
024C  11 11 11     DEFB   £11,£11,£11 ;...
024F  00           DEFB   £00         ;.
0250  D7           RST    £10         ;W
0251  54           DEFB   £54         ;T
```

```
0252  DF          RST  £18          ;_
0253  64          DEFB £64          ;d
0254  38 4C       JR    C,£02A2     ;8L
0256  7E          LD   A,(HL)       ;~
0257  B7          OR   A            ;7
0258  28 48       JR    Z,£02A2     ;(H
025A  23          INC  HL           ;£
025B  D5          PUSH DE           ;U
025C  5E          LD   E,(HL)       ;^
025D  23          INC  HL           ;£
025E  56          LD   D,(HL)       ;V
025F  EB          EX   DE,HL        ;k
0260  D1          POP  DE           ;Q
0261  06 00       LD   B,£00        ;..
0263  E5          PUSH HL           ;e
0264  DF          RST  £18          ;_
0265  64          DEFB £64          ;d
0266  7E          LD   A,(HL)       ;~
0267  B7          OR   A            ;7
0268  28 07       JR    Z,£0271     ;(.
026A  23          INC  HL           ;£
026B  7E          LD   A,(HL)       ;~
026C  E1          POP  HL           ;a
026D  77          LD   (HL),A       ;w
026E  04          INC  B            ;.
026F  23          INC  HL           ;£
0270  E5          PUSH HL           ;e
0271  E1          POP  HL           ;a
0272  1A          LD   A,(DE)       ;.
0273  FE 2E       CP   £2E          ;~.
0275  C8          RET   Z           ;H
0276  FE 2C       CP   £2C          ;~,
0278  20 05       JR   NZ,£027F     ; .
027A  13          INC  DE           ;.
027B  1A          LD   A,(DE)       ;.
027C  13          INC  DE           ;.
027D  18 EE       JR   £026D        ;.n
027F  78          LD   A,B          ;x
0280  B7          OR   A            ;7
0281  20 01       JR   NZ,£0284     ; .
0283  23          INC  HL           ;£
0284  1A          LD   A,(DE)       ;.
0285  FE 3A       CP   £3A          ;~:
0287  20 04       JR   NZ,£028D     ; .
0289  2B          DEC  HL           ;+
028A  2B          DEC  HL           ;+
028B  18 B5       JR   £0242        ;.5
```

```
028D  FE 2F        CP   £2F          ;~/
028F  20 0A        JR   NZ,£029B     ; .
0291  13           INC  DE           ;.
0292  DF           RST  £18          ;_
0293  64           DEFB £64          ;d
0294  38 0C        JR    C,£02A2     ;8.
0296  2A 21 0C     LD   HL,(£0C21)   ;*!.
0299  18 A7        JR   £0242        ;.'
029B  B7           OR   A            ;7
029C  28 A4        JR    Z,£0242     ;($
029E  FE 20        CP   £20          ;~
02A0  28 C1        JR    Z,£0263     ;(A
02A2  DF           RST  £18          ;_
02A3  6B           DEFB £6B          ;k
02A4  18 9A        JR   £0240        ;..
02A6  E5           PUSH HL           ;e
02A7  CD 8A 03     CALL £038A        ;M..
02AA  AF           XOR  A            ;/
02AB  32 26 0C     LD   (£0C26),A    ;2&.
02AE  21 1A 04     LD   HL,£041A     ;!..
02B1  22 7E 0C     LD   (£0C7E),HL   ;"~.
02B4  E1           POP  HL           ;a
02B5  E5           PUSH HL           ;e
02B6  DF           RST  £18          ;_
02B7  7B           DEFB £7B          ;{
02B8  F7           RST  £30          ;w
02B9  FE 0D        CP   £0D          ;~.
02BB  20 F9        JR   NZ,£02B6     ; y
02BD  2A 29 0C     LD   HL,(£0C29)   ;*).
02C0  11 C0 FF     LD   DE,£FFC0     ;.@.
02C3  19           ADD  HL,DE        ;.
02C4  EB           EX   DE,HL        ;k
02C5  E1           POP  HL           ;a
02C6  C9           RET               ;I
02C7  C5           PUSH BC           ;E
02C8  18 17        JR   £02E1        ;..
02CA  B7           OR   A            ;7
02CB  ED 52        SBC  HL,DE        ;mR
02CD  19           ADD  HL,DE        ;.
02CE  38 06        JR    C,£02D6     ;8.
02D0  C1           POP  BC           ;A
02D1  EF           RST  £28          ;o
02D2  2E           DEFM /./          ;.
02D3  0D 00        DEFB £0D,£00      ;..
02D5  C9           RET               ;I
02D6  78           LD   A,B          ;x
```

```
02D7  B1                  OR    C               ;1
02D8  20 07               JR    NZ,£02E1        ; .
02DA  CF                  RST   £08             ;O
02DB  FE 1B               CP    £1B             ;~.
02DD  28 F1               JR     Z,£02DO        ;(q
02DF  C1                  POP   BC              ;A
02E0  C5                  PUSH  BC              ;E
02E1  0B                  DEC   BC              ;.
02E2  C5                  PUSH  BC              ;E
02E3  0E 00               LD    C,£00           ;..
02E5  EF                  RST   £28             ;o
02E6  20 20               DEFM /  /             ;
02E8  00                  DEFB £00              ;.
02E9  DF                  RST   £18             ;_
02EA  66                  DEFB £66              ;f
02EB  06 08               LD    B,£08           ;..
02ED  7E                  LD    A,(HL)          ;~
02EE  DF                  RST   £18             ;_
02EF  67                  DEFB £67              ;g
02F0  23                  INC   HL              ;£
02F1  DF                  RST   £18             ;_
02F2  69                  DEFB £69              ;i
02F3  10 F8               DJNZ £02ED            ;.x
02F5  79                  LD    A,C             ;y
02F6  DF                  RST   £18             ;_
02F7  68                  DEFB £68              ;h
02F8  EF                  RST   £28             ;o
02F9  08 08 0D            DEFB £08,£08,£0D      ;...
02FC  00                  DEFB £00              ;.
02FD  C1                  POP   BC              ;A
02FE  18 CA               JR    £02CA           ;.J
0300  7C                  LD    A,H             ;|
0301  DF                  RST   £18             ;_
0302  67                  DEFB £67              ;g
0303  7D                  LD    A,L             ;}
0304  DF                  RST   £18             ;_
0305  67                  DEFB £67              ;g
0306  3E 20               LD    A,£20           ;>
0308  F7                  RST   £30             ;w
0309  C9                  RET                   ;I
030A  EF                  RST   £28             ;o
030B  45 72 72 6F         DEFM /Erro/           ;Erro
030F  72                  DEFM /r/              ;r
0310  00                  DEFB £00              ;.
0311  3E 0D               LD    A,£0D           ;>.
```

```
0313  F7           RST  £30          ;w
0314  C9           RET               ;I
0315  F5           PUSH AF           ;u
0316  81           ADD  A,C          ;.
0317  4F           LD   C,A          ;O
0318  F1           POP  AF           ;q
0319  F5           PUSH AF           ;u
031A  1F           RRA               ;.
031B  1F           RRA               ;.
031C  1F           RRA               ;.
031D  1F           RRA               ;.
031E  D7           RST  £10          ;W
031F  01           DEFB £01          ;.
0320  F1           POP  AF           ;q
0321  E6 0F        AND  £0F          ;f.
0323  C6 30        ADD  A,£30        ;F0
0325  FE 3A        CP   £3A          ;~:
0327  38 02        JR    C,£032B     ;8.
0329  C6 07        ADD  A,£07        ;F.
032B  F7           RST  £30          ;w
032C  C9           RET               ;I
032D  D7           RST  £10          ;W
032E  00           DEFB £00          ;.
032F  DF           RST  £18          ;_
0330  66           DEFB £66          ;f
0331  EB           EX   DE,HL        ;k
0332  C9           RET               ;I
0333  1A           LD   A,(DE)       ;.
0334  FE 20        CP   £20          ;~
0336  13           INC  DE           ;.
0337  28 FA        JR    Z,£0333     ;(z
0339  1B           DEC  DE           ;.
033A  21 00 00     LD   HL,£0000     ;!..
033D  22 21 0C     LD   (£0C21),HL   ;"!.
0340  AF           XOR  A            ;/
0341  21 20 0C     LD   HL,£0C20     ;!  .
0344  77           LD   (HL),A       ;w
0345  1A           LD   A,(DE)       ;.
0346  B7           OR   A            ;7
0347  C8           RET   Z           ;H
0348  FE 20        CP   £20          ;~
034A  C8           RET   Z           ;H
034B  D6 30        SUB  £30          ;V0
034D  D8           RET   C           ;X
034E  FE 0A        CP   £0A          ;~.
0350  38 0B        JR    C,£035D     ;8.
```

```
0352  D6 07          SUB  £07          ;V.
0354  FE 0A          CP   £0A          ;~.
0356  D8             RET   C           ;X
0357  FE 10          CP   £10          ;~.
0359  38 02          JR    C,£035D     ;8.
035B  37             SCF               ;7
035C  C9             RET               ;I
035D  13             INC  DE           ;.
035E  34             INC  (HL)         ;4
035F  23             INC  HL           ;£
0360  ED 6F          RLD               ;mo
0362  23             INC  HL           ;£
0363  ED 6F          RLD               ;mo
0365  2B             DEC  HL           ;+
0366  2B             DEC  HL           ;+
0367  28 DC          JR    Z,£0345     ;(\
0369  1B             DEC  DE           ;.
036A  37             SCF               ;7
036B  C9             RET               ;I
036C  01 0B 0C       LD   BC,£0C0B     ;...
036F  AF             XOR  A            ;/
0370  02             LD   (BC),A       ;.
0371  DF             RST  £18          ;_
0372  64             DEFB £64          ;d
0373  D8             RET   C           ;X
0374  7E             LD   A,(HL)       ;~
0375  B7             OR   A            ;7
0376  C8             RET   Z           ;H
0377  23             INC  HL           ;£
0378  03             INC  BC           ;.
0379  7E             LD   A,(HL)       ;~
037A  02             LD   (BC),A       ;.
037B  23             INC  HL           ;£
037C  03             INC  BC           ;.
037D  7E             LD   A,(HL)       ;~
037E  02             LD   (BC),A       ;.
037F  21 0B 0C       LD   HL,£0C0B     ;!..
0382  34             INC  (HL)         ;4
0383  7E             LD   A,(HL)       ;~
0384  FE 0B          CP   £0B          ;~.
0386  38 E9          JR    C,£0371     ;8i
0388  37             SCF               ;7
0389  C9             RET               ;I
038A  2A 23 0C       LD   HL,(£0C23)   ;*£.
038D  7E             LD   A,(HL)       ;~
038E  32 25 0C       LD   (£0C25),A    ;2%.
0391  C9             RET               ;I
```

```
0392  2A 23 0C          LD    HL,(£0C23)   ;*£.
0395  3A 25 0C          LD    A,(£0C25)    ;:%.
0398  77                LD    (HL),A       ;w
0399  C9                RET                ;I
039A  D7                RST   £10          ;W
039B  F6                DEFB  £F6          ;v
039C  11 00 0C          LD    DE,£0C00     ;...
039F  06 6B             LD    B,£6B        ;.k
03A1  AF                XOR   A            ;/
03A2  12                LD    (DE),A       ;.
03A3  13                INC   DE           ;.
03A4  10 FC             DJNZ  £03A2        ;.¦
03A6  21 3C 01          LD    HL,£013C     ;!<.
03A9  01 15 00          LD    BC,£0015     ;...
03AC  ED B0             LDIR               ;m0
03AE  EF                RST   £28          ;o
03AF  0C 00             DEFB  £0C,£00      ;..
03B1  C9                RET                ;I
03B2  31 61 0C          LD    SP,£0C61     ;1a.
03B5  2A 3C 01          LD    HL,(£013C)   ;*<.
03B8  22 6B 0C          LD    (£0C6B),HL   ;"k.
03BB  EF                RST   £28          ;o
03BC  48 4F 50 53       DEFM  /HOPS/       ;HOPS
03C0  59 53 54 45       DEFM  /YSTE/       ;YSTE
03C4  4D                DEFM  /M/          ;M
03C5  0D 00             DEFB  £0D,£00      ;..
03C7  D7                RST   £10          ;W
03C8  C9                DEFB  £C9          ;I
03C9  CD A6 02          CALL  £02A6        ;M&.
03CC  01 2B 0C          LD    BC,£0C2B     ;.+.
03CF  1A                LD    A,(DE)       ;.
03D0  FE 20             CP    £20          ;~
03D2  20 05             JR    NZ,£03D9     ; .
03D4  0A                LD    A,(BC)       ;.
03D5  FE 53             CP    £53          ;~S
03D7  20 F0             JR    NZ,£03C9     ; p
03D9  FE 41             CP    £41          ;~A
03DB  38 0D             JR     C,£03EA     ;8.
03DD  FE 5B             CP    £5B          ;~[
03DF  30 09             JR    NC,£03EA     ;0.
03E1  02                LD    (BC),A       ;.
03E2  32 0A 0C          LD    (£0C0A),A    ;2..
03E5  13                INC   DE           ;.
03E6  DF                RST   £18          ;_
03E7  79                DEFB  £79          ;y
03E8  30 04             JR    NC,£03EE     ;0.
```

```
03EA  DF          RST  £18          ;_
03EB  6B          DEFB £6B          ;k
03EC  18 DB       JR   £03C9        ;.[
03EE  DF          RST  £18          ;_
03EF  60          DEFB £60          ;`
03F0  DF          RST  £18          ;_
03F1  5C          DEFB £5C          ;\
03F2  18 D5       JR   £03C9        ;.U
03F4  3E FF       LD   A,£FF        ;>.
03F6  32 26 0C    LD   (£0C26),A    ;2&.
03F9  F1          POP  AF           ;q
03FA  3A 0B 0C    LD   A,(£0C0B)    ;:..
03FD  B7          OR   A            ;7
03FE  28 03       JR    Z,£0403     ;(.
0400  22 69 0C    LD   (£0C69),HL   ;"i.
0403  C1          POP  BC           ;A
0404  D1          POP  DE           ;Q
0405  F1          POP  AF           ;q
0406  F1          POP  AF           ;q
0407  2A 6B 0C    LD   HL,(£0C6B)   ;*k.
040A  F9          LD   SP,HL        ;y
040B  2A 69 0C    LD   HL,(£0C69)   ;*i.
040E  E5          PUSH HL           ;e
040F  2A 65 0C    LD   HL,(£0C65)   ;*e.
0412  F5          PUSH AF           ;u
0413  3E 08       LD   A,£08        ;>.
0415  D3 00       OUT  (£00),A      ;S.
0417  F1          POP  AF           ;q
0418  ED 45       RETN              ;mE
041A  F5          PUSH AF           ;u
041B  E5          PUSH HL           ;e
041C  3A 00 0C    LD   A,(£0C00)    ;:..
041F  D3 00       OUT  (£00),A      ;S.
0421  3A 26 0C    LD   A,(£0C26)    ;:&.
0424  B7          OR   A            ;7
0425  28 0D       JR    Z,£0434     ;(.
0427  CD 8A 03    CALL £038A        ;M..
042A  36 E7       LD   (HL),£E7     ;6g
042C  AF          XOR  A            ;/
042D  32 26 0C    LD   (£0C26),A    ;2&.
0430  E1          POP  HL           ;a
0431  F1          POP  AF           ;q
0432  ED 45       RETN              ;mE
0434  D5          PUSH DE           ;U
0435  C5          PUSH BC           ;E
0436  21 00 00    LD   HL,£0000     ;!..
0439  39          ADD  HL,SP        ;9
```

```
043A  31 61 0C        LD    SP,£0C61      ;1a.
043D  11 61 0C        LD    DE,£0C61      ;.a.
0440  01 08 00        LD    BC,£0008      ;...
0443  ED B0           LDIR                ;m0
0445  5E              LD    E,(HL)        ;^
0446  23              INC   HL            ;£
0447  56              LD    D,(HL)        ;V
0448  23              INC   HL            ;£
0449  ED 53 69 0C     LD    (£0C69),DE    ;mSi.
044D  22 6B 0C        LD    (£0C6B),HL    ;"k.
0450  EF              RST   £28           ;o
0451  18 00           DEFB  £18,£00       ;..
0453  21 6D 0C        LD    HL,£0C6D      ;!m.
0456  06 06           LD    B,£06         ;..
0458  2B              DEC   HL            ;+
0459  7E              LD    A,(HL)        ;~
045A  DF              RST   £18           ;_
045B  68              DEFB  £68           ;h
045C  2B              DEC   HL            ;+
045D  7E              LD    A,(HL)        ;~
045E  DF              RST   £18           ;_
045F  68              DEFB  £68           ;h
0460  DF              RST   £18           ;_
0461  69              DEFB  £69           ;i
0462  10 F4           DJNZ  £0458         ;.t
0464  ED 57           LD    A,I           ;mW
0466  DF              RST   £18           ;_
0467  68              DEFB  £68           ;h
0468  DF              RST   £18           ;_
0469  69              DEFB  £69           ;i
046A  DD E5           PUSH  IX            ;]e
046C  E1              POP   HL            ;a
046D  DF              RST   £18           ;_
046E  66              DEFB  £66           ;f
046F  FD E5           PUSH  IY            ;}e
0471  E1              POP   HL            ;a
0472  DF              RST   £18           ;_
0473  66              DEFB  £66           ;f
0474  3A 67 0C        LD    A,(£0C67)     ;:g.
0477  11 8B 04        LD    DE,£048B      ;...
047A  06 08           LD    B,£08         ;..
047C  13              INC   DE            ;.
047D  17              RLA                 ;.
047E  F5              PUSH  AF            ;u
047F  1A              LD    A,(DE)        ;.
0480  30 01           JR    NC,£0483      ;0.
0482  F7              RST   £30           ;w
```

```
0483  F1           POP   AF          ;q
0484  10 F6        DJNZ  £047C       ;.v
0486  EF           RST   £28         ;o
0487  18 00        DEFB  £18,£00     ;..
0489  C3 C7 03     JP    £03C7       ;CG.
048C  53           LD    D,E         ;S
048D  5A           LD    E,D         ;Z
048E  00           NOP               ;.
048F  48           LD    C,B         ;H
0490  00           NOP               ;.
0491  50           LD    D,B         ;P
0492  4E           LD    C,(HL)      ;N
0493  43           LD    B,E         ;C
0494  DF           RST   £18         ;_
0495  5F           DEFB  £5F         ;_
0496  DF           RST   £18         ;_
0497  77           DEFB  £77         ;w
0498  E5           PUSH  HL          ;e
0499  DF           RST   £18         ;_
049A  78           DEFB  £78         ;x
049B  E5           PUSH  HL          ;e
049C  CF           RST   £08         ;O
049D  E6 7F        AND   £7F         ;f.
049F  FE 2E        CP    £2E         ;~.
04A1  28 3A        JR     Z,£04DD    ;(:
04A3  FE 0D        CP    £0D         ;~.
04A5  28 07        JR     Z,£04AE    ;(.
04A7  FE 20        CP    £20         ;~
04A9  38 F1        JR     C,£049C    ;8q
04AB  F7           RST   £30         ;w
04AC  18 EE        JR    £049C       ;.n
04AE  2A 29 0C     LD    HL,(£0C29)  ;*).
04B1  DF           RST   £18         ;_
04B2  7C           DEFB  £7C         ;|
04B3  EB           EX    DE,HL       ;k
04B4  DF           RST   £18         ;_
04B5  79           DEFB  £79         ;y
04B6  38 21        JR     C,£04D9    ;8!
04B8  21 0C 0C     LD    HL,£0C0C    ;!..
04BB  AF           XOR   A           ;/
04BC  06 12        LD    B,£12       ;..
04BE  86           ADD   A,(HL)      ;.
04BF  23           INC   HL          ;£
04C0  10 FC        DJNZ  £04BE       ;.|
04C2  BE           CP    (HL)        ;>
04C3  20 14        JR    NZ,£04D9    ; .
```

```
04C5  2A 0C 0C        LD    HL,(£0C0C)    ;*..
04C8  11 0E 0C        LD    DE,£0C0E      ;...
04CB  06 08           LD    B,£08         ;..
04CD  1A              LD    A,(DE)        ;.
04CE  77              LD    (HL),A        ;w
04CF  23              INC   HL            ;£
04D0  13              INC   DE            ;.
04D1  13              INC   DE            ;.
04D2  10 F9           DJNZ  £04CD         ;.y
04D4  EF              RST   £28           ;o
04D5  1B 00           DEFB  £1B,£00       ;..
04D7  18 C3           JR    £049C         ;.C
04D9  DF              RST   £18           ;_
04DA  6A              DEFB  £6A           ;j
04DB  18 BF           JR    £049C         ;.?
04DD  CF              RST   £08           ;O
04DE  E6 7F           AND   £7F           ;f.
04E0  FE 0D           CP    £0D           ;^.
04E2  20 B9           JR    NZ,£049D      ; 9
04E4  F7              RST   £30           ;w
04E5  C3 86 06        JP    £0686         ;C..
04E8  DF              RST   £18           ;_
04E9  5F              DEFB  £5F           ;_
04EA  DF              RST   £18           ;_
04EB  5D              DEFB  £5D           ;]
04EC  DF              RST   £18           ;_
04ED  77              DEFB  £77           ;w
04EE  E5              PUSH  HL            ;e
04EF  AF              XOR   A             ;/
04F0  47              LD    B,A           ;G
04F1  DF              RST   £18           ;_
04F2  6F              DEFB  £6F           ;o
04F3  10 FC           DJNZ  £04F1         ;.|
04F5  DF              RST   £18           ;_
04F6  60              DEFB  £60           ;`
04F7  ED 5B 0E 0C     LD    DE,(£0C0E)    ;m[..
04FB  EB              EX    DE,HL         ;k
04FC  37              SCF                 ;7
04FD  ED 52           SBC   HL,DE         ;mR
04FF  DA 8A 06        JP     C,£068A      ;Z..
0502  EB              EX    DE,HL         ;k
0503  AF              XOR   A             ;/
0504  FF              RST   £38           ;.
0505  06 05           LD    B,£05         ;..
0507  DF              RST   £18           ;_
0508  6F              DEFB  £6F           ;o
0509  3E FF           LD    A,£FF         ;>.
```

```
050B  10 FA        DJNZ  £0507        ;.z
050D  AF           XOR   A            ;/
050E  BA           CP    D            ;:
050F  20 02        JR    NZ,£0513     ; .
0511  43           LD    B,E          ;C
0512  04           INC   B            ;.
0513  58           LD    E,B          ;X
0514  7D           LD    A,L          ;}
0515  DF           RST   £18          ;_
0516  6F           DEFB  £6F          ;o
0517  7C           LD    A,H          ;|
0518  DF           RST   £18          ;_
0519  6F           DEFB  £6F          ;o
051A  7B           LD    A,E          ;{
051B  DF           RST   £18          ;_
051C  6F           DEFB  £6F          ;o
051D  7A           LD    A,D          ;z
051E  DF           RST   £18          ;_
051F  6F           DEFB  £6F          ;o
0520  0E 00        LD    C,£00        ;..
0522  DF           RST   £18          ;_
0523  6C           DEFB  £6C          ;l
0524  79           LD    A,C          ;y
0525  DF           RST   £18          ;_
0526  6F           DEFB  £6F          ;o
0527  DF           RST   £18          ;_
0528  6D           DEFB  £6D          ;m
0529  06 0B        LD    B,£0B        ;..
052B  79           LD    A,C          ;y
052C  DF           RST   £18          ;_
052D  6F           DEFB  £6F          ;o
052E  AF           XOR   A            ;/
052F  10 FB        DJNZ  £052C        ;.{
0531  DF           RST   £18          ;_
0532  6A           DEFB  £6A          ;j
0533  18 C2        JR    £04F7        ;.B
0535  B7           OR    A            ;7
0536  ED 52        SBC   HL,DE        ;mR
0538  19           ADD   HL,DE        ;.
0539  30 09        JR    NC,£0544     ;0.
053B  0B           DEC   BC           ;.
053C  EB           EX    DE,HL        ;k
053D  09           ADD   HL,BC        ;.
053E  EB           EX    DE,HL        ;k
053F  09           ADD   HL,BC        ;.
0540  03           INC   BC           ;.
```

```
0541  ED B8          LDDR            ;m8
0543  C9             RET             ;I
0544  ED B0          LDIR            ;m0
0546  C9             RET             ;I
0547  EB             EX    DE,HL     ;k
0548  E5             PUSH  HL        ;e
0549  19             ADD   HL,DE     ;.
054A  DF             RST   £18      ;_
054B  66             DEFB  £66      ;f
054C  E1             POP   HL        ;a
054D  B7             OR    A         ;7
054E  ED 52          SBC   HL,DE     ;mR
0550  DF             RST   £18      ;_
0551  66             DEFB  £66      ;f
0552  2B             DEC   HL        ;+
0553  2B             DEC   HL        ;+
0554  7C             LD    A,H       ;|
0555  FE FF          CP    £FF      ;~.
0557  20 0A          JR    NZ,£0563  ;  .
0559  CB 7D          BIT   7,L       ;K}
055B  20 0D          JR    NZ,£056A  ;  .
055D  EF             RST   £28      ;o
055E  3F 3F          DEFM  /??/      ;??
0560  0D 00          DEFB  £0D,£00  ;..
0562  C9             RET             ;I
0563  B7             OR    A         ;7
0564  20 F7          JR    NZ,£055D  ;  w
0566  CB 7D          BIT   7,L       ;K}
0568  20 F3          JR    NZ,£055D  ;  s
056A  7D             LD    A,L       ;}
056B  DF             RST   £18      ;_
056C  68             DEFB  £68      ;h
056D  C3 11 03       JP    £0311     ;C..
0570  2B             DEC   HL        ;+
0571  3B             DEC   SP        ;;
0572  3B             DEC   SP        ;;
0573  F5             PUSH  AF        ;u
0574  D5             PUSH  DE        ;U
0575  5E             LD    E,(HL)    ;^
0576  7B             LD    A,E       ;{
0577  17             RLA             ;.
0578  9F             SBC   A,A       ;.
0579  57             LD    D,A       ;W
057A  23             INC   HL        ;£
057B  19             ADD   HL,DE     ;.
057C  D1             POP   DE        ;Q
057D  F1             POP   AF        ;q
```

```
057E  E3          EX    (SP),HL      ;c
057F  C9          RET                ;I
0580  2B          DEC   HL           ;+
0581  3B          DEC   SP           ;;
0582  3B          DEC   SP           ;;
0583  F5          PUSH  AF           ;u
0584  D5          PUSH  DE           ;U
0585  5E          LD    E,(HL)       ;^
0586  16 00       LD    D,£00        ;..
0588  2A 71 0C    LD    HL,(£0C71)   ;*q.
058B  19          ADD   HL,DE        ;.
058C  19          ADD   HL,DE        ;.
058D  5E          LD    E,(HL)       ;^
058E  23          INC   HL           ;£
058F  56          LD    D,(HL)       ;V
0590  EB          EX    DE,HL        ;k
0591  18 E9       JR    £057C        ;.i
0593  E5          PUSH  HL           ;e
0594  F5          PUSH  AF           ;u
0595  D5          PUSH  DE           ;U
0596  3A 0A 0C    LD    A,(£0C0A)    ;:..
0599  5F          LD    E,A          ;_
059A  16 00       LD    D,£00        ;..
059C  18 EA       JR    £0588        ;.j
059E  FF          RST   £38          ;.
059F  FF          RST   £38          ;.
05A0  FF          RST   £38          ;.
05A1  FF          RST   £38          ;.
05A2  FF          RST   £38          ;.
05A3  FF          RST   £38          ;.
05A4  FF          RST   £38          ;.
05A5  FF          RST   £38          ;.
05A6  08          EX    AF,AF'       ;.
05A7  FF          RST   £38          ;.
05A8  8E          ADC   A,(HL)       ;.
05A9  FF          RST   £38          ;.
05AA  88          ADC   A,B          ;.
05AB  09          ADD   HL,BC        ;.
05AC  FF          RST   £38          ;.
05AD  FF          RST   £38          ;.
05AE  FF          RST   £38          ;.
05AF  3E 2E       LD    A,£2E        ;>.
05B1  46          LD    B,(HL)       ;F
05B2  36 BE       LD    (HL),£BE     ;6>
05B4  AE          XOR   (HL)         ;.
05B5  0E FF       LD    C,£FF        ;..
```

```
05B7  FF           RST  £38          ;.
05B8  FF           RST  £38          ;.
05B9  89           ADC  A,C          ;.
05BA  FF           RST  £38          ;.
05BB  FF           RST  £38          ;.
05BC  FF           RST  £38          ;.
05BD  FF           RST  £38          ;.
05BE  14           INC  D            ;.
05BF  9C           SBC  A,H          ;.
05C0  9B           SBC  A,E          ;.
05C1  A3           AND  E            ;£
05C2  92           SUB  D            ;.
05C3  C2 BA B2     JP   NZ,£B2BA     ;B:2
05C6  AA           XOR  D            ;*
05C7  A2           AND  D            ;"
05C8  98           SBC  A,B          ;.
05C9  A0           AND  B            ;
05CA  29           ADD  HL,HL        ;)
05CB  0A           LD   A,(BC)       ;.
05CC  21 19 1A     LD   HL,£1A19     ;!..
05CF  1C           INC  E            ;.
05D0  1B           DEC  DE           ;.
05D1  23           INC  HL           ;£
05D2  12           LD   (DE),A       ;.
05D3  42           LD   B,D          ;B
05D4  3A 32 2A     LD   A,(£2A32)    ;:2*
05D7  22 18 20     LD   (£2018),HL   ;".
05DA  A9           XOR  C            ;)
05DB  8A           ADC  A,D          ;.
05DC  A1           AND  C            ;!
05DD  99           SBC  A,C          ;.
05DE  0D           DEC  C            ;.
05DF  2C           INC  L            ;,
05E0  41           LD   B,C          ;A
05E1  13           INC  DE           ;.
05E2  3B           DEC  SP           ;;
05E3  33           INC  SP           ;3
05E4  43           LD   B,E          ;C
05E5  10 40        DJNZ £0627        ;.@
05E7  2D           DEC  L            ;-
05E8  38 30        JR    C,£061A     ;80
05EA  28 31        JR    Z,£061D     ;(1
05EC  39           ADD  HL,SP        ;9
05ED  25           DEC  H            ;%
05EE  1D           DEC  E            ;.
05EF  24           INC  H            ;$
05F0  15           DEC  D            ;.
```

```
05F1  34                  INC   (HL)          ;4
05F2  45                  LD    B,L           ;E
05F3  35                  DEC   (HL)          ;5
05F4  11 2B 44            LD    DE,£442B      ;.+D
05F7  3D                  DEC   A             ;=
05F8  3C                  INC   A             ;<
05F9  1E 9E               LD    E,£9E         ;..
05FB  16 9A               LD    D,£9A         ;..
05FD  96                  SUB   (HL)          ;.
05FE  7D                  LD    A,L           ;}
05FF  32 27 0C            LD    (£0C27),A     ;2'.
0602  C9                  RET                 ;I
0603  22 23 0C            LD    (£0C23),HL    ;"£.
0606  C9                  RET                 ;I
0607  44                  LD    B,H           ;D
0608  4D                  LD    C,L           ;M
0609  ED 59               OUT   (C),E         ;mY
060B  C9                  RET                 ;I
060C  44                  LD    B,H           ;D
060D  4D                  LD    C,L           ;M
060E  ED 78               IN    A,(C)         ;mx
0610  DF                  RST   £18           ;_
0611  68                  DEFB  £68           ;h
0612  C3 11 03            JP    £0311         ;C..
0615  ED 4B 10 0C         LD    BC,(£0C10)    ;mK..
0619  ED 5B 0E 0C         LD    DE,(£0C0E)    ;m[..
061D  2A 0C 0C            LD    HL,(£0C0C)    ;*..
0620  C9                  RET                 ;I
0621  21 7A 07            LD    HL,£077A      ;!z.
0624  DF                  RST   £18           ;_
0625  71                  DEFB  £71           ;q
0626  E5                  PUSH  HL            ;e
0627  21 4C 06            LD    HL,£064C      ;!L.
062A  06 06               LD    B,£06         ;..
062C  7E                  LD    A,(HL)        ;~
062D  F7                  RST   £30           ;w
062E  0E 14               LD    C,£14         ;..
0630  AF                  XOR   A             ;/
0631  FF                  RST   £38           ;.
0632  0D                  DEC   C             ;.
0633  20 FC               JR    NZ,£0631      ; |
0635  23                  INC   HL            ;£
0636  10 F4               DJNZ  £062C         ;.t
0638  DF                  RST   £18           ;_
0639  57                  DEFB  £57           ;W
063A  AF                  XOR   A             ;/
```

```
063B  FF          RST   £38          ;.
063C  3E 45       LD    A,£45        ;>E
063E  F7          RST   £30          ;w
063F  2A 10 0C    LD    HL,(£0C10)   ;*..
0642  DF          RST   £18          ;_
0643  66          DEFB  £66          ;f
0644  3E 0D       LD    A,£0D        ;>.
0646  F7          RST   £30          ;w
0647  E1          POP   HL           ;a
0648  22 73 0C    LD    (£0C73),HL   ;"s.
064B  C9          RET                ;I
064C  0D          DEC   C            ;.
064D  45          LD    B,L          ;E
064E  30 0D       JR    NC,£065D     ;0.
0650  52          LD    D,D          ;R
0651  0D          DEC   C            ;.
0652  0E 00       LD    C,£00        ;..
0654  7E          LD    A,(HL)       ;~
0655  81          ADD   A,C          ;.
0656  4F          LD    C,A          ;O
0657  7E          LD    A,(HL)       ;~
0658  DF          RST   £18          ;_
0659  6F          DEFB  £6F          ;o
065A  23          INC   HL           ;£
065B  10 F7       DJNZ  £0654        ;.w
065D  C9          RET                ;I
065E  DF          RST   £18          ;_
065F  5F          DEFB  £5F          ;_
0660  DF          RST   £18          ;_
0661  77          DEFB  £77          ;w
0662  E5          PUSH  HL           ;e
0663  DF          RST   £18          ;_
0664  78          DEFB  £78          ;x
0665  E5          PUSH  HL           ;e
0666  CF          RST   £08          ;O
0667  FE FF       CP    £FF          ;~.
0669  20 0B       JR    NZ,£0676     ; .
066B  06 03       LD    B,£03        ;..
066D  CF          RST   £08          ;O
066E  FE FF       CP    £FF          ;~.
0670  20 04       JR    NZ,£0676     ; .
0672  10 F9       DJNZ  £066D        ;.y
0674  18 1B       JR    £0691        ;..
0676  FE 1B       CP    £1B          ;~.
0678  20 EC       JR    NZ,£0666     ; l
067A  06 03       LD    B,£03        ;..
067C  CF          RST   £08          ;O
```

```
067D  FE 1B        CP   £1B          ;~.
067F  20 E6        JR   NZ,£0667     ; f
0681  10 F9        DJNZ £067C        ;.y
0683  EF           RST  £28          ;o
0684  18 00        DEFB £18,£00      ;..
0686  E1           POP  HL           ;a
0687  22 75 0C     LD   (£0C75),HL   ;"u.
068A  E1           POP  HL           ;a
068B  22 73 0C     LD   (£0C73),HL   ;"s.
068E  C3 51 00     JP   £0051        ;CQ.
0691  CF           RST  £08          ;O
0692  6F           LD   L,A          ;o
0693  CF           RST  £08          ;O
0694  67           LD   H,A          ;g
0695  CF           RST  £08          ;O
0696  5F           LD   E,A          ;_
0697  CF           RST  £08          ;O
0698  57           LD   D,A          ;W
0699  0E 00        LD   C,£00        ;..
069B  DF           RST  £18          ;_
069C  6C           DEFB £6C          ;l
069D  CF           RST  £08          ;O
069E  B9           CP   C            ;9
069F  20 1E        JR   NZ,£06BF     ; .
06A1  43           LD   B,E          ;C
06A2  0E 00        LD   C,£00        ;..
06A4  3A 2B 0C     LD   A,(£0C2B)    ;:+.
06A7  FE 52        CP   £52          ;~R
06A9  28 03        JR    Z,£06AE     ;(.
06AB  CF           RST  £08          ;O
06AC  18 02        JR   £06B0        ;..
06AE  CF           RST  £08          ;O
06AF  77           LD   (HL),A       ;w
06B0  E5           PUSH HL           ;e
06B1  2A 29 0C     LD   HL,(£0C29)   ;*).
06B4  77           LD   (HL),A       ;w
06B5  E1           POP  HL           ;a
06B6  81           ADD  A,C          ;.
06B7  4F           LD   C,A          ;O
06B8  23           INC  HL           ;£
06B9  10 E9        DJNZ £06A4        ;.i
06BB  CF           RST  £08          ;O
06BC  B9           CP   C            ;9
06BD  28 06        JR    Z,£06C5     ;(.
06BF  EF           RST  £28          ;o
06C0  3F 20        DEFM /? /         ;?
06C2  00           DEFB £00          ;.
```

```
06C3  18 A1        JR    £0666       ;.!
06C5  EF           RST   £28         ;o
06C6  2E 20        DEFM  /. /        ;.
06C8  00           DEFB  £00         ;.
06C9  AF           XOR   A           ;/
06CA  BA           CP    D           ;:
06CB  20 99        JR    NZ,£0666    ;  .
06CD  18 B4        JR    £0683       ;.4
06CF  21 81 07     LD    HL,£0781    ;!..
06D2  DF           RST   £18         ;_
06D3  72           DEFB  £72         ;r
06D4  21 7E 07     LD    HL,£077E    ;!~.
06D7  DF           RST   £18         ;_
06D8  71           DEFB  £71         ;q
06D9  C9           RET               ;I
06DA  7D           LD    A,L         ;}
06DB  32 28 0C     LD    (£0C28),A   ;2(.
06DE  21 85 07     LD    HL,£0785    ;!..
06E1  DF           RST   £18         ;_
06E2  72           DEFB  £72         ;r
06E3  21 7D 07     LD    HL,£077D    ;!}.
06E6  DF           RST   £18         ;_
06E7  71           DEFB  £71         ;q
06E8  C9           RET               ;I
06E9  DF           RST   £18         ;_
06EA  70           DEFB  £70         ;p
06EB  D0           RET   NC          ;P
06EC  E6 7F        AND   £7F         ;f.
06EE  F5           PUSH  AF          ;u
06EF  21 28 0C     LD    HL,£0C28    ;!(.
06F2  CB 6E        BIT   5,(HL)      ;Kn
06F4  CC 21 07     CALL   Z,£0721    ;L!.
06F7  D7           RST   £10         ;W
06F8  20           DEFB  £20         ;
06F9  F1           POP   AF          ;q
06FA  FE 7F        CP    £7F         ;~.
06FC  20 01        JR    NZ,£06FF    ;  .
06FE  AF           XOR   A           ;/
06FF  FE 1B        CP    £1B         ;~.
0701  28 05        JR     Z,£0708    ;(.
0703  B7           OR    A           ;7
0704  28 02        JR     Z,£0708    ;(.
0706  CB FE        SET   7,(HL)      ;K~
0708  37           SCF               ;7
0709  C9           RET               ;I
070A  F5           PUSH  AF          ;u
070B  21 28 0C     LD    HL,£0C28    ;!(.
```

```
070E  CB 7E        BIT  7,(HL)       ;K~
0710  CC 17 07     CALL  Z,£0717     ;L..
0713  CB BE        RES  7,(HL)       ;K>
0715  F1           POP  AF           ;q
0716  C9           RET               ;I
0717  D7           RST  £10          ;W
0718  08           DEFB £08          ;.
0719  FE 0D        CP   £0D          ;~.
071B  C0           RET  NZ           ;@
071C  CB 66        BIT  4,(HL)       ;Kf
071E  C0           RET  NZ           ;@
071F  3E 0A        LD   A,£0A        ;>.
0721  B7           OR   A            ;7
0722  C8           RET   Z           ;H
0723  F5           PUSH AF           ;u
0724  EA 29 07     JP   PE,£0729     ;j).
0727  EE 80        XOR  £80          ;n.
0729  CB 46        BIT  0,(HL)       ;KF
072B  28 02        JR    Z,£072F     ;(.
072D  EE 80        XOR  £80          ;n.
072F  DF           RST  £18          ;_
0730  6F           DEFB £6F          ;o
0731  F1           POP  AF           ;q
0732  C9           RET               ;I
0733  DF           RST  £18          ;_
0734  7B           DEFB £7B          ;{
0735  F7           RST  £30          ;w
0736  18 FB        JR   £0733        ;.{
0738  DF           RST  £18          ;_
0739  78           DEFB £78          ;x
073A  21 7F 07     LD   HL,£077F     ;!..
073D  E5           PUSH HL           ;e
073E  2A 73 0C     LD   HL,(£0C73)   ;*s.
0741  E3           EX   (SP),HL      ;c
0742  22 73 0C     LD   (£0C73),HL   ;"s.
0745  E1           POP  HL           ;a
0746  C9           RET               ;I
0747  21 82 07     LD   HL,£0782     ;!..
074A  E5           PUSH HL           ;e
074B  2A 75 0C     LD   HL,(£0C75)   ;*u.
074E  E3           EX   (SP),HL      ;c
074F  22 75 0C     LD   (£0C75),HL   ;"u.
0752  E1           POP  HL           ;a
0753  C9           RET               ;I
0754  E5           PUSH HL           ;e
0755  21 75 0C     LD   HL,£0C75     ;!u.
0758  18 03        JR   £075D        ;..
```

```
075A  21 73 0C        LD    HL,£0C73     ;!s.
075D  D5              PUSH  DE           ;U
075E  C5              PUSH  BC           ;E
075F  5E              LD    E,(HL)       ;^
0760  23              INC   HL           ;£
0761  56              LD    D,(HL)       ;V
0762  F5              PUSH  AF           ;u
0763  1A              LD    A,(DE)       ;.
0764  13              INC   DE           ;.
0765  B7              OR    A            ;7
0766  28 0D           JR     Z,£0775     ;(.
0768  32 0A 0C        LD    (£0C0A),A    ;2..
076B  F1              POP   AF           ;q
076C  D5              PUSH  DE           ;U
076D  B7              OR    A            ;7
076E  CD 93 05        CALL  £0593        ;M..
0771  D1              POP   DE           ;Q
0772  30 EE           JR    NC,£0762     ;On
0774  F5              PUSH  AF           ;u
0775  F1              POP   AF           ;q
0776  C1              POP   BC           ;A
0777  D1              POP   DE           ;Q
0778  E1              POP   HL           ;a
0779  C9              RET                ;I
```

SUBROUTINE TABLE.

```
0788 47 05 03 06 44 05 0A 03 3A
0790 F4 03 0A 03 21 06 33 07 FC
0798 35 05 FA FF FE 05 94 04 6D
07A0 40 02 38 07 07 06 0A 03 42
07A8 0C 06 5E 06 F9 03 C7 02 EA
07B0 CF 06 5E 06 E8 04 DA 06 BC
07B8 0A 03 FD FF B2 03 93 05 15
07C0 3E 00 45 00 51 00 15 06 B6
07C8 88 00 54 07 B5 02 33 03 9F
07D0 4F 01 00 03 15 03 19 03 5E
07D8 06 03 11 03 0A 03 2D 03 39
07E0 52 06 0A 07 5B 00 81 00 2C
07E8 3D 07 4A 07 5D 07 E9 06 D7
07F0 77 0C 7A 0C 3A 07 47 07 8F
07F8 6C 03 21 03 72 00 34 02
```

OUTPUT TABLES.

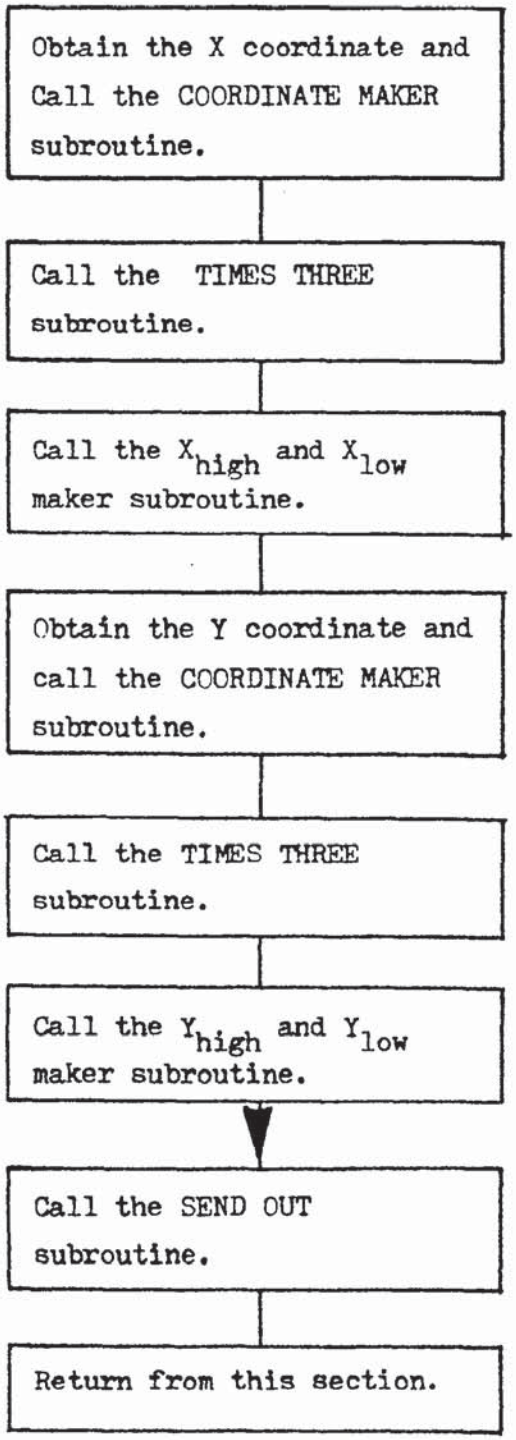
```
077A 65 6F 00 6E 75 65 00
```

INPUT TABLES.

```
0781 76 61 70 00 74 61 00
```

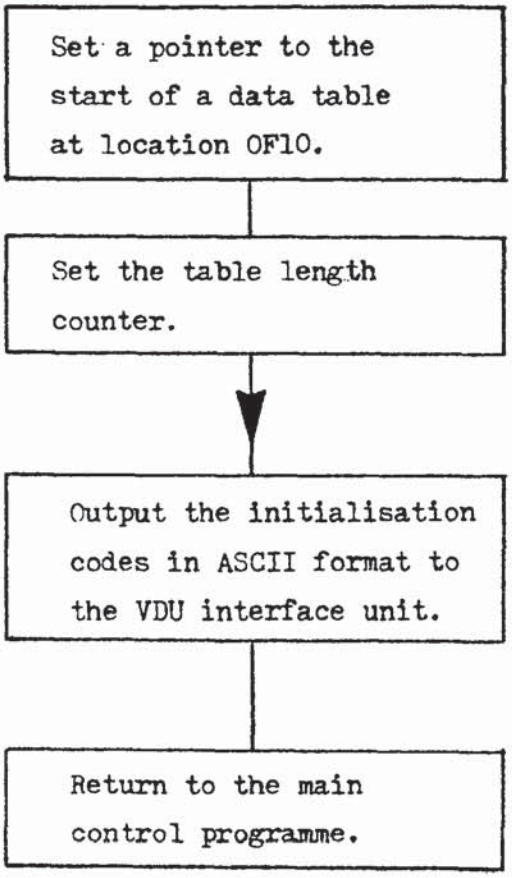
APPENDIX 4. TOOL CONTOUR DISPLAY.

MAIN CONTROL SECTION.



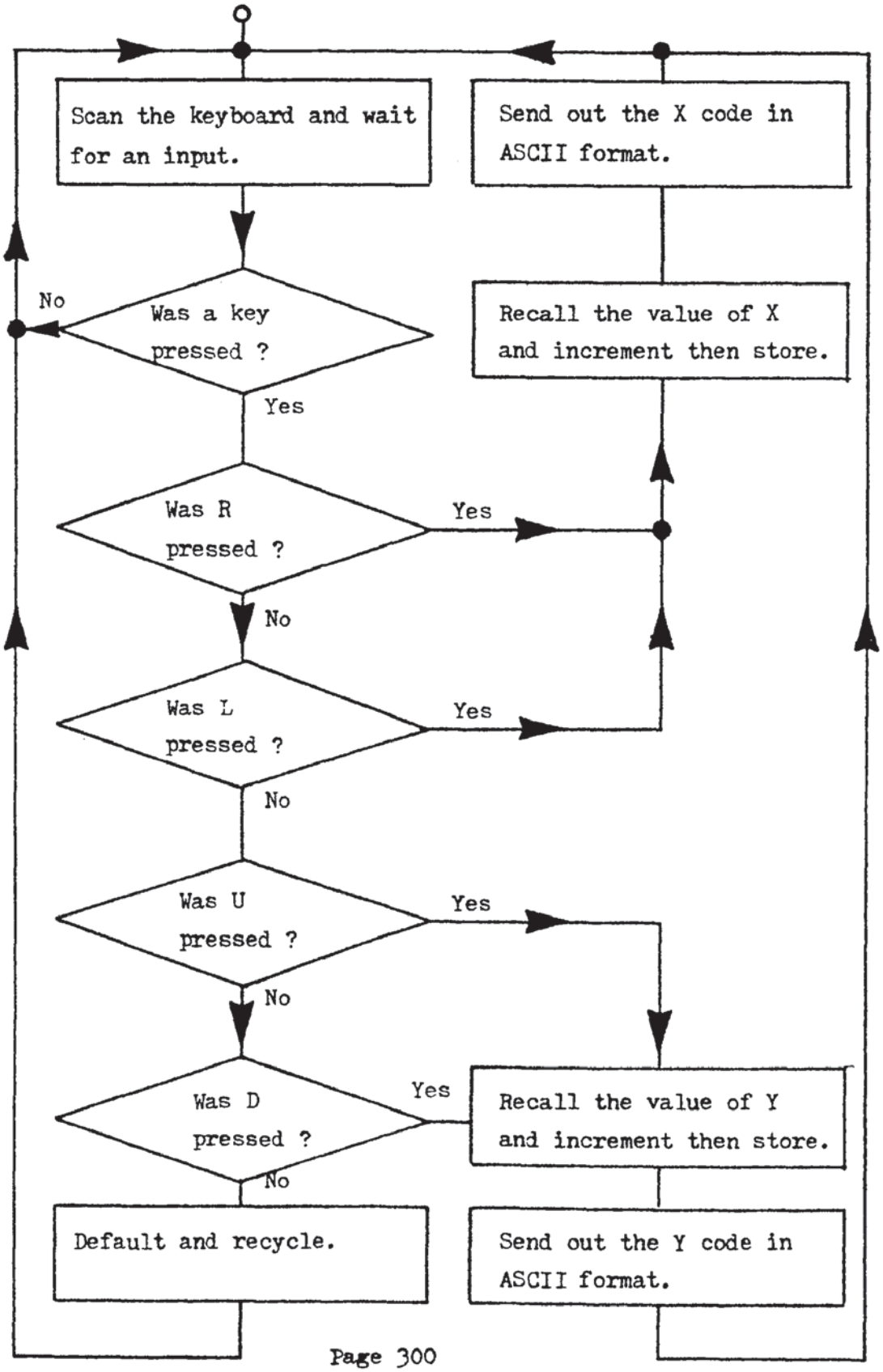
S T A R T O F M A I N P R O G R A M M E.

Initialisation.



```
                0113 ;VDU GRAPHICS CONTROL PROGRAM.
                0114 ;
                0115 ;Assembled by J HOPTON 1982.
                0116 ;
0D7C            0117 INIT   ORG  0D7CH
                0118 ;
                0119 ;START OF MAIN PROGRAM.
                0120 ;
                0121 ;
                0122 ;
0D7C 21100F     0123        LD   HL,0F10H  ;Point to data table.
0D7F 0608       0124        LD   B,08H     ;Table length counter.
0D81 DF         0125        RST  18H       ;Send out initial
0D82 6D         0126        DEFB 6DH       ;data to VDU.
0D83 3E40       0127        LD   A,40H     ;Initial coordinates.
0D85 32070F     0128        LD   (0F07H),A ;Store as Y coord.
0D88 32060F     0129        LD   (0F06H),A ;Store as X coord.
0D8B C3000E     0130        JP   0E00H     ;Continue from 0E00.
                0131 ;
                0132 ;
```

KEYBOARD AND DIRECTION CONTROLLER



```
                0133 ;VDU GRAPHICS CONTROL PROGRAM.
                0134 ;
                0135 ;Assembled by J HOPTON 1982.
                0136 ;
0D95            0137 CONT   ORG  0D95H
                0138 ;
                0139 ;MAIN CONTROL SECTION.
                0140 ;
                0141 ;
                0142 ;
0D95 3A060F     0143        LD   A,(0F06H) ;Get X coord.
0D98 CD000D     0144        CALL 0D00H     ;Call Coord Maker.
0D9B CD580D     0145        CALL 0D58H     ;Call Mult by 3.
0D9E CD3C0D     0146        CALL 0D3CH     ;Call X high/X low.
0DA1 3A070F     0147        LD   A,(0F07H) ;Get Y coord.
0DA4 CD000D     0148        CALL 0D00H     ;Call Coord Maker.
0DA7 CD580D     0149        CALL 0D58H     ;Call mULT by 3.
0DAA CD200D     0150        CALL 0D20H     ;Call Y high/Y low.
0DAD CD710D     0151        CALL 0D71H     ;Call VDU out.
0DB0 C9         0152        RET  ;         ;Return from sub.
                0153 ;
                0154 ;
```

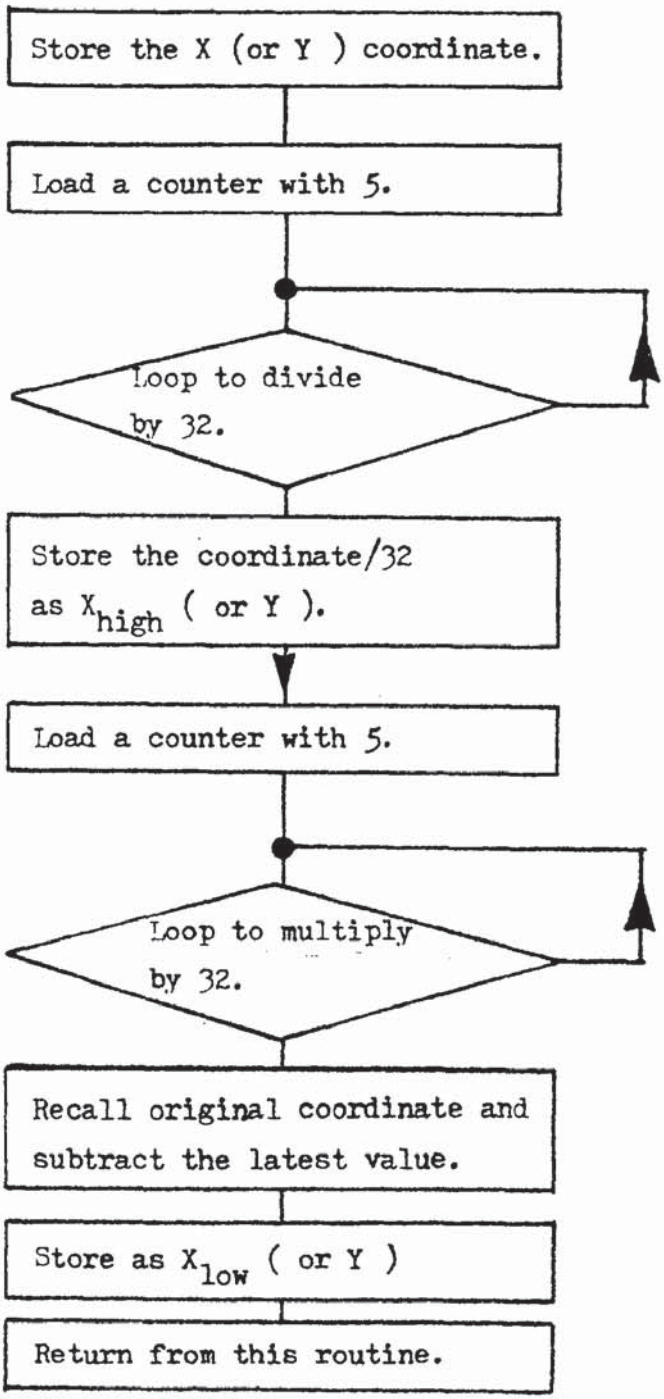
```
                0267 ;VDU GRAPHICS CONTROL PROGRAM.
                0268 ;
                0269 ;Assembled by J HOPTON 1982.
                0270 ;
0F20            0271 KEYBD  ORG  0F20H
                0272 ;
                0273 ;KEYBOARD CONTROLLER.
                0274 ;
0F20 DF         0275        RST  18H        ;Check for Kbd input.
0F21 62         0276        DEFB 62H        ;Set carry if so.
0F22 D22B0F     0277        JP   NC,0F2BH   ;If no input jump.
0F25 32080F     0278        LD   (0F08H),A  ;Store Kbd press.
0F28 C3030E     0279        JP   0E03H      ;Jump to 0E03.
0F2B 00         0280        NOP  ;          ;No operation.
0F2C 00         0281        NOP  ;          ;
0F2D 00         0282        NOP  ;          ;
0F2E 0601       0283        LD   B,01H      ;Set up a delay loop.
0F30 3EFF       0284        LD   A,255      ;Delay loop constant.
0F32 FF         0285        RST  38H        ;Call monitor delay.
0F33 10FB       0286        DJNZ 0F30H      ;Loop until B=0 .
0F35 3A080F     0287        LD   A,(0F08H)  ;Get Kbd press.
0F38 C3030E     0288        JP   0E03H      ;Continue from 0E03.
                0289 ;
```

```
                0176 ;VDU GRAPHICS CONTROL PROGRAM.
                0177 ;
                0178 ;Assembled by J HOPTON 1982.
                0179 ;
0E00            0180 DIRCT  ORG  0E00H
                0181 ;
                0182 ;DIRECTION CONTROLLER.
                0183 ;
                0184 ;
                0185 ;
0E00 C3200F     0186        JP   0F20H      ;Jump to 0F20.
0E03 FE52       0187        CP   52H        ;R Pressed ?
0E05 CA1A0E     0188        JP   Z,0E1AH    ;If so jump to RIGHT.
0E08 FE4C       0189        CP   4CH        ;L Pressed ?
0E0A CA270E     0190        JP   Z,0E27H    ;If so jump to LEFT.
0E0D FE55       0191        CP   55H        ;U Pressed ?
0E0F CA2E0E     0192        JP   Z,0E2EH    ;If so jump to UP.
0E12 FE44       0193        CP   44H        ;D Pressed ?
0E14 CA380E     0194        JP   Z,0E38H    ;If so jump to DOWN.
0E17 C3200F     0195        JP   0F20H      ;Jump to 0F20.
                0196 ;
                0197 ;
```

```
                0221 ;VDU GRAPHICS CONTROL PROGRAM.
                0222 ;
                0223 ;Assembled by J HOPTON 1982.
                0224 ;
0E2E            0225 UP     ORG  0E2EH
                0226 ;
                0227 ;MOVE UPWARDS.
                0228 ;
0E2E 3A070F     0229        LD   A,(0F07H) ;Get Key pressed.
0E31 3C         0230        INC  A         ;Move upwards.
0E32 3A070F     0231        LD   A,(0F07H) ;Store latest position.
0E35 CD950D     0232        CALL 0D95H     ;Call Control.
0E38 C3000E     0233        JP   0E00H     ;Jump to 0E00.
                0234 ;
                0235 ;
0E3B            0236 DOWN   ORG  0E3BH
                0237 ;
                0238 ;MOVE DOWNWARDS.
                0239 ;
0E3B 3A070F     0240        LD   A,(0F07H) ;Get Key pressed.
0E3E 3D         0241        DEC  A         ;Move downwards.
0E3F C3320E     0242        JP   0E32H      ;Jump to 0E32.
                0243 ;
```

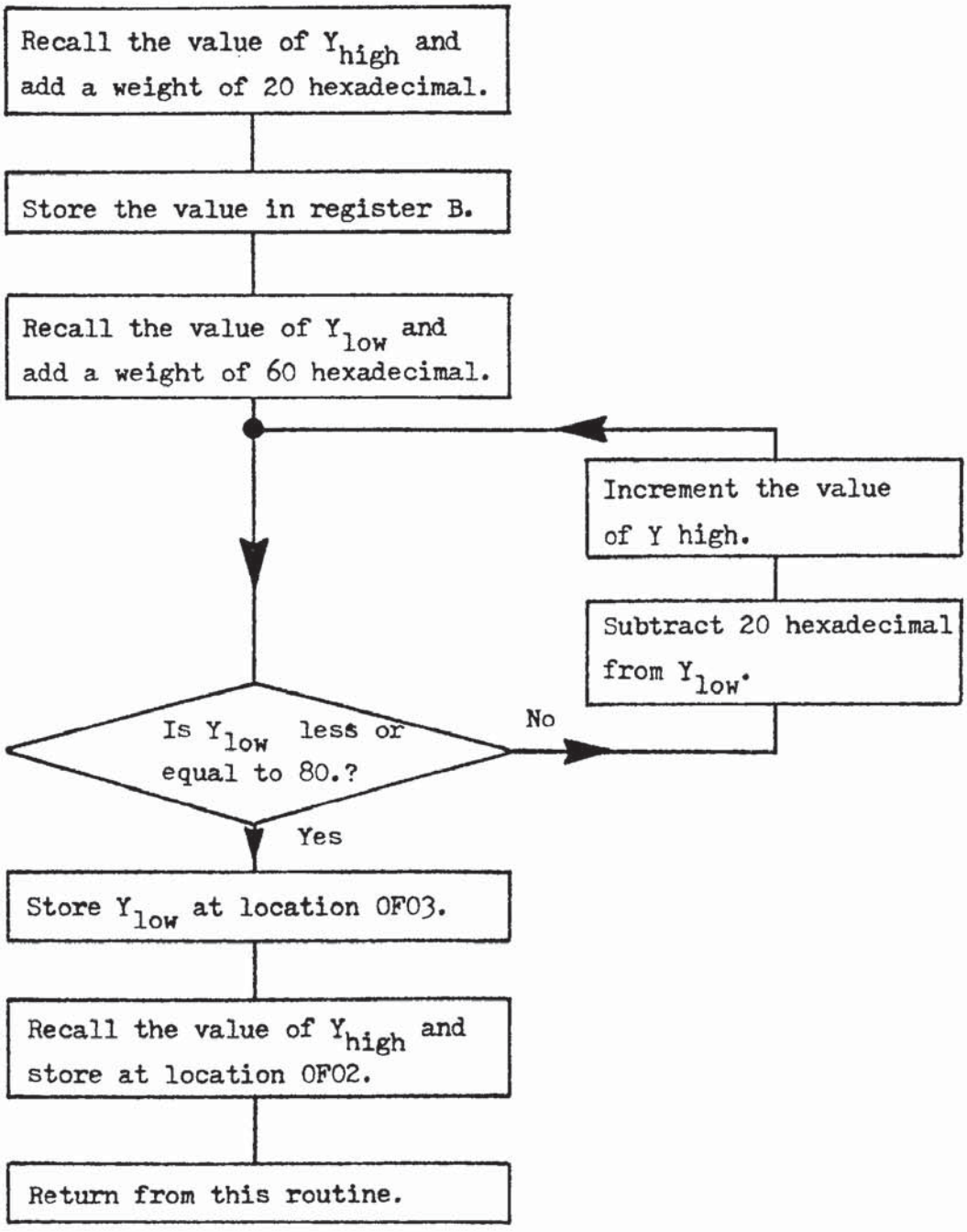
```
                0198 ;VDU GRAPHICS CONTROL PROGRAM.
                0199 ;
                0200 ;Assembled by J HOPTON 1982.
                0201 ;
0E1A            0202 RIGHT  ORG  0E1AH
                0203 ;
                0204 ;MOVE TO THE RIGHT.
                0205 ;
0E1A 3A060F     0206        LD   A,(0F06H) ;Get Key pressed.
0E1D 3C         0207        INC  A         ;Move to right.
0E1E 3A060F     0208        LD   A,(0F06H) ;Store latest position.
0E21 CD950D     0209        CALL 0D95H     ;Call Control.
0E24 C3000E     0210        JP   0E00H     ;Jump to 0E00.
                0211 ;
                0212 ;
0E27            0213 LEFT   ORG  0E27H
                0214 ;
                0215 ;MOVE TO THE LEFT
                0216 ;
0E27 3A060F     0217        LD   A,(0F06H) ;Get Key pressed.
0E2A 3D         0218        DEC  A         ;Move to the left.
0E2B C31E0E     0219        JP   0E1EH      ;Jump to 0E1E.
                0220 ;
```

C O O R D I N A T E M A K E R



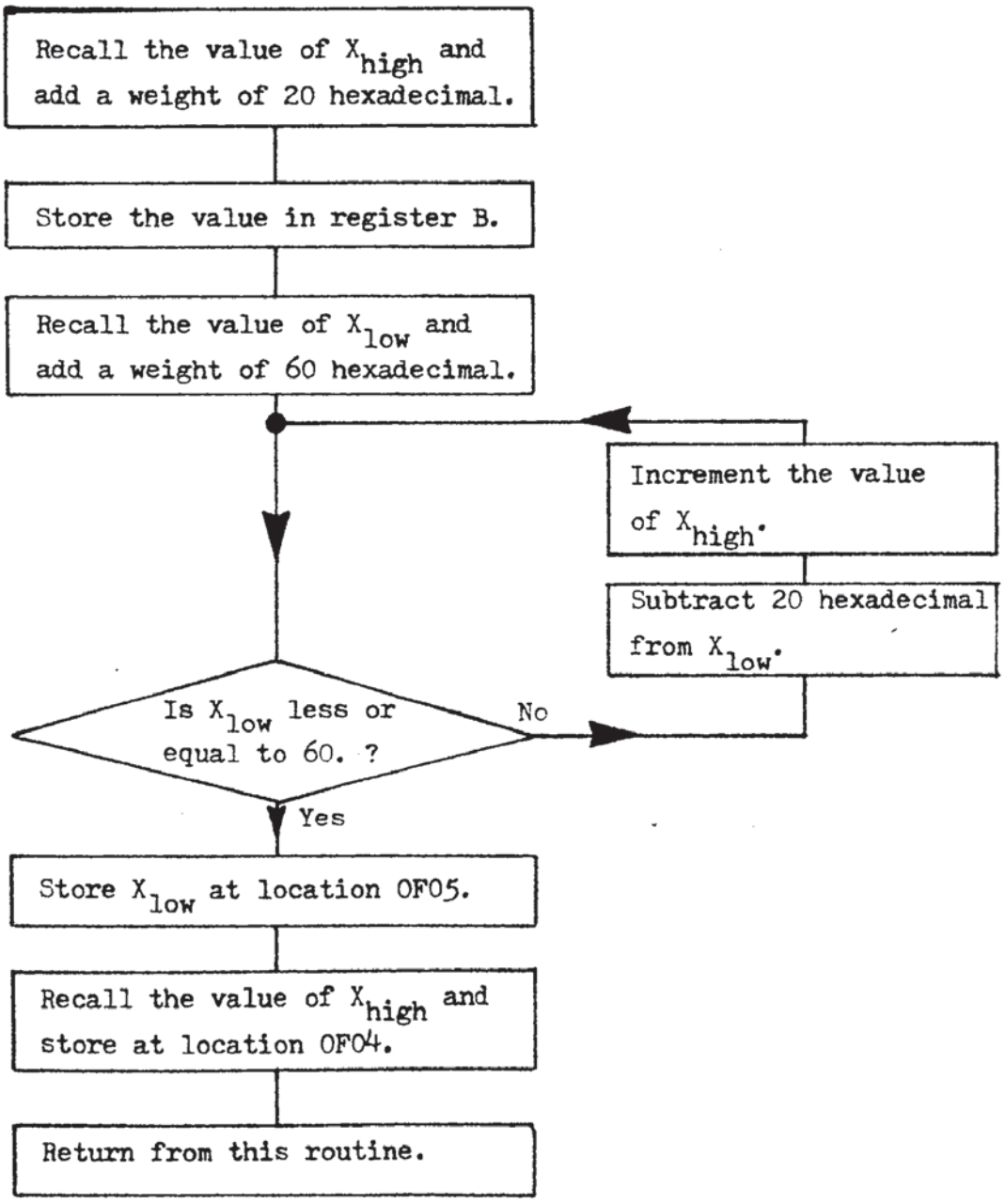
```
                0001 ;VDU GRAPHICS CONTROL PROGRAM.
                0002 ;
                0003 ;Assembled by J HOPTON 1982.
                0004 ;
0D00            0005 COORD  ORG  0D00H
                0006 ;
                0007 ;COORDINATE MAKER.
                0008 ;
0D00 F5         0009        PUSH AF          ;Save value of A.
0D01 0605       0010        LD   B,05H       ;Loop for / by 32.
0D03 B7         0011        OR   A           ;Clear flags.
0D04 1F         0012        RRA  ;           ;Divide by 2.
0D05 10FC       0013        DJNZ 0D03H       ;Loop if B <> 0.
0D07 32000F     0014        LD   (0F00H),A   ;Store X/Y high.
0D0A 0605       0015        LD   B,05H       ;Loop for * 32.
0D0C B7         0016        OR   A           ;Clear flags.
0D0D 17         0017        RLA  ;           ;Multiply by 2.
0D0E 10FC       0018        DJNZ 0D0CH       ;Loop if B <> 0.
0D10 47         0019        LD   B,A         ;Place in B.
0D11 F1         0020        POP  AF          ;Recall original A.
0D12 90         0021        SUB  B           ;Sub (A=A-B).
0D13 32010F     0022        LD   (0F01H),A   ;Store X/Y low.
0D16 C9         0023        RET  ;           ;Return from sub.
```

Y (High) AND Y (Low) MAKER



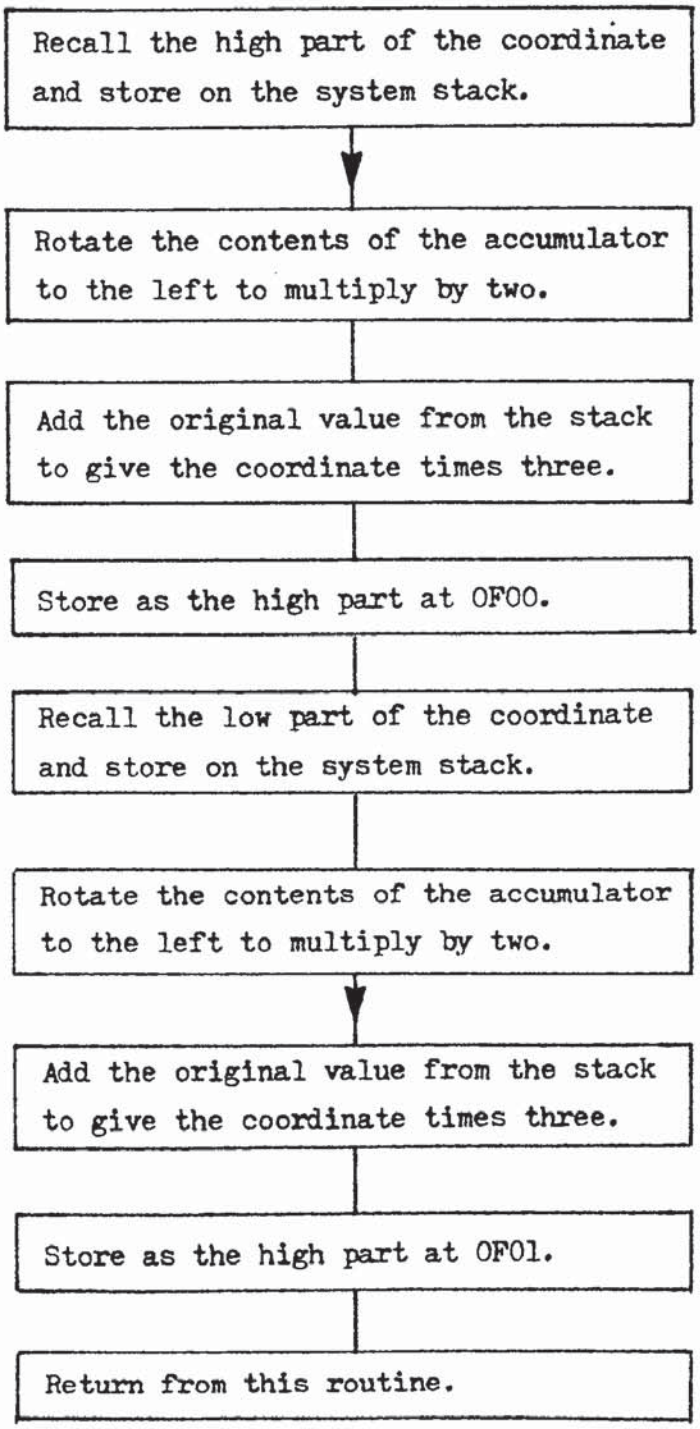
```
                0024 ;VDU GRAPHICS CONTROL PROGRAM.
                0025 ;
                0026 ;Assembled by J HOPTON 1982.
                0027 ;
0D20            0028 VH:VL  ORG  0D20H
                0029 ;
                0030 ;Y (HIGH) Y (LOW) MAKER.
                0031 ;
0D20 3A000F     0032        LD   A,(0F00H)  ;Recall Y high.
0D23 C620       0033        ADD  A,20H      ;Add weight of 20H.
0D25 47         0034        LD   B,A        ;Copy into B.
0D26 3A010F     0035        LD   A,(0F01H)  ;Recall Y low.
0D29 C660       0036        ADD  A,60H      ;Add weight of 60H.
0D2B FE80       0037        CP   80H        ;Is Y low =< 80H ?
0D2D 3805       0038        JR   C,0D34H    ;Jump if not.
0D2F D620       0039        SUB  20H        ;Subtract 20H.
0D31 04         0040        INC  B          ;Inc Y high.
0D32 18F7       0041        JR   0D2BH      ;Jump to 0D2B.
0D34 32030F     0042        LD   (0F03H),A  ;Store Y low.
0D37 78         0043        LD   A,B        ;Recal Y high.
0D38 32020F     0044        LD   (0F02H),A  ;Store Y high.
0D3B C9         0045        RET  ;          ;Return from sub.
                0046 ;
```

X (High) AND X (Low) MAKER



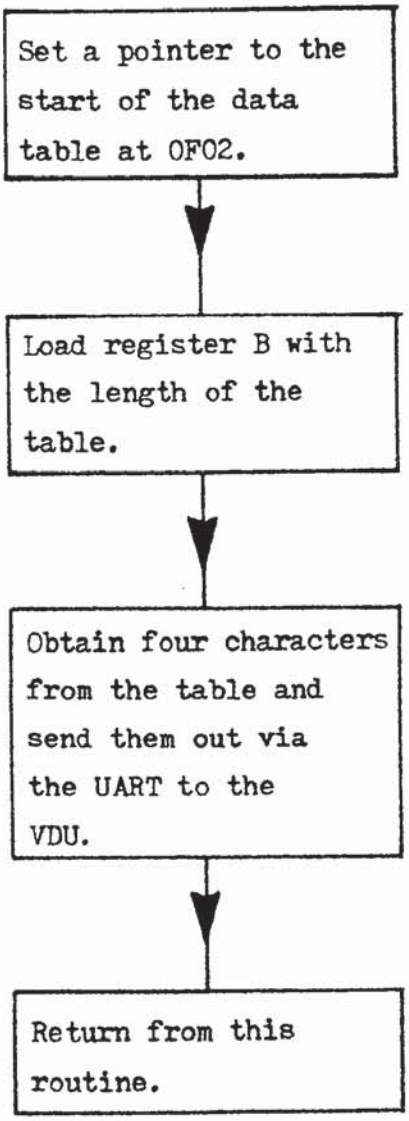
```
                0047 ;VDU GRAPHICS CONTROL PROGRAM.
                0048 ;
                0049 ;Assembled by J HOPTON 1982.
                0050 ;
0D3C            0051 XH:XL  ORG  0D3CH
                0052 ;
                0053 ;X (HIGH) X (LOW) MAKER.
                0054 ;
0D3C 3A000F     0055        LD   A,(0F00H)  ;Recall X high.
0D3F C620       0056        ADD  A,20H      ;Add weight of 20H.
0D41 47         0057        LD   B,A        ;Copy into B.
0D42 3A010F     0058        LD   A,(0F01H)  ;Recall X low.
0D45 C660       0059        ADD  A,60H      ;Add weight of 60H.
0D47 FE60       0060        CP   60H        ;Is X low =< 60H.
0D49 3805       0061        JR   C,0D50H    ;Jump if not.
0D4B D620       0062        SUB  20H        ;Subtract 20H.
0D4D 04         0063        INC  B          ;Increment X high.
0D4E 18F7       0064        JR   0D47H      ;Jump to 0D47.
0D50 32050F     0065        LD   (0F05H),A  ;Store X low.
0D53 78         0066        LD   A,B        ;Recal X high.
0D54 32040F     0067        LD   (0F04H),A  ;Store X high.
0D57 C9         0068        RET  ;          ;Return from sub.
                0069 ;
```

MULTIPLICATION BY THREE



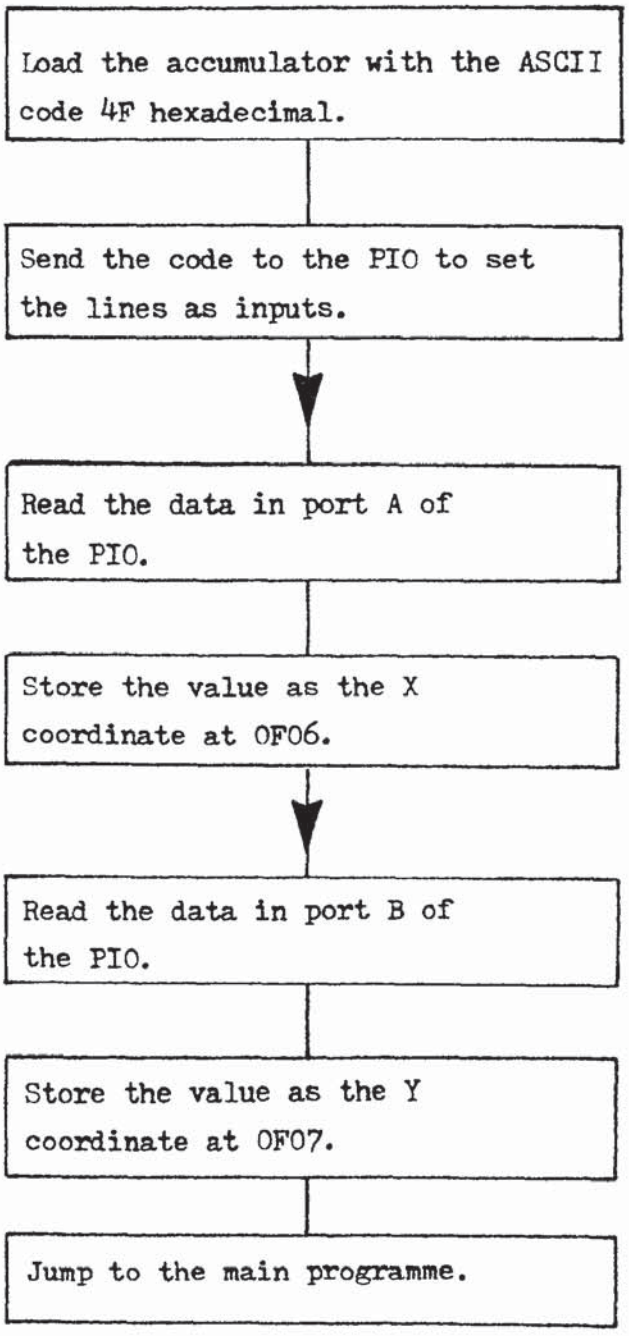
```
                0070 ;UDU GRAPHICS CONTROL PROGRAM.
                0071 ;
                0072 ;Assembled by J HOPTON 1982.
                0073 ;
0D58            0074 MAG:3  ORG  0D58H
                0075 ;
                0076 ;MAGNIFY BY A FACTOR OF 3.
                0077 ;
0D58 3A000F     0078        LD   A,(0F00H)  ;Recall X/Y high.
0D5B F5         0079        PUSH AF         ;Store  X/Y high.
0D5C B7         0080        OR   A          ;Clear flags.
0D5D 17         0081        RLA  ;          ;Mult by 2.
0D5E 47         0082        LD   B,A        ;Copy into B.
0D5F F1         0083        POP  AF         ;Recall X/Y high.
0D60 80         0084        ADD  A,B        ;Add 2X/Y and X/Y.
0D61 32000F     0085        LD   (0F00H),A  ;Store  X/Y  high.
0D64 3A010F     0086        LD   A,(0F01H)  ;Recall X/Y low.
0D67 F5         0087        PUSH AF         ;Store  X/Y low.
0D68 B7         0088        OR   A          ;Clear flags.
0D69 17         0089        RLA  ;          ;Mult by 2.
0D6A 47         0090        LD   B,A        ;Copy into B.
0D6B F1         0091        POP  AF         ;Recall X/Y low.
0D6C 80         0092        ADD  A,B        ;Add 2X/Y and X/Y.
0D6D 32010F     0093        LD   (0F01H),A  ;Store X/Y low.
0D70 C9         0094        RET  ;          ;Return from sub.
                0095 ;
```

OUTPUT CHARACTER TO VDU



```
                0096 ;VDU GRAPHICS CONTROL PROGRAM.
                0097 ;
                0098 ;Assembled by J HOPTON 1982.
                0099 ;
0D71            0100 OUT    ORG  0D71H
                0101 ;
                0102 ;SEND CHARACTER TO VDU.
                0103 ;
                0104 ;
                0105 ;
0D71 21020F     0106        LD   HL,0F02H  ;Point to table.
0D74 0604       0107        LD   B,04H     ;Table length.
0D76 DF         0108        RST  18H       ;Send out to VDU.
0D77 6D         0109        DEFB 6DH       ;
0D78 C9         0110        RET  ;         ;Return from sub.
                0111 ;
                0112 ;
```

R E A D T H E I N T E R F A C E B O A R D



```
                0155 ;VDU GRAPHICS CONTROL PROGRAM.
                0156 ;
                0157 ;Assembled by J HOPTON 1982.
                0158 ;
0DC0            0159 READ   ORG  0DC0H
                0160 ;
                0161 ;READ INTERFACE BOARD.
                0162 ;
                0163 ;
                0164 ;
0DC0 3E4F       0165        LD   A,4FH      ;Set PIO lines
0DC2 D307       0166        OUT  (07H),A    ;as inputs.
0DC4 D306       0167        OUT  (06H),A    ;
0DC6 DB04       0168        IN   A,(04H)    ;Read Port A.
0DC8 32060F     0169        LD   (0F06H),A  ;Store X coord.
0DCB DB05       0170        IN   A,(05H)    ;Read Port B.
0DCD 32070F     0171        LD   (0F07H),A  ;Store Y coorD.
0DD0 CD950D     0172        CALL 0D95H      ;Call main prog.
0DD3 C3C60D     0173        JP   0DC6H      ;Jump to 0DC6.
                0174 ;
                0175 ;
```

```
                0244 ;VDU GRAPHICS CONTROL PROGRAM.
                0245 ;
                0246 ;Assembled by J HOPTON 1982.
                0247 ;
0F00            0248 STORE  ORG  0F00H
                0249 ;
                0250 ;SYSTEM STORAGE LOCATIONS.
                0251 ;
                0252 ;0F00 TO 0F08 is used by the system for
                0253 ;storing program variables.
                0254 ;
                0255 ;
0F10            0256 VECT   ORG  0F10H
                0257 ;
                0258 ;VDU INITIALISATION VECTORS.
                0259 ;
                0260 ;19            = Clear screen.
                0261 ;1D            = Vector Mode.
                0262 ;30 70         = Y Coordinates.
                0263 ;30 50         = X Coordinates.
                0264 ;1D            = Vector Mode.
                0265 ;1D            = Vector Mode.
                0266 ;
```

APPENDIX 5. BEZIER SURFACE GENERATION.

```
MON
HOPSYSTEM
```

```
10 REMBEZIERSURFMK3FULLMEM
11 CLEAR:D=3330:E=3328
15 GOSUB1220:REMINMC/BORDER
20 GOTO520:REMMAIN
30 REM:INPUT"NBYM";N,M
32 PRINT"32 Inp Cond":N=3:M=3:P=7
40 REMINPUT"GRIDS";P
50 A=P*P:B=(P-1)^2
60 DIM S(3,A),PX(B,P),PY(B,P),PZ(B,P),PC(B,2)
62 DIMX(A,B),Y(A,B),Z(A,B):RESTORE3000
66 PRINT"66 Coord Inp"
70 FORA=1TON+1:FORB=1TOM+1
75 READX(A,B),Y(A,B),Z(A,B)
80 REM:INPUT"CordsX,Y,Z";X(A,B),Y(A,B),Z(A,B)
90 NEXT:NEXT
110 FORC=1TO3:F=0:ONCGOTO120,140,160
120 PRINT"120 X Cords":FORA=1TON+1:FORB=1TOM+1
130 B(A,B)=X(A,B):NEXT:NEXT:GOTO185
140 PRINT"140 Y Cords":FORA=1TON+1:FORB=1TOM+1
150 B(A,B)=Y(A,B):NEXT:NEXT:GOTO185
160 PRINT"160 Z Cords":FORA=1TON+1
170 FORB=1TOM+1:B(A,B)=Z(A,B):NEXT:NEXT
185 PRINT"185 Step C=";C
190 FORU=0TO1STEP1/(P-1):FORW=0TO1STEP1/(P-1)
200 F=F+1:FORI=0TON:X=N:GOSUB460:G=A
210 X=I:GOSUB460:H=A:X=N-I:GOSUB460:L=A
230 Z=(G/(H*L))*U^I*(1-U)^(N-I):FORJ=0TOM
240 X=M:GOSUB460:G=A:X=J:GOSUB460:H=A:X=M-J
250 GOSUB460:L=A
270 K=(G/(H*L))*W^J*(1-W)^(M-J)
280 REMSTOREVERTS
290 S(C,F)=S(C,F)+B(I+1,J+1)*Z*K
300 NEXT:NEXTI:NEXTW:NEXTU:NEXTC:RETURN
460 IFX=0THENA=1:RETURN
470 A=1
480 A=A*X:X=X-1
490 IFX=0THENRETURN
500 GOTO480
520 GOSUB30
530 PRINT"530Fin Surf Gen":GOSUB630
550 PRINT"550Fin Poly Def":GOSUB740
570 PRINT"570Fin Priority Order":GOSUB820
590 PRINT"590Fin Dispy"
605 DOKE4100,-16872:A=USR(1):REMPrintBuffer
610 END
612 REM
614 REM
```

```
616 REM
618 REM
620 PRINT"620Def Polys"
630 C=0:Z=0:FORA=1TOP-1:FORB=1TOP-1
640 Z=Z+1:PX(Z,1)=S(1,C+B):PY(Z,1)=S(2,C+B)
650 PZ(Z,1)=S(3,C+B):PX(Z,2)=S(1,C+B+1)
660 PY(Z,2)=S(2,C+B+1):PZ(Z,2)=S(3,C+B+1)
670 PX(Z,3)=S(1,C+B+1+P):PY(Z,3)=S(2,C+B+1+P)
680 PZ(Z,3)=S(3,C+B+1+P):PX(Z,4)=S(1,C+B+P)
690 PY(Z,4)=S(2,C+B+P):PZ(Z,4)=S(3,C+B+P)
700 F=1000:FORK=1TO4
710 IFPZ(Z,K)<FTHENF=PZ(Z,K):NEXT
720 PC(Z,1)=F:PC(Z,2)=Z:NEXTB:C=C+P:NEXTA
730 RETURN
740 PRINT"740Sort Vert"
750 FORB=1TOZ-1:A=0:FORC=1TOZ-B
760 IFPC(C+1,1)<=PC(C,1)THEN790
770 F=PC(C,1):K=PC(C,2):PC(C,1)=PC(C+1,1)
780 PC(C,2)=PC(C+1,2):PC(C+1,1)=F:PC(C+1,2)=K:A
=1
790 NEXT:IFA=0THENRETURN
801 NEXT:RETURN
820 PRINT"810Disp Surf C=":GOSUB4000
825 FORC=1TOZ:PRINTC;"of";Z:F=PC(C,2)
830 I=PX(F,1):U=PY(F,1):FORJ=2TO4:K=PX(F,J)
850 W=PY(F,J):GOSUB1910:I=PX(F,J):U=PY(F,J)
870 NEXT:K=PX(F,1):W=PY(F,1):GOSUB1910
875 NEXT:RETURN
900 REM
910 REM
920 REM
930 REM
1210 REM M/C INS
1220 IFDEEK(-16896)=3902THENGOTO1270
1230 PRINT"Ent M/C at BE00H"
1260 FORA=0TO188:READB:DOKE-16896+2*A,B:NEXT
1270 PRINT:PRINT"M/C ENTERED"
1280 DOKE4100,-16896:A=USR(1):REM INIT PIO
1290 DOKE4100,-16813:REM Markers in Buffer
1300 FORA=20TO220STEP20:DOKED,A:DOKEE,4
1320 B=USR(1):DOKED,3:DOKEE,A:B=USR(1)
1330 DOKED,A:DOKEE,223:B=USR(1):DOKED,219
1340 DOKEE,A:B=USR(1):NEXT
1350 POKE27267,1:POKE27483,1:RETURN
1400 REM
1410 REM
```

```
1420 REM
1430 REM
1500 DATA 3902,2003,-194,1747,1235,-1730,1747
1510 DATA-18399,8894,3192,-4659,-13890,8454,33
1530 DATA4512,-24353,16162,-4673,16979,-14913
1540 DATA16077,1727,16127,-13056,-16712,-1776
1550 DATA27075,10943,-16577,8729,-16577,16938
1560 DATA6591,16930,-15937,-8432,3390,-18227
1580 DATA16062,-13046,-16712,-8759,33,4352,224
1590 DATA33,-4864,75,2829,2827,2827,2827,9181
1610 DATA30745,28926,-3784,11229,-4681,-5294,33
1620 DATA-18500,21229,23533,3330,-8935,1058
1630 DATA-8947,-8919,-8919,-8919,1570,8717,3336
1650 DATA1578,-4851,1115,-18675,21229,23533
1660 DATA3328,-18453,21229,15741,-13385,-13529
1670 DATA-13529,-14809,12998,-16714,2090
1680 DATA-13555,-13866,-13323,10367,-13538
1690 DATA8311,15876,6176,-10730,20256,7910
1700 DATA16331,18961,-29824,31327,206,6743
1710 DATA31063,-7706,-11342,-9467,-13564
1720 DATA8263,16122,-11267,15876,-11265,-3836
1730 DATA16073,12800,-24352,-8159,4512,-24351
1750 DATA8193,-4837,16048,12896,-24576,33
1760 DATA4512,-24575,-8447,-4864,16048,12803
1770 DATA-17408,33,4540,-17407,-8447,-4864
1780 DATA1712,15905,8575,-24576,-8431,30464
1790 DATA30499,4121,1786,15905,8575,-24355
1810 DATA-8431,30464,30499,4121,-13830,33
1820 DATA4512,-24095,6974,-18227,16062
1830 DATA-13005,-16712,4158,-18227,16062
1840 DATA-13029,-16712,13886,-18227,32446
1850 DATA-13021,-16712,-4681,6482,6352,1780
1860 DATA15875,-13056,-16712,-1776
1870 DATA-8175,-15616,-16841,0,255
1880 REM
1885 REM
1888 REM
```

```
1889 REM
1900 REM HPLOT(St Line I,U to K,W)
1910 X=K-I:Y=W-U:L=ABS(X):G=ABS(Y)
1930 IFG>LTHENL=G
1935 IFL=0THENRETURN
1940 X=X/L:Y=Y/L:V=I+0.5-X:H=U+0.5-Y
1980 FORA=0TOL:V=V+X:H=H+Y
1990 DOKED,INT(V):DOKEE,INT(H):B=USR(1)
2000 NEXT:RETURN
2010 REM
2020 REM
2030 REM
2040 REM
3000 DATA 120,  0,120,120,120,120,  0,120,120
3010 DATA   0,  0,120, 80, 40, 80, 80, 80, 80
3020 DATA  40, 80, 80, 40, 40, 80, 80, 40, 40
3030 DATA  80, 80, 40, 40, 80, 40, 40, 40, 40
3040 DATA 120,  0,  0,120,120,  0,  0,120,  0
3050 DATA   0,  0,  0
3090 REM
3095 REM
3096 REM
3097 REM
4000 F=0:W=0:V=0:FORA=1TOZ:FORB=1TOP
4010 IFPX(A,B)>FTHENF=PX(A,B)
4012 IFPY(A,B)>WTHENW=PY(A,B)
4014 IFPZ(A,B)>VTHENV=PZ(A,B)
4025 NEXT:NEXT:U=F:IFW>FTHENU=W
4026 IFV>UTHENU=V
4030 FORA=1TOZ:FORB=1TOP
4040 C=(U*(2+N)+PZ(A,B)/2-PX(A,B)*0.866026)/2
4050 PX(A,B)=(PX(A,B)/2+PZ(A,B)*0.866026)*U/C
4060 PY(A,B)=224+(PY(A,B)-1.5*U)*U/C:NEXT:NEXT
4105 F=0:X=900:W=0:Y=900:FORA=1TOZ:FORB=1TOP
4110 IFPX(A,B)>FTHENF=PX(A,B)
4112 IFPX(A,B)<XTHENX=PX(A,B)
4120 IFPY(A,B)>WTHENW=PY(A,B)
4122 IFPY(A,B)<YTHENY=PY(A,B)
4135 NEXT:NEXT:I=F-X:J=W-Y:FORA=1TOZ
4150 FORB=1TOP:PX(A,B)=16+(PX(A,B)-X)*192/I
4160 PY(A,B)=16+(PY(A,B)-Y)*192/J
4210 NEXT:NEXT:PRINTX;F,Y;W:RETURN
Ok
```

```
4E00            0001 START  ORG  4E00H
                0002 ;
                0003 ;Assembled by J HOPTON. Aug 82.
                0004 ;
                0005 ;NOTE The program runs at BE00H.
                0006 ;But is assembled at 4E00H.
                0007 ;
                0008 ;** I N I T I A L I S E   P I O **
                0009 ;
4E00 0C78       0010 L0C78  EQU  £0C78
                0011 ;
                0012 ;
4E00 3E0F       0013        LD   A,£0F      ;Set PIO lines
4E02 D307       0014        OUT  (£07),A    ;for Printer.
4E04 3EFF       0015        LD   A,£FF
4E06 D306       0016        OUT  (£06),A
4E08 D304       0017        OUT  (£04),A
4E0A 3EF9       0018        LD   A,£F9
4E0C D306       0019        OUT  (£06),A
                0020 ;
                0021 ;
                0022 ;
                0023 ;
                0024 ;
                0025 ;
```

```
                0026 ;
                0027 ;
                0028 ;
                0029 ;
4E0E 21B8BE     0030        LD    HL,LBEB8    ;Load Print
4E11 22780C     0031        LD    (LOC78),HL  ;address.
4E14 CDEDBE     0032        CALL  LBEED       ;Clear Buffer
4E17 C9         0033        RET   ;           ;Retn from Sub
4E18 0621       0034        LD    B,£21
                0035 ;
                0036 ;
4E1A BEB8       0037 LBEB8  EQU   £BEB8
4E1A BEED       0038 LBEED  EQU   £BEED
```

```
4E18                0001 START  ORG  4E18H
                    0002 ;
                    0003 ;Assembled by J HOPTON. AUG 82.
                    0004 ;
                    0005 ;NOTE the Program runs at BE18H.
                    0006 ;But is assembled at 4E18H.
                    0007 ;
                    0008 ;    *** P R I N T   B U F F E R ***
                    0009 ;
4E18 A000           0010 LA000  EQU  £A000
4E18 A0DF           0011 LA0DF  EQU  £A0DF
4E18 0621           0012        LD   B,£21        ;33 Lines.
4E1A 2100A0         0013        LD   HL,LA000     ;Buff Start.
4E1D 11DFA0         0014        LD   DE,LA0DF     ;Buff Right.
4E20 223FBF         0015        LD   (LBF3F),HL ;Line Space.
4E23 ED5342BF       0016        LD   (LBF42),DE
4E27 C5             0017 LBE27  PUSH BC
4E28 CD3EBF         0018        CALL LBF3E        ;Line Space.
4E2B 06FF           0019        LD   B,£FF
4E2D 3E00           0020 LBE2D  LD   A,£00        ;Buff Nulls.
                    0021 ;
                    0022 ;
                    0023 ;
                    0024 ;
                    0025 ;
```

```
                0026 ;
                0027 ;
                0028 ;
                0029 ;
4E2F CDB8BE     0030            CALL LBEB8      ;Print  Out.
4E32 10F9       0031            DJNZ LBE2D
4E34 C369BF     0032            JP   LBF69
4E37 2A3FBF     0033            LD   HL,(LBF3F)
4E3A 19         0034            ADD  HL,DE
4E3B 223FBF     0035            LD   (LBF3F),HL
4E3E 2A42BF     0036            LD   HL,(LBF42)
4E41 19         0037            ADD  HL,DE
4E42 2242BF     0038            LD   (LBF42),HL
4E45 C1         0039            POP  BC
4E46 10DF       0040            DJNZ LBE27
4E48 3E0D       0041            LD   A,£0D      ;Carriage
4E4A CDB8BE     0042            CALL LBEB8      ;LF  and
4E4D 3E0A       0043            LD   A,£0A      ;Retn.
4E4F CDB8BE     0044            CALL LBEB8
4E52 C9         0045            RET
4E53 BEB8       0046 LBEB8      EQU  £BEB8
4E53 BF3E       0047 LBF3E      EQU  £BF3E
4E53 BF3F       0048 LBF3F      EQU  £BF3F
4E53 BF42       0049 LBF42      EQU  £BF42
4E53 BF69       0050 LBF69      EQU  £BF69
```

```
4E53            0001 START  ORG  4E53H
                0002 ;
                0003 ;Assembled by J HOPTON. Aug 82.
                0004 ;
                0005 ;NOTE The program runs at BE53H.
                0006 ;But is assembled at 4E53H.
                0007 ;
                0008 ;** D O T   I N   B U F F E R **
                0009 ;
4E53 0000       0010 L0000  EQU  £0000
4E53 00E0       0011 L00E0  EQU  £00E0
4E53 0D00       0012 L0D00  EQU  £0D00
4E53 0D02       0013 L0D02  EQU  £0D02
4E53 0D04       0014 L0D04  EQU  £0D04
4E53 0D06       0015 L0D06  EQU  £0D06
4E53 0D08       0016 L0D08  EQU  £0D08
4E53 BC00       0017 LBC00  EQU  £BC00
4E53 DD210000   0018         LD   IX,L0000
                0019 ;
                0020 ;
                0021 ;
                0022 ;
                0023 ;
                0024 ;
```

```
                ---- ;
                0029 ;
4E57 11E000     0030          LD    DE,L00E0    ;Buff Width.
4E5A 210000     0031          LD    HL,L0000
4E5D ED4B000D   0032          LD    BC,(L0D00)
4E61 0B         0033 LBE61    DEC   BC
4E62 0B         0034          DEC   BC
4E63 0B         0035          DEC   BC
4E64 0B         0036          DEC   BC
4E65 0B         0037          DEC   BC
4E66 0B         0038          DEC   BC
4E67 0B         0039          DEC   BC
4E68 DD23       0040          INC   IX
4E6A 19         0041          ADD   HL,DE
4E6B 78         0042          LD    A,B
4E6C FE70       0043          CP    £70
4E6E 38F1       0044          JR    C,LBE61
4E70 DD2B       0045          DEC   IX
4E72 B7         0046          OR    A
4E73 ED52       0047          SBC   HL,DE
4E75 EB         0048          EX    DE,HL
                0049 ;
                0050 ;
```

```
                0058 ;
                0059 ;
4E76 2100BC     0060            LD    HL,LBC00    ;Last line of
4E79 B7         0061            OR    A           ;Buffer.
4E7A ED52       0062            SBC   HL,DE
4E7C ED5B020D   0063            LD    DE,(LOD02)
4E80 19         0064            ADD   HL,DE
4E81 DD22040D   0065            LD    (LOD04),IX
4E85 DD29       0066            ADD   IX,IX
4E87 DD29       0067            ADD   IX,IX
4E89 DD29       0068            ADD   IX,IX
4E8B DD22060D   0069            LD    (LOD06),IX
4E8F 22080D     0070            LD    (LOD08),HL
4E92 2A060D     0071            LD    HL,(LOD06)
4E95 ED5B040D   0072            LD    DE,(LOD04)
4E99 B7         0073            OR    A
4E9A ED52       0074            SBC   HL,DE
4E9C ED5B000D   0075            LD    DE,(LOD00)
4EA0 EB         0076            EX    DE,HL
4EA1 B7         0077            OR    A
4EA2 ED52       0078            SBC   HL,DE
                0079 ;
                0080 -
```

```
                0086 ;
                0087 ;
                0088 ;
                0089 ;
4EA4 7D         0090          LD    A,L
4EA5 3D         0091          DEC   A
4EA6 B7         0092          OR    A
4EA7 CB27       0093          SLA   A             ;Set Correct
4EA9 CB27       0094          SLA   A             ;bit in
4EAB CB27       0095          SLA   A             ;display
4EAD C6C6       0096          ADD   A,£C6         ;Buffer.
4EAF 32B64E     0097          LD    (LBEB6),A
4EB2 2A080D     0098          LD    HL,(LOD08)
4EB5 CBC6       0099          SET   0,(HL)        ;Auto Set.
4EB7 4EB6       0100 LBEB6    EQU   $-1
4EB7 C9         0101          RET   ;             ;Return.
                0102 ;
                0103 ;
                0104 ;
                0105 ;
                0106 ;
```

```
4EB8            0001 START  ORG  4EB8H
                0002 ;
                0003 ;Assembled by J HOPTON. Aug 82.
                0004 ;
                0005 ;NOTE The program runs at BEB8H.
                0006 ;But is assembled at 4EB8H.
                0007 ;
                0008 ; *** P R I N T   O U T ***
                0009 ;
4EB8 804A       0010 L804A  EQU  £804A
4EB8 F5         0011        PUSH AF
4EB9 CB7F       0012        BIT  7,A
4EBB 281E       0013        JR   Z,LBEDB
4EBD CB77       0014        BIT  6,A
4EBF 2004       0015        JR   NZ,LBEC5
4EC1 3E20       0016        LD   A,£20
4EC3 1816       0017        JR   LBEDB
4EC5 D620       0018 LBEC5  SUB  £20
4EC7 4F         0019        LD   C,A
4EC8 E61E       0020        AND  £1E
                0021 ;
                0022 ;
                0023 ;
                0024 ;
```

```
                    0029 ;
4ECA CB3F           0030            SRL   A
4ECC 114A80         0031            LD    DE,L804A
4ECF 8B             0032            ADC   A,E
4ED0 5F             0033            LD    E,A
4ED1 7A             0034            LD    A,D
4ED2 CE00           0035            ADC   A,£00
4ED4 57             0036            LD    D,A
4ED5 1A             0037            LD    A,(DE)
4ED6 57             0038            LD    D,A
4ED7 79             0039            LD    A,C
4ED8 E6E1           0040            AND   £E1
4EDA B2             0041            OR    D
4EDB D305           0042 LBEDB      OUT   (£05),A
4EDD DB04           0043 LBEDD      IN    A,(£04)
4EDF CB47           0044            BIT   0,A
4EE1 20FA           0045            JR    NZ,LBEDD
4EE3 3EFD           0046            LD    A,£FD
4EE5 D304           0047            OUT   (£04),A
4EE7 3EFF           0048            LD    A,£FF
4EE9 D304           0049            OUT   (£04),A
4EEB F1             0050            POP   AF
4EEC C9             0051            RET
                    0052 ;
                    0053 :
```

```
4EED            0001 START  ORG  4EEDH
                0002 ;
                0003 ;Assembled by J HOPTON. Aug 82.
                0004 ;
                0005 ;NOTE The program runs at BEEDH.
                0006 ;But is assembled at 4EEDH.
                0007 ;
                0008 ;*C L E A R  B U F F E R (Border)*
                0009 ;
4EED 00DF       0010 L00DF  EQU  £00DF
4EED 1B20       0011 L1B20  EQU  £1B20
4EED A000       0012 LA000  EQU  £A000
4EED A001       0013 LA001  EQU  £A001
4EED A0DD       0014 LA0DD  EQU  £A0DD
4EED A0E0       0015 LA0E0  EQU  £A0E0
4EED A0E1       0016 LA0E1  EQU  £A0E1
4EED BC00       0017 LBC00  EQU  £BC00
4EED BC01       0018 LBC01  EQU  £BC01
4EED 3E00       0019        LD   A,£00
4EEF 32E0A0     0020        LD   (LA0E0),A
                0021 ;
                0022 ;
                0023 ;
                0024 ;
```

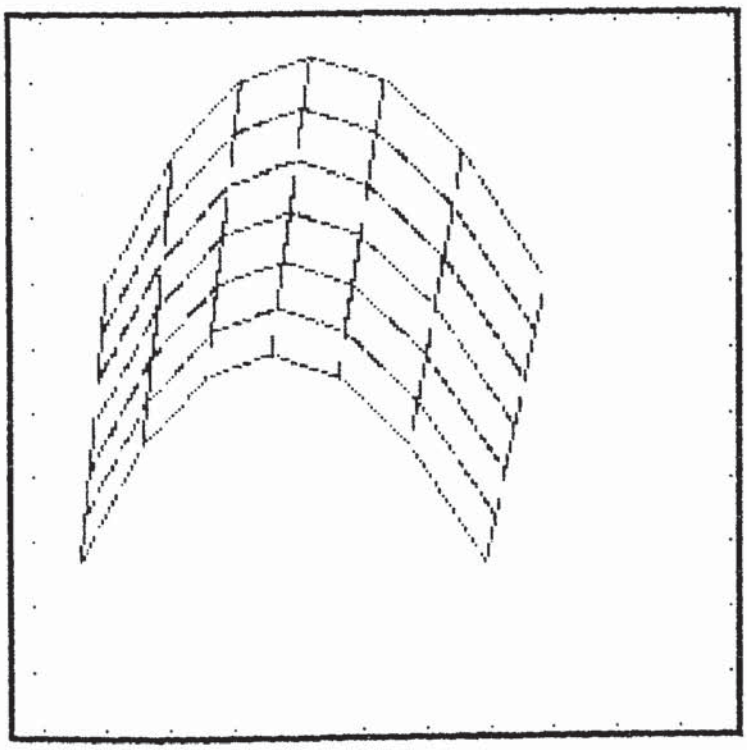
```
                ---- ;
                0029 ;
4EF2 21E0A0     0030        LD    HL,LA0E0    ;Buffer 2nd
4EF5 11E1A0     0031        LD    DE,LA0E1    ;line.
4EF8 01201B     0032        LD    BC,L1B20
4EFB EDB0       0033        LDIR
4EFD 3E60       0034        LD    A,£60
4EFF 3200A0     0035        LD    (LA000),A   ;Start of
4F02 2100A0     0036        LD    HL,LA000    ;Buffer.
4F05 1101A0     0037        LD    DE,LA001
4F08 01DF00     0038        LD    BC,L00DF    ;Width of
4F0B EDB0       0039        LDIR ;            ;Buffer.
4F0D 3E03       0040        LD    A,£03
4F0F 3200BC     0041        LD    (LBC00),A   ;End of
4F12 2100BC     0042        LD    HL,LBC00    ;Buffer.
4F15 1101BC     0043        LD    DE,LBC01
4F18 01DF00     0044        LD    BC,L00DF    ;Width of
4F1B EDB0       0045        LDIR ;            ;Buffer.
4F1D 0621       0046        LD    B,£21       ;33 lines of
4F1F 3E7F       0047        LD    A,£7F       ;border
4F21 2100A0     0048        LD    HL,LA000    ;pattern.
4F24 11DF00     0049        LD    DE,L00DF
4F27 77         0050 LBF27  LD    (HL),A
                0051 ;
```

```
                0058 ;
                0059 ;
4F28 23         0060          INC   HL
4F29 77         0061          LD    (HL),A
4F2A 19         0062          ADD   HL,DE
4F2B 10FA       0063          DJNZ  LBF27       ;Loop.
4F2D 0621       0064          LD    B,£21       ;33 lines OF
4F2F 3E7F       0065          LD    A,£7F       ;border
4F31 21DDA0     0066          LD    HL,LA0DD    ;pattern.
4F34 11DF00     0067          LD    DE,L00DF
4F37 77         0068 LBF37    LD    (HL),A
4F38 23         0069          INC   HL
4F39 77         0070          LD    (HL),A
4F3A 19         0071          ADD   HL,DE
4F3B 10FA       0072          DJNZ  LBF37       ;Loop.
4F3D C9         0073          RET   ;           ;Return.
                0074 ;
                0075 ;
                0076 ;
                0077 ;
                0078 ;
```

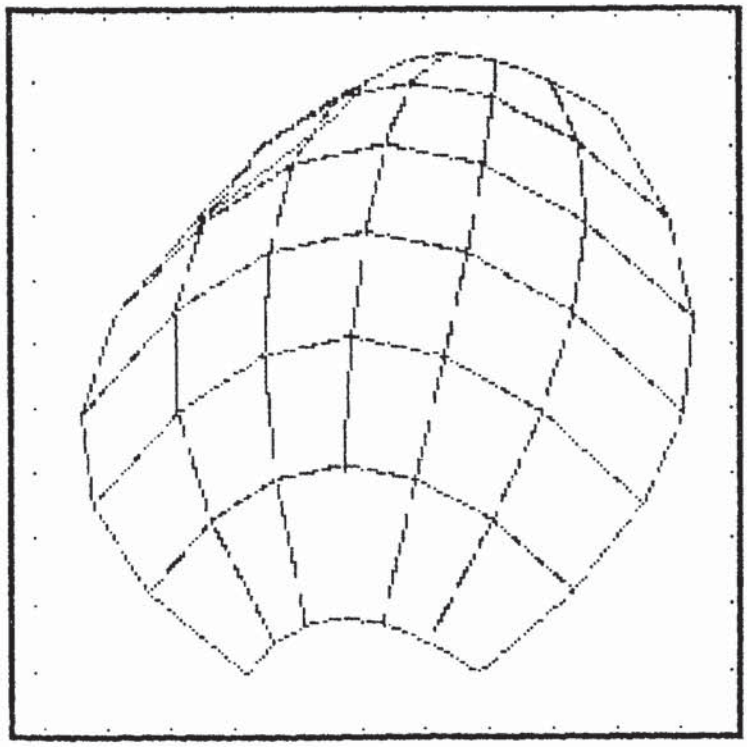
```
4F3E            0001 START  ORG  4F3EH
                0002 ;
                0003 ;Assembled by J HOPTON. Aug 82.
                0004 ;
                0005 ;NOTE The program runs at BF3EH.
                0006 ;But is assembled at 4F3EH.
                0007 ;
                0008 ;* P R O G   L I N E S P A C E *
                0009 ;
4F3E A000       0010 LA000  EQU  £A000
4F3E A1E1       0011 LA1E1  EQU  £A1E1
4F3E BEB8       0012 LBEB8  EQU  £BEB8
4F3E 2100A0     0013        LD   HL,LA000   ;Auto
4F41 11E1A1     0014        LD   DE,LA1E1   ;Changed.
4F44 3E1B       0015        LD   A,£1B      ;1B 33 10
4F46 CDB8BE     0016        CALL LBEB8      ;are print
4F49 3E33       0017        LD   A,£33      ;codes for
4F4B CDB8BE     0018        CALL LBEB8      ;PRINTOUT
4F4E 3E10       0019        LD   A,£10      ;routines.
4F50 CDB8BE     0020        CALL LBEB8
                0021 ;
                0022 ;
                0023 ;
                0024 ;
```

```
                0025 ;
                0026 ;
                0027 ;
                0028 ;
                0029 ;
4F53 3E1B       0030          LD    A,£1B
4F55 CDB8BE     0031          CALL  LBEB8
4F58 3E36       0032          LD    A,£36
4F5A CDB8BE     0033          CALL  LBEB8
4F5D 7E         0034 LBF5D    LD    A,(HL)
4F5E 23         0035          INC   HL
4F5F CDB8BE     0036          CALL  LBEB8
4F62 B7         0037          OR    A
4F63 ED52       0038          SBC   HL,DE
4F65 19         0039          ADD   HL,DE
4F66 D0         0040          RET   NC;         ;Return.
4F67 18F4       0041          JR    LBF5D       ;Loop.
                0042 ;
                0043 ;
                0044 ;
                0045 ;
                0046 ;
                0047 ;
```

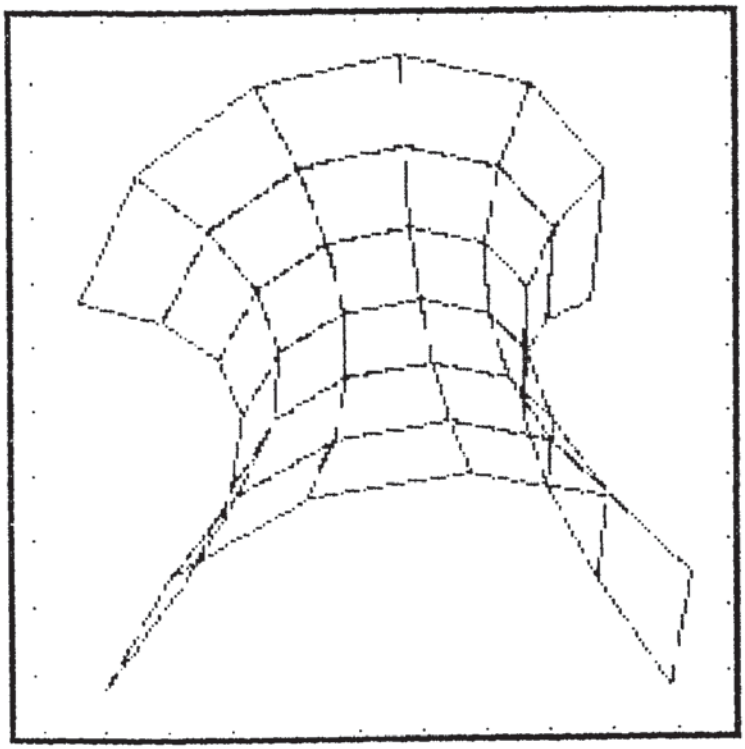
```
32 PRINT"32 Inp Cond":N=2:M=2:P=7
Ok
LIST3000

3000 DATA 150,110,0,150,200,100,150,290,0
3010 DATA 100,110,0,100,200,100,100,290,0
3020 DATA  50,110,0, 50,200,100, 50,290,0
Ok
```



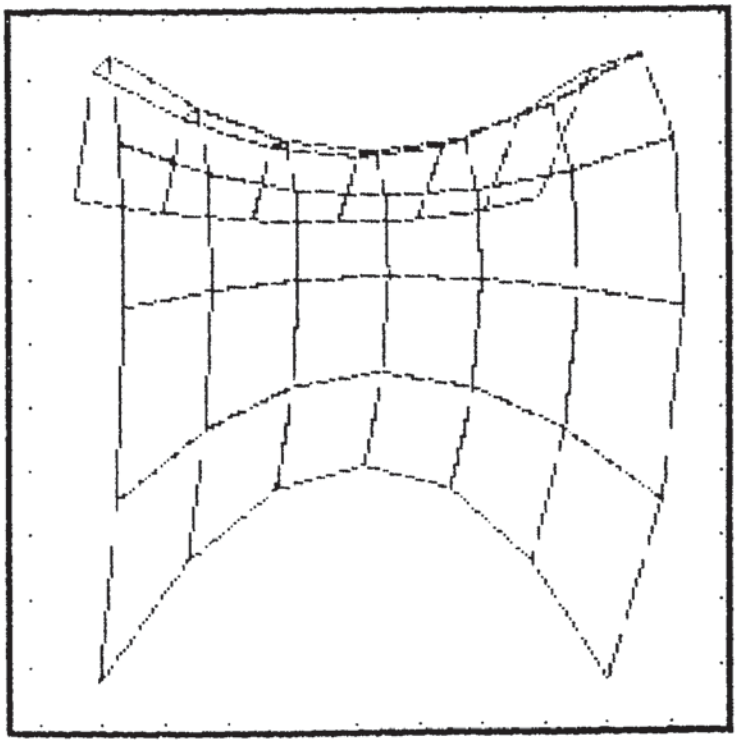
```
32 PRINT"32 Inp Cond":N=4:M=2:P=7
Ok
LIST3000

3000 DATA 39,11,0,39,20, 5,39,25,0
3010 DATA 30, 3,0,30,20,17,30,37,0
3020 DATA 20, 0,0,20,20,20,20,40,0
3030 DATA 10, 3,0,10,20,17,10,37,0
3040 DATA  1,11,0, 1,20, 5, 1,25,0
Ok
```



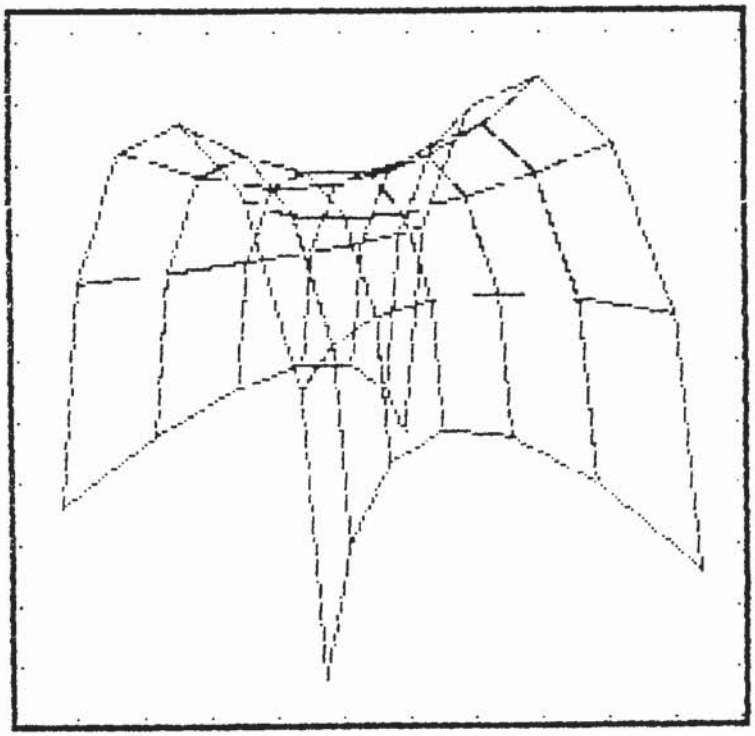
```
32 PRINT"32 Inp Cond":N=3:M=3:P=7
Ok
LIST3000

3000 DATA   0,120,120,120,120,120,120,  0,120
3010 DATA   0,  0,120, 40, 80, 80, 80, 80, 80
3020 DATA  80, 40, 80, 40, 40, 80, 40, 80, 40
3030 DATA  80, 80, 40, 80, 40, 40, 40, 40, 40
3040 DATA   0,120,  0,120,120,  0,120,  0,  0
3050 DATA   0,  0,  0
Ok
```



```
32 PRINT"32 Inp Cond":N=3:M=3:P=7
Ok
LIST3000

3000 DATA    0,120,120,120,120,120,120,120,   0
3010 DATA    0,120,   0,  40,  80,  80,  80,  80,  80
3020 DATA   80,  80,  40,  40,  80,  40,  40,  40,  80
3030 DATA   80,  40,  80,  80,  40,  40,  40,  40,  40
3040 DATA    0,   0,120,120,   0,120,120,   0,   0
3050 DATA    0,   0,   0
Ok
```



```
32 PRINT"32 Inp Cond":N=3:M=3:P=7
Ok
LIST3000

3000 DATA 120,  0,120,120,120,120,  0,120,120
3010 DATA   0,  0,120,  80,  40,  80,  80,  80,  80
3020 DATA  40,  80,  80,  40,  40,  80,  80,  40,  40
3030 DATA  80,  80,  40,  40,  80,  40,  40,  40,  40
3040 DATA 120,  0,  0,120,120,  0,  0,120,  0
3050 DATA   0,  0,  0
Ok
```

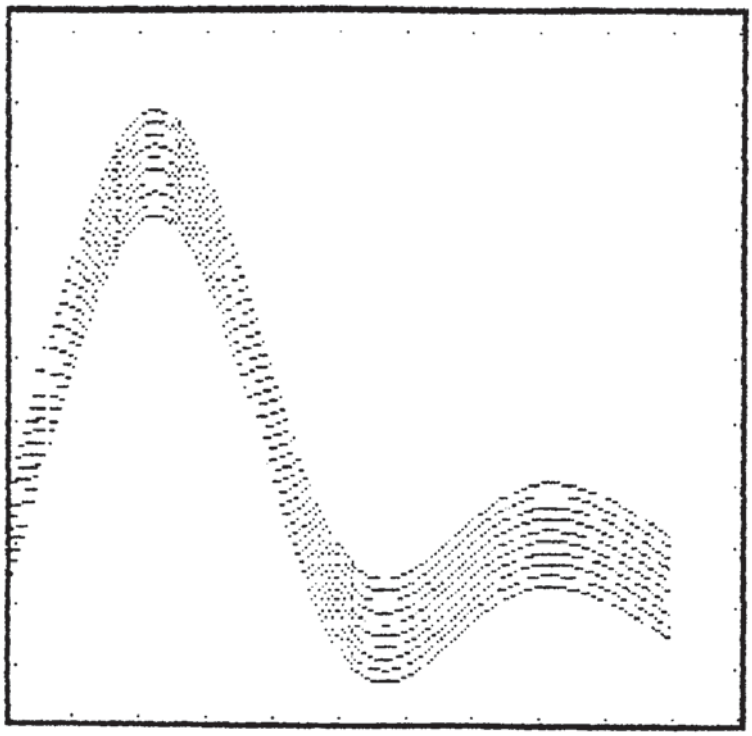


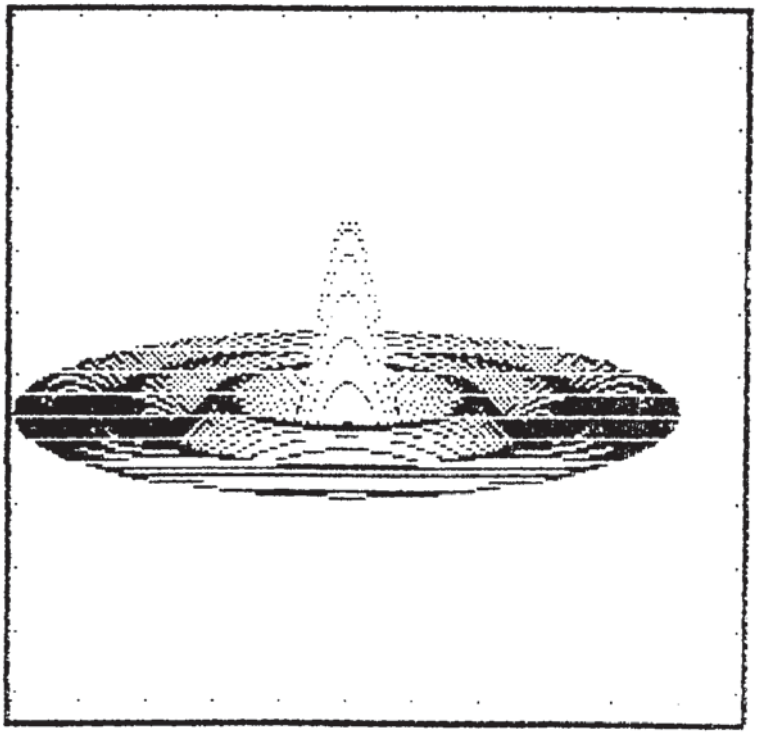
```
MON
HOPSYSTEM

----

10 REM 3 DIMENSIONAL FIGURE.
11 CLEAR:D=3330:E=3328
15 GOSUB1220:REMINMC/BORDER
20 DOKE4100,-16813:REMDOT IN BUFFER
25 FORX=0TO100:E=X*X:B=-50:A=SQR(10000-E)
30 FORI=-ATOASTEP2.5:R=20*SQR(E+I*I)/100
35 IFR=0THEN300
40 F=SIN(R)/R:Y=I/5+F*50:IFY<=BTHEN300
45 B=Y:Y=(50-Y)*1.3:DOKE3330,(INT(100-X)+20)
50 DOKE3328,(INT(Y)+70):A=USR(1)
55 DOKE3330,(INT(100+X)+20):A=USR(1)
300 NEXTI:NEXTX
605 DOKE4100,-16872:A=USR(1):REMPrintBuffer
610 END
700 REM
710 REM
720 REM
730 REM
1210 REM M/C INS
1220 IFDEEK(-16896)=3902THENGOTO1270
1230 PRINT"Ent M/C at BE00H"
1260 FORA=0TO188:READB:DOKE-16896+2*A,B:NEXT
1270 PRINT:PRINT"M/C ENTERED"
1280 DOKE4100,-16896:A=USR(1):REM INIT PIO
1290 DOKE4100,-16813:REM Markers in Buffer
1300 FORA=20TO220STEP20:DOKED,A:DOKEE,4
1320 B=USR(1):DOKED,3:DOKEE,A:B=USR(1)
1330 DOKED,A:DOKEE,223:B=USR(1):DOKED,219
1340 DOKEE,A:B=USR(1):NEXT
1350 POKE27267,1:POKE27483,1:RETURN
1400 REM
1410 REM
1420 REM
```

```
1430 REM
1500 DATA 3902,2003,-194,1747,1235,-1730,1747
1510 DATA-18399,8894,3192,-4659,-13890,8454,33
1530 DATA4512,-24353,16162,-4673,16979,-14913
1540 DATA16077,1727,16127,-13056,-16712,-1776
1550 DATA27075,10943,-16577,8729,-16577,16938
1560 DATA6591,16930,-15937,-8432,3390,-18227
1580 DATA16062,-13046,-16712,-8759,33,4352,224
1590 DATA33,-4864,75,2829,2827,2827,2827,9181
1610 DATA30745,28926,-3784,11229,-4681,-5294,33
1620 DATA-18500,21229,23533,3330,-8935,1058
1630 DATA-8947,-8919,-8919,-8919,1570,8717,3336
1650 DATA1578,-4851,1115,-18675,21229,23533
1660 DATA3328,-18453,21229,15741,-13385,-13529
1670 DATA-13529,-14809,12998,-16714,2090
1680 DATA-13555,-13866,-13323,10367,-13538
1690 DATA8311,15876,6176,-10730,20256,7910
1700 DATA16331,18961,-29824,31327,206,6743
1710 DATA31063,-7706,-11342,-9467,-13564
1720 DATA8263,16122,-11267,15876,-11265,-3836
1730 DATA16073,12800,-24352,-8159,4512,-24351
1750 DATA8193,-4837,16048,12896,-24576,33
1760 DATA4512,-24575,-8447,-4864,16048,12803
1770 DATA-17408,33,4540,-17407,-8447,-4864
1780 DATA1712,15905,8575,-24576,-8431,30464
1790 DATA30499,4121,1786,15905,8575,-24355
1810 DATA-8431,30464,30499,4121,-13830,33
1820 DATA4512,-24095,6974,-18227,16062
1830 DATA-13005,-16712,4158,-18227,16062
1840 DATA-13029,-16712,13886,-18227,32446
1850 DATA-13021,-16712,-4681,6482,6352,1780
1860 DATA15875,-13056,-16712,-1776
1870 DATA-8175,-15616,-16841,0,255
Ok
```

```
10 REM CUBIC SPLINES.J HOPTON. AUG 82.
20 CLEAR:D=3330:E=3328
30 GOSUB190:REMINMC/BORDER
40 DOKE4100,-16813:REMDOT IN BUFFER
50 X(0)=0:X(1)=1:X(2)=2:X(3)=3:X(4)=4
60 Y(0)=1:Y(1)=5:Y(2)=0.5:Y(3)=1:Y(4)=0.6
70 FORJ=1TO10:FORW=0TO4
80 Y(W)=Y(W)-0.125
90 NEXTW
100 GOSUB700
110 NEXTJ
120 DOKE4100,-16872:A=USR(1):REMPrintBuffer
130 END
140 REM
150 REM
160 REM
170 REM
180 REM M/C INS
190 IFDEEK(-16896)=3902THENGOTO220
200 PRINT"Ent M/C at BE00H"
210 FORA=0TO188:READB:DOKE-16896+2*A,B:NEXT
220 PRINT:PRINT"M/C ENTERED"
230 DOKE4100,-16896:A=USR(1):REM INIT PIO
240 DOKE4100,-16813:REM Markers in Buffer
250 FORA=20TO220STEP20:DOKED,A:DOKEE,4
260 B=USR(1):DOKED,3:DOKEE,A:B=USR(1)
270 DOKED,A:DOKEE,223:B=USR(1):DOKED,219
280 DOKEE,A:B=USR(1):NEXT
290 POKE27267,1:POKE27483,1:RETURN
300 REM
310 REM
680 REM
690 REM
700 REM CUBIC SPLINE SUBROUTINE.
710 Q=50:P=1/Q:GOSUB810
720 FORI=0TO3:FORS=ITO1+ISTEPP
730 A=-Y(I)*(S-X(I+1))+Y(I+1)*(S-X(I))
740 B=-D(I)*((S-X(I+1))^3-(S-X(I+1)))/6
750 C=D(I+1)*((S-X(I))^3-(S-X(I)))/6
760 S(I)=A+B+C:XI=S*Q
770 YI=INT(30*S(I)+0.5)+50
780 DOKE3330,XI:DOKE3328,YI:A=USR(1)
790 NEXTS:NEXTI:RETURN
800 REM MATRIX ELEMENTS.
810 FORZ=0TO2:V(Z)=Y(Z)-2*Y(Z+1)+Y(Z+2):NEXTZ
820 D(1)=(15*V(0)-4*V(1)+V(2))*3/28
830 D(2)=(-V(0)+4*V(1)-V(2))*3/7
840 D(3)=(V(0)-4*V(1)+15*V(2))*3/28
850 RETURN
Ok
```

```
340 DATA 3902,2003,-194,1747,1235,-1730,1747
350 DATA-18399,8894,3192,-4659,-13890,8454,33
360 DATA4512,-24353,16162,-4673,16979,-14913
370 DATA16077,1727,16127,-13056,-16712,-1776
380 DATA27075,10943,-16577,8729,-16577,16938
390 DATA6591,16930,-15937,-8432,3390,-18227
400 DATA16062,-13046,-16712,-8759,33,4352,224
410 DATA33,-4864,75,2829,2827,2827,2827,9181
420 DATA30745,28926,-3784,11229,-4681,-5294,33
430 DATA-18500,21229,23533,3330,-8935,1058
440 DATA-8947,-8919,-8919,-8919,1570,8717,3336
450 DATA1578,-4851,1115,-18675,21229,23533
460 DATA3328,-18453,21229,15741,-13385,-13529
470 DATA-13529,-14809,12998,-16714,2090
480 DATA-13555,-13866,-13323,10367,-13538
490 DATA8311,15876,6176,-10730,20256,7910
500 DATA16331,18961,-29824,31327,206,6743
510 DATA31063,-7706,-11342,-9467,-13564
520 DATA8263,16122,-11267,15876,-11265,-3836
530 DATA16073,12800,-24352,-8159,4512,-24351
540 DATA8193,-4837,16048,12896,-24576,33
550 DATA4512,-24575,-8447,-4864,16048,12803
560 DATA-17408,33,4540,-17407,-8447,-4864
570 DATA1712,15905,8575,-24576,-8431,30464
580 DATA30499,4121,1786,15905,8575,-24355
590 DATA-8431,30464,30499,4121,-13830,33
600 DATA4512,-24095,6974,-18227,16062
610 DATA-13005,-16712,4158,-18227,16062
620 DATA-13029,-16712,13886,-18227,32446
630 DATA-13021,-16712,-4681,6482,6352,1780
640 DATA15875,-13056,-16712,-1776
650 DATA-8175,-15616,-16841,0,255
```





APPENDIX 6. STEPPER MOTOR CONTROL.

A Single Step.

Memory.	Opcode.	Meaning, and Mnemonics.	
0C80	3E 0F	LD A , 0F	Set both input ports.
0C82	D3 06	OUT(Port 6),A	Port A.
0C84	D3 07	OUT(Port 7),A	Port B.
0C86	3E 00	LD A , 00	Set motor to run.
0C88	D3 04	OUT(Port 4),A	
0C8A	3E 01	LD A , 01	
0C8C	D3 05	OUT(Port 5),A	
0C8E	06 0A	LD B , 0A	Set a delay.
0C90	10 FE	DJNZ	Delay loop.
0C92	3E 00	LD A , 00	End of pulse.
0C94	D3 05	OUT(Port 5),A	
0C96	DF 5B	STOP	Return to the system monitor.

This programme will step the motor once, in the direction set at location 0C8B.

Enter 00 at 0C8B to set clockwise rotation.

Enter 01 at 0C8B to set anticlockwise rotation.

Continuous Stepping.

Memory.	Opcode.	Mnemonic.	Meaning.
0C80	3E 0F	LD A , 0F	Set both Ports.
0C82	D3 07	OUT (Port 6),A	Port A.
0C84	D3 06	OUT (Port 7),A	Port B.
0C86	3E 00	LD A , 00	Set the direction.
0C88	D3 04	OUT (Port 4),A	
0C8A	3E 01	LD A , 01	
0C8C	D3 05	OUT (Port 5),A	
0C8E	06 0A	LD B , 0A	Set a delay.
0C90	10 FE	DJNZ	Loop.
0C92	3E 00	LD A , 00	End the pulse.
0C94	D3 05	OUT (Port 5),A	
0C96	01 00 20	LD BC , 2000	Loop constant.
0C99	0B	DEC B	Reduce the constant.
0C9A	78	LD A , B	
0C9B	B1	OR C	Test constant (=0)
0C9C	20 FB	JRNZ	If not return (0C99)
0C9E	18 EA	JR	If so return (0C8A)

The direction can be changed at location 0C87 and the stepping rate changed at locations 0C97 and 0C98.

More than one step.

Memory.	Opcode.	Mnemonic.	Meaning.
0C80	3E 0F	LD A , 0F	Set both Ports.
0C82	D3 07	OUT (Port 7),A	Port B.
0C84	D3 06	OUT (Port 6),A	Port A.
0C86	3E 00	LD A , 00	Set direction.
0C88	D3 04	OUT (Port 4),A	
0C8A	11 30 00	LD DE , 30	Set number of steps.
0C8D	7A	LD A , D	
0C8E	B3	OR E	Check for end.
0C8F	CA 89 0C	JP Z	If so jump to 0C89.
0C92	1B	DEC DE	If not decrement the number of steps.
0C93	3E 01	LD A , 01	Delay.
0C95	D3 05	OUT (Port 5),A	Port A.
0C97	06 0A	LD B , 0A	Delay counter.
0C99	10 FE	DJNZ	Loop to 0C99.
0C9B	3E 00	LD A , 00	End of pulse.
0C9D	D3 05	OUT (Port 5),A	
0C9F	01 00 20	LD BC , 2000	Set a time constant.

Continued. [More than one step.]

Memory.	Opcode.	Mnemonic.	Meaning.
0CA2	0B	DEC BC	Reduce the time constant.
0CA3	78	LD A , B	
0CA4	B1	OR C	Is the constant 0 ?
0CA5	20 FB	JRNZ	If not return to 0CA2.
0CA7	18 E1	JR	If so go to 0C8A.
0CA9	DF 5B	STOP	Return to the monitor system.

The duration of the pulse stepping rate can be changed at locations 0C90 and 0C91.

The direction of rotation can be changed at location 0C87.

The number of pulses can be changed at locations 0C8B and 0C8C.

APPENDIX 7. EPROM PROGRAMMER CONTROL SEQUENCE.

Memory.	Label.	Opcode.	Mnemonic.
0C80	Start	EF 20 31 2C 20 50 52	RST Prs
		4F 47 52 41 4D 0D 20	[1, Program]
0C8E		32 2C 20 52 45 2D 50	
		52 4F 52 41 45 4D 0D	
		20	[2, Re-Program]
0C9D		33 2C 20 43 4F 50 59	
		20 54 4F 20 4D 45 4D	
		4F 52 59 0D	[3, Copy to memory]
0CB0		DF 63	RST SCAL DEFB Inlin
0CB2		DF 79	RST SCAL DEFB Rlin
0CB4		38 CA	JR C Start
0CB6		3A 0B 0C	LD A (Argn)
0CB9		FE 02	CP 2
0CBB		20 0D	JR NZ Return
0CBD		DF 60	RST SCAL DEFB Args
0CBF		EB 1D	EX DE HL. Dec E
0CC1		28 09	JR Z Erchk
0CC3		1D 28 26	DEC C. JR Z Erchk
0CC4		28 26	JR Z Prog
0CC6		1D	Dec E
0CC7		CA 61 0D	JP Copy.

Memory.	Label.	Opcode.	Mnemonics.
0CCA	Return	DF 5B	RST SCAL DEFB MRET.
0CCC	Erchk	06 0A	LD B , 10
0CCE	Er 1	CD 6C 0D	CALL Setr
0CD1	Er 2	DB 04	IN A(4)
0CD3		3C	INC A
0CD4		28 0F	JR Z Er 3
0CD6		EF 4E 4F 54	RST Prs
		20 45 52 41	[Not Erased]
		53 45 44	
0CE1		00 0D	Newline
0CE3		DF 5B	RST SCAL DEFB MRET
0CE5	Er 3	CD 8D 0D	CALL Incp
0CE8		28 E7	JR Z Er 2
0CEA		10 E2	DJNZ Er 1
0CEC	Prog	EF 50 52 4F	[Prog]
		47 3a 00	
0CF3		01 01 90	LD BC , 9001 H
0CF6	P 1	CD 6E 0D	Call Setw
0CF9	P 2	7E	LD A, (HL)

Memory.		Opcode.	Mnemonics.
0CFA		D3 04	OUT (4) , A
0CFC		3E 02	LD A , 2
0CFE		FF	RST RDEL
0CFF		3E 03	LD A , 03
0D01		D3 05	OUT (5) , A
0D03		3E 02	LD A , 02
0D05		FF	RST RDEL
0D06		79	LD A , C
0D07		D3 05	OUT (5) , A
0D09	P3	DB 05	IN A , (5)
0D0B		E6 20	AND 20H
0D0D		28 FA	JR Z P3
0D0F		CD 8D 0D	CALL INCP
0D12		28 E5	JR Z P2
0D14		10 E0	DJNZ P1
0D16		D7 54	RST SCAL DEFB SETR
0D18		0E 0A	LD C , 10
0D1A	P4	DF 5D	RST SCAL DEFB TDEL
0D1C		0D	DEC C

Memory.		Opcode.	Mnemonics.
0D1D		20 FB	JR NZ P4
0D1F		EF 56 45 52 49	RST PRS
0D24		46 59 00	[VERIFY]
0D27		0D	NEWLINE
0D28		06 14	LD b, 20
0D2A	V1	D7 40	RST RCAL DEFB SETR
0D2C	V2	DB 04	IN A (4)
0D2E		BE	CP (HL)
0D2F		20 0F	JR NZ ERR1
0D31		D7 5A	RST RSCAL DEFB INCP
0D33		28 F7	JR Z V2
0D35		10 F3	DJNZ V1
0D37	V3	EF 44 4F 4E 45	RST PRS
0D3C		0D 00	[DONE]
0D3E		DF 5B	RST SCAK DEFB MRET
0D40	ERR1	EF 42 49 54 20 46 41	RST PRS
0D47		55 4c 54 59 20 41 54	[BIT FAULTY AT]
0D4E		20 00	
0D50		ED 5B 0E 0C	LD DE , (ARG2)
0D54		AF	XOR A

Memory.		Opcode.	Mnemonics.
0D55		ED 52	SBC HL, DE
0D57		7C	LD A , H
0D58		DF 7A	RST SCAL DEFB B1HEX
0D5A		7D	LD A, L
0D5B		DF 68	RST SCAL DEFB B2HEX
0D5D		DF 6A	RST SCAL DEFB CRLF
0D5F		DF 5B	RST SCAL DEFB MRET
0D61	COPY	D7 09	RST RCAL DEFB SETR
0D63	COPY1	DB 04	IN A , (4)
0D65		77	LD (HL) , A
0D66		D7 25	RST RCAL DEFB INCP
0D68		28 F9	JR COPY1
0D6A		18 CB	JR V3
0D6C	SETR	0E 00	LD C , 00
0D6E	SETW	3E CF	LD A , CF
0D70		F5	PUSH AF
0D71		D3 06	OUT (6) , A
0D73		79	LD A , C
0D74		3D	DEC A
0D75		D3 06	OUT (6) , A

Memory.		Opcode.	Mnemonics.
0D77		F1	POP AF
0D78		D3 07	OUT (7) , A
0D7A		3E 30	LD A , 30
0D7C		D3 07	OUT (7) , A
0D7E		79	LD A , C
0D7F		C6 08	ADD A , 8
0D81		D3 05	OUT (5) , A
0D83		3E 02	LD A , 02
0D85		FF	RST RDEL
0D86		79	LD A , C
0D87		D3 05	OUT (5) , A
0D89		2A 0E 0C	LD HL , (ARG2)
0D8C		C9	RET
0D8D	INCP	79	LD A , C
0D8E		C6 04	ADD A , 4
0D90		D3 05	OUT (5) , A
0D92		3E 02	LD A , 02
0D94		FF	RST RDEL
0D95		79	LD A , C
0D96		D3 05	OUT (5) , A
0D98		23	INC HL
0D99		3E 02	LD A , 02
0D9B		FF	RST RDEL
0D9C		DB 05	IN A , (5)
0D9E		E6 10	AND 10
0DA0		C9	RET

REFERENCES.

References.

1. Z80 Instant Programs. J Hopton. Sigma Technical Press. ISBN 0 905104 099 (1st Ed) 1979. ISBN 0 905104 196 (2nd Ed) 1982.

2. Hydraulic Feedrate on the "AutoWard", investigation and re-design. D G Fox. MSc Project. Jan 1967. University of Aston.

3. Catalogue. Radio Spares. Electronic Spares Distributor. 1979/82.

4. Z80 MK 3880 CPU. Mostek Technical Manual. 1977. MK 78505.

5. Memory Products. Mostek Catalogue. 1977. MM 2026/177.

6. Memory Data Book. Radio Shack. National Semi-Conductors. Catalogue 62-1376.

7. HD 6402 UART. Harris Data Sheet on the CMOS LSI Universal Asynchronous Receiver and Transmitter. Jan 1978.

8. Z80 MK 3881 PIO. Mostek Technical Manual on the Parallel Input Output Controller. MK 78506. Feb 1978.

References (Continued.)

9. TTL Data Book. — Texas Instruments Data Book 1980. ISBN 0904047 - 27 - X .

10. Solid State Integrated Circuits Book. — R C A . Somerville. 1976. USA 4 / 76.

11. CMOS Semiconductors. — Motorola CMOS Semiconductor 1976. Library. Volume 5. Series B.

12. RG 512 Graphics. — Lear Siegler Retro User's Graphics Manual. ADM 3A. 1977.

13. The Unisurf System. — P E Bezier. Proc Roy Soc. Vol A321, pp 207-218, 1971.

14. Numerical Control Mathematics and Applications. — P Bezier. J Wiley & Sons, Inc London. 1972.

15. SAA 1027 Drive. — Mullard Data Sheet on the Stepper Motor Drive. Jan 1980.

16. Integrated Optics. — M J Broolfield. Communications International. Aug 1978. Pages 21 to 27.

References. (Continued)

17. Optical Fibre Technology. — F Wilkinsons. Electronic Engineering. April 1980. P 93,97.

18. Engineering Approach To The Fibre Optic Link. — D L Jones. Electronic Engineering Pages 65,84. April 1980.

19. Fibre Optic Components. — Centronics Electro Optics Information Sheet. FTIR 1 / 2. 1977.

Bibliography.

BARDEN W . The Z80 Microcomputer Handbook.
H W Sams & Co . Indiana.
ISBN 0672 21500 4.

GOLDSBROUGH P F. Microcomputer Interfacing with the 8255.
H W Sams & Co . Indiana.
ISBN 0672 21614 0.

LARSEN D G. Experiments in Digital Electronics: 8080A.
H W Sams & Co . Indiana.
ISBN 0672 21551 9.

NICHOLS J C. Z80 Microprocessor Interfacing. Book 3.
H W Sams & Co . Indiana.
ISBN 0672 21610 8.

OSBORNE A. Z80 Programming for Logic Design.
Osborne & Associates. California. No 7001.

POE C E. S100 and other MICRO Buses.
H W Sams & Co . Indiana.
ISBN 0672 21587 X.

RONY P R. The 8080A Bugbook.
H W Sams & Co . Indiana.
ISBN 0672 21447 4.

Texas. Interface Circuits Data Book.
Texas Instruments.
ISBN 0 904047 24 5.

Texas. Memory and Microprocessor Data Book.
Texas Instruments.
ISBN 0 904047 15 6.

Texas. MOS Memory Data Book.
Texas Instruments.
ISBN 0 89512 105 0.

Texas. TTL Data Book.
Texas Instruments.
ISBN 0 904047 27 X.

ZAKS R. How to Programme the Z80.
Sybex. California.
ISBN 089 588 047 4.

ZAKS R. Microprocessor Interfacing Techniques.
Sybex. California.
ISBN 089 588 003 2.

ZILOG. Data Book.
Zilog Berkshire. Z80 / 12. 1978.